



Piano di qualifica

SnakeByte (Gruppo 1):

Valeria Baleanu, Leonardo Pellizzon, Filippo Venzo, Giuseppe De Fina,
Francesco Pasqual, Christian Libralato, Luca Granziero
(2109911, 2111006, 2113705, 2113187, 2103119, 2101047, 2075512)

Informazioni documento			
Versione	Data	Stato	Destinatari
2.0.0	05/04/2026	Approvato	Interni: SnakeByte Esterni: prof. Vardanega Tullio, prof. Cardin Riccardo, Vimar

Contatti: snakebyteteam@gmail.com

Registro delle modifiche					
Versione	Data	Autore	Verificatore	Approvatore	Descrizione
2.0.0	05/04/2026	-	-	C. Libralato	Approvazione
1.1.0	04/04/2026	L. Granziero	G. De Fina	-	Aggiornamento allo sprint 11, Aggiunta sezione di test
1.0.3	01/04/2026	L. Granziero	G. De Fina	-	Aggiornamento allo sprint 10
1.0.2	13/03/2026	F. Venzo	C. Libralato	-	Aggiornamento allo sprint 9
1.0.1	08/03/2026	L. Pellizzon	F. Pasqual	-	Aggiornamento allo sprint 8
1.0.0	13/02/2026	-	-	F. Venzo	Approvazione
0.2.2	10/02/2026	F. Venzo	F. Pasqual, L. Granziero	-	Aggiunto elenco delle tabelle
0.2.1	04/02/2026	F. Venzo	V. Baleanu	-	Aggiunta descrizione ai grafici, campagna di test di unità e campagna di test di integrazione
0.2.0	28/12/2025	V. Baleanu	F. Pasqual	-	Aggiunta tabelle delle soglie per le metriche di processo e di prodotto
0.1.1	27/12/2025	V. Baleanu	F. Pasqual	-	Aggiunta Introduzione
0.1.0	22/12/2025	V. Baleanu	F. Pasqual	-	Prima Stesura

Indice

1	Introduzione	5
1.1	Finalità del documento	5
1.2	Glossario	5
1.3	Riferimenti	5
1.3.1	Riferimenti Normativi	5
1.3.2	Riferimenti Informativi	5
2	Qualità di processo	6
2.1	Processi primari	6
2.1.1	Fornitura	6
2.1.2	Sviluppo	6
2.2	Processi di supporto	6
2.2.1	Documentazione	6
2.2.2	Verifica	7
2.2.3	Gestione della qualità	7
2.3	Processi organizzativi	7
2.3.1	Gestione dei processi	7
3	Qualità di prodotto	7
3.1	Funzionalità	8
3.2	Affidabilità	8
3.3	Usabilità	8
3.4	Efficienza	8
3.5	Manutenibilità	8
3.6	Portabilità	9
4	Strategie di Testing	9
4.1	Test di sistema	9
4.2	Test di accettazione	15
4.3	Test di integrazione	17
4.4	Test di unità	100
5	Cruscotto di qualità	212
5.1	MPC1 e MPC2 - Planned Value e Earned Value	212
5.2	MPC3 e MPC7 - Actual Cost e Estimate To Complete	213
5.3	MPC4 e MPC5 - Cost Performance Index e Schedule Performance Index	213
5.4	MPC6 - Estimate At Completion	214
5.5	MPC8 - Requirements Stability Index	214
5.6	MPC9 - Indice di Gulpease	215
5.7	MPC10 - Indice di Frammentazione	215
5.8	MPC12 - Percentuale Metriche Soddisfatte	216
5.9	MPC13 - Percentuale Rischi Inattesi	216
5.10	MPC14 - Labor Efficiency	217
5.11	MPD1 - Percentuale Requisiti Obbligatoriosi Soddisfatti	217
5.12	MPD2 - Percentuale Requisiti Opzionali Soddisfatti	218
5.13	MPD3 - Percentuale Requisiti Desiderabili Soddisfatti	218
5.14	MPD4 - Line Coverage	219
5.15	MPD5 - Branch Coverage	219
5.16	MPD6 - Statement Coverage	220
5.17	MPD7 - Profondità Massima di Navigazione	220
5.18	MPD8 - Tempo Medio di Risposta	221
5.19	MPD9 - Complessità Ciclomatica	221
5.20	MPD10 - Compatibilità Cross-Browser	222

Elenco delle figure

1	Andamento PV e EV in base al periodo	212
2	Andamento AC e ETC in base al periodo	213
3	Andamento CPI e SPI in base al periodo	213
4	Andamento EAC in base al periodo	214
5	Andamento RSI in base al periodo	214
6	Andamento IG dei documenti principali, in base al periodo	215
7	Andamento IF dei documenti principali, in base al periodo	215
8	Andamento PMS in base al periodo	216
9	Andamento PRI in base al periodo	216
10	Andamento LE in base al periodo	217
11	Andamento PROS in base al periodo	217
12	Andamento PRPS in base al periodo	218
13	Andamento PRDS in base al periodo	218
14	Andamento LC in base al periodo	219
15	Andamento BC in base al periodo	219
16	Andamento SC in base al periodo	220
17	Andamento PMN in base al periodo	220
18	Andamento TMR in base al periodo	221
19	Andamento CC in base al periodo	221
20	Andamento CCB in base al periodo	222

Elenco delle tabelle

1	Tabella dei valori soglia per metriche di fornitura	6
2	Tabella dei valori soglia per metriche di sviluppo	6
3	Tabella dei valori soglia per metriche di documentazione	6
4	Tabella dei valori soglia per metriche di verifica	7
5	Tabella dei valori soglia per metriche di gestione della qualità	7
6	Tabella dei valori soglia per metriche di gestione dei processi	7
7	Tabella dei valori soglia per metriche di funzionalità	8
8	Tabella dei valori soglia per metriche di affidabilità	8
9	Tabella dei valori soglia per metriche di usabilità	8
10	Tabella dei valori soglia per metriche di efficienza	8
11	Tabella dei valori soglia per metriche di manutenibilità	8
12	Tabella dei valori soglia per metriche di portabilità	9
13	Tabella dei test di sistema	15
14	Tabella dei test di accettazione	17
15	Tabella dei test di integrazione	99
16	Tabella dei test di unita	212

1 Introduzione

1.1 Finalità del documento

Il presente documento ha lo scopo di definire e documentare gli obiettivi di qualità che il gruppo *SnakeByte* si impegna a perseguire durante l'intero ciclo di vita del progetto. In particolare, il Piano di qualifica_G comprende i seguenti elementi:

- **Piano della Qualità:** tutte le attività volte a definire gli obiettivi qualitativi e a stabilire i processi e le risorse necessarie per il loro conseguimento;
- **Controllo di Qualità:** le attività finalizzate ad accertare la conformità dei processi e dei prodotti agli standard, ai requisiti e alle metriche definite;
- **Miglioramento continuo:** le azioni intraprese per analizzare i risultati del controllo della qualità e per introdurre interventi correttivi e migliorativi sui processi e sul prodotto.

Il documento si rivolge sia ai membri del gruppo, come guida per il mantenimento degli standard qualitativi, sia ai committenti e al proponente, come dimostrazione dell'impegno del gruppo verso la qualità del prodotto finale.

1.2 Glossario

Il documento cita alcuni termini la cui definizione può risultare ambigua. Per questo è possibile consultare il glossario_G il quale contiene le definizioni di tali espressioni, che saranno marcate da una lettera G a pedice.

1.3 Riferimenti

1.3.1 Riferimenti Normativi

- **Norme di Progetto 2.0.0:**
https://snakebyteteam.github.io/3-PB/Documenti_interni/Norme_di_progetto/Norme_di_Progetto_v2.0.0.pdf
(consultato il 05/04/2026).
- **Vimar View4Life Capitolato di Ingegneria del Software Università di Padova 2025-2026:**
<https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C9.pdf>
(consultato il 05/04/2026).

1.3.2 Riferimenti Informativi

- **Glossario:**
https://snakebyteteam.github.io/3-PB/Documenti_interni/Glossario/glossario_v2.0.0.pdf
(consultato il 05/04/2026).
- **Standard ISO 9000:2015 - Quality management systems. Fundamentals and vocabulary:**
<https://www.iso.org/standard/45481.html>
(consultato il 05/04/2026).
- **Standard ISO/IEC 12207:1995:**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
(consultato il 05/04/2026).
- **Standard ISO/IEC 9126:2001:**
https://it.wikipedia.org/wiki/ISO/IEC_9126
(consultato il 05/04/2026).

2 Qualità di processo

La qualità di processo_G ha lo scopo di garantire che le attività di sviluppo del software siano pianificate, eseguite, monitorate e migliorate in modo sistematico, al fine di ridurre il rischio di errori o incongruenze durante l'intero ciclo di vita del progetto.

Per la definizione e la classificazione dei processi di qualità, il progetto fa riferimento allo standard ISO/IEC 12207:1995, che descrive i processi del ciclo di vita del software e li suddivide nelle seguenti categorie:

- **Processi primari:** sono direttamente coinvolti nella realizzazione, fornitura e mantenimento del prodotto *software*;
- **Processi di supporto:** affiancano i processi primari fornendo attività di supporto come la documentazione e la verifica;
- **Processi organizzativi:** forniscono le risorse necessarie all'esecuzione dei processi primari e di supporto, favorendo il miglioramento continuo dell'organizzazione.

2.1 Processi primari

2.1.1 Fornitura

Nella tabella che segue il termine *BAC* (*Budget At Completion*) rappresenta il budget totale previsto inizialmente per il completamento del progetto.

Codice	Nome	Valore accettabile	Valore ottimo
MPC1	Planned Value	≥ 0	$\leq BAC$
MPC2	Earned Value	≥ 0	$\leq EAC$
MPC3	Actual Cost	≥ 0	$\leq EAC$
MPC4	Cost Performance Index	≥ 0.80	1.00
MPC5	Schedule Performance Index	≥ 0.80	1.00
MPC6	Estimate At Completion (EAC)	$\leq 120\% \cdot BAC$	$\leq BAC$
MPC7	Estimate To Complete	≥ 0	$\leq BAC$

Tabella 1: Tabella dei valori soglia per metriche di fornitura

2.1.2 Sviluppo

Codice	Nome	Valore accettabile	Valore ottimo
MPC8	Requirements Stability Index	$\geq 80\%$	100%

Tabella 2: Tabella dei valori soglia per metriche di sviluppo

2.2 Processi di supporto

2.2.1 Documentazione

Codice	Nome	Valore accettabile	Valore ottimo
MPC9	Indice di Gulpease	≥ 60	≥ 80
MPC10	Indice di Frammentazione (IF)	$5\% \leq IF \leq 20\%$	$10\% \leq IF \leq 15\%$

Tabella 3: Tabella dei valori soglia per metriche di documentazione

2.2.2 Verifica

Codice	Nome	Valore accettabile	Valore ottimo
MPC11	Test Success Rate	$\geq 80\%$	100%

Tabella 4: Tabella dei valori soglia per metriche di verifica

2.2.3 Gestione della qualità

Codice	Nome	Valore accettabile	Valore ottimo
MPC12	Percentuale Metriche Soddisfatte	$\geq 80\%$	100%

Tabella 5: Tabella dei valori soglia per metriche di gestione della qualità

2.3 Processi organizzativi

2.3.1 Gestione dei processi

Codice	Nome	Valore accettabile	Valore ottimo
MPC13	Percentuale Rischi Inattesi	$\leq 20\%$	0%
MPC14	Labor Efficiency	$\geq 70\%$	$\geq 100\%$

Tabella 6: Tabella dei valori soglia per metriche di gestione dei processi

3 Qualità di prodotto

La qualità di prodotto_G è definita secondo un modello basato sullo standard ISO/IEC 9126, che struttura la qualità in caratteristiche e sottocaratteristiche misurabili attraverso metriche specifiche.

Per il prodotto *software* sviluppato, il gruppo *SnakeByte* ha individuato le seguenti caratteristiche qualitative da monitorare e garantire:

- **Funzionalità:** capacità di fornire funzioni che soddisfino i requisiti definiti;
- **Affidabilità:** capacità di mantenere il livello di prestazione richiesto quando utilizzato in condizioni specifiche;
- **Usabilità:** facilità con cui gli utenti possono apprendere, comprendere e utilizzare il prodotto;
- **Efficienza:** capacità di fornire prestazioni adeguate in relazione alle risorse utilizzate;
- **Manutenibilità:** facilità con cui il prodotto può essere modificato, corretto o esteso, favorendo interventi di manutenzione;
- **Portabilità:** capacità di essere trasferito e adattato a diversi ambienti operativi, piattaforme o configurazioni hardware e software.

3.1 Funzionalità

Codice	Nome	Valore accettabile	Valore ottimo
MPD1	Percentuale Requisiti Obbligatori Soddisfatti	100%	100%
MPD2	Percentuale Requisiti Opzionali Soddisfatti	0%	100%
MPD3	Percentuale Requisiti Desiderabili Soddisfatti	0%	100%

Tabella 7: Tabella dei valori soglia per metriche di funzionalità

3.2 Affidabilità

Codice	Nome	Valore accettabile	Valore ottimo
MPD4	Line Coverage	$\geq 75\%$	100%
MPD5	Branch Coverage	$\geq 75\%$	100%
MPD6	Statement Coverage	$\geq 75\%$	100%

Tabella 8: Tabella dei valori soglia per metriche di affidabilità

3.3 Usabilità

Codice	Nome	Valore accettabile	Valore ottimo
MPD7	Profondità Massima di Navigazione	≤ 7 click	≤ 3 click

Tabella 9: Tabella dei valori soglia per metriche di usabilità

3.4 Efficienza

Codice	Nome	Valore accettabile	Valore ottimo
MPD8	Tempo Medio di Risposta	≤ 5 secondi	≤ 1 secondo

Tabella 10: Tabella dei valori soglia per metriche di efficienza

3.5 Manutenibilità

Codice	Nome	Valore accettabile	Valore ottimo
MPD9	Complessità Ciclomatica	≤ 20	≤ 10

Tabella 11: Tabella dei valori soglia per metriche di manutenibilità

3.6 Portabilità

Codice	Nome	Valore accettabile	Valore ottimo
MPD10	Compatibilità Cross-Browser	≥ 80%	100%

Tabella 12: Tabella dei valori soglia per metriche di portabilità

4 Strategie di Testing

4.1 Test di sistema

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-1	Verificare che un Utente non autenticato possa effettuare l'autenticazione con il Sistema	RF1-OB	S
TS-2	Verificare che l'Utente non autenticato, durante il processo di autenticazione, possa inserire il proprio username	RF2-OB	S
TS-3	Verificare che l'Utente non autenticato, durante il processo di autenticazione, possa inserire la propria password	RF3-OB	S
TS-4	Verificare che l'Utente non autenticato riceva un messaggio di errore a seguito di un tentativo di autenticazione fallito	RF4-OB	S
TS-5	Verificare che l'Utente autenticato possa impostare una nuova password	RF5-OB	S
TS-6	Verificare che l'Utente non registrato possa essere autenticato con la password temporanea fornita dall'amministratore e impostare una nuova password	RF6-OB	S
TS-7	Verificare che l'Utente autenticato con la password temporanea possa impostare una nuova password	RF7-OB	S
TS-8	Verificare che l'Utente non registrato riceva un errore a seguito di un tentativo di autenticazione con password temporanea fallito	RF8-OB RF9-OB	S
TS-9	Verificare che l'Utente autenticato con la password temporanea riceva un errore nel caso la nuova password sia uguale alla password temporanea	RF10-OB	S
TS-10	Verificare che l'Utente autenticato con la password temporanea riceva un errore nel caso la nuova password non rispetti i criteri richiesti	RF11-OB	S
TS-11	Verificare che l'amministratore possa visualizzare se un account MyVimar è configurato nel Sistema	RF12-OB	S
TS-12	Verificare che l'amministratore possa collegare un account MyVimar al Sistema	RF13-OB	S
TS-13	Verificare che l'amministratore possa rimuovere un account MyVimar dal Sistema	RF14-OB	S
TS-14	Verificare che l'amministratore possa visualizzare l'elenco degli utenti del Sistema	RF15-DE	S
TS-15	Verificare che l'amministratore, visualizzando l'elenco degli utenti del Sistema, possa visualizzare un utente del Sistema nel dettaglio	RF16-OB	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-16	Verificare che l'amministratore, visualizzando un utente del Sistema nel dettaglio, ne visualizzi il nome	RF17-OB	S
TS-17	Verificare che l'amministratore, visualizzando un utente del Sistema nel dettaglio, ne visualizzi il cognome	RF18-OB	S
TS-18	Verificare che l'amministratore, visualizzando un utente del Sistema nel dettaglio, ne visualizzi l'username	RF19-OB	S
TS-19	Verificare che l'amministratore possa creare un utente Operatore Sanitario	RF20-DE	S
TS-20	Verificare che l'amministratore, creando un utente Operatore Sanitario, possa inserirne il nome	RF21-DE	S
TS-21	Verificare che l'amministratore, creando un utente Operatore Sanitario, possa inserirne il cognome	RF22-DE	S
TS-22	Verificare che l'amministratore, creando un utente Operatore Sanitario, possa inserire un username	RF23-DE	S
TS-23	Verificare che l'amministratore, creando un utente Operatore Sanitario, possa generare una password temporanea	RF24-DE	S
TS-24	Verificare che l'amministratore riceva un errore se l'username utilizzato per la creazione di un utente Operatore Sanitario è già in uso	RF25-DE	S
TS-25	Verificare che l'amministratore possa eliminare un utente Operatore Sanitario	RF26-DE	S
TS-26	Verificare che l'amministratore possa visualizzare tutti i reparti del Sistema	RF27-OB	S
TS-27	Verificare che l'amministratore possa visualizzare un reparto del Sistema nel dettaglio	RF28-OB	S
TS-28	Verificare che l'amministratore, visualizzando un reparto del Sistema nel dettaglio, possa visualizzarne il nome	RF29-OB	S
TS-29	Verificare che l'amministratore possa creare un nuovo reparto	RF30-OB	S
TS-30	Verificare che l'amministratore, creando un nuovo reparto, possa inserirne il nome	RF31-OB	S
TS-31	Verificare che l'amministratore riceva un errore se il nome inserito per creare il nuovo reparto è già in uso	RF32-OB	S
TS-32	Verificare che l'amministratore possa modificare il nome di un reparto del Sistema	RF33-OB	S
TS-33	Verificare che l'amministratore possa eliminare un reparto del Sistema	RF34-OB	S
TS-34	Verificare che l'amministratore possa assegnare un reparto ad un utente Operatore Sanitario	RF35-OB	S
TS-35	Verificare che l'amministratore, assegnando un reparto ad un utente Operatore Sanitario, selezioni un utente Operatore Sanitario	RF36-OB	S
TS-36	Verificare che l'amministratore, assegnando un reparto ad un utente Operatore Sanitario, selezioni un reparto del Sistema	RF37-OB	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-37	Verificare che l'amministratore possa rimuovere un utente Operatore Sanitario dall'assegnazione ad un reparto	RF38-OB	S
TS-38	Verificare che l'amministratore possa assegnare un reparto del Sistema ad un appartamento	RF39-OB	S
TS-39	Verificare che l'amministratore, assegnando un reparto del Sistema ad un appartamento, selezioni un appartamento	RF40-OB	S
TS-40	Verificare che l'amministratore, assegnando un reparto del Sistema ad un appartamento, selezioni un reparto del Sistema	RF41-OB	S
TS-41	Verificare che l'amministratore possa rimuovere un appartamento dall'assegnazione ad un reparto	RF42-OB	S
TS-42	Verificare che l'utente possa visualizzare una dashboard che rappresenta lo stato del sistema	RF43-OB	S
TS-43	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di gestione allarmi	RF44-OB	S
TS-44	Verificare che l'utente, visualizzando il modulo di gestione allarmi, possa visualizzare ogni singolo allarme	RF45-OB	S
TS-45	Verificare che l'utente, visualizzando un singolo allarme, possa visualizzare il segnale di pericolo	RF46-OB	S
TS-46	Verificare che l'utente, visualizzando un singolo allarme, possa visualizzarne il nome	RF47-OB	S
TS-47	Verificare che l'utente, visualizzando un singolo allarme, possa visualizzarne il tempo trascorso dallo scatto	RF48-OB	S
TS-48	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di statistiche allarmi	RF49-OB	S
TS-49	Verificare che l'utente, visualizzando il modulo di statistiche allarmi, possa visualizzare il numero di allarmi risolti	RF50-OB	S
TS-50	Verificare che l'utente, visualizzando il modulo di statistiche allarmi, possa visualizzare il numero di allarmi attivi	RF51-OB	S
TS-51	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di informazioni Utente	RF52-OB	S
TS-52	Verificare che l'utente, visualizzando il modulo di informazioni utente, possa visualizzare il nome Utente	RF53-OB	S
TS-53	Verificare che l'utente, visualizzando il modulo di informazioni utente, possa visualizzare il cognome Utente	RF54-OB	S
TS-54	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di analisi clima	RF55-OB	S
TS-55	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di analisi consumi	RF56-OB	S
TS-56	Verificare che l'utente, visualizzando la dashboard, possa visualizzare un modulo di analisi presenze	RF57-OB	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-57	Verificare che l'amministratore possa aggiungere alla visualizzazione della dashboard il modulo di statistiche allarmi	RF58-DE	NI
TS-58	Verificare che l'amministratore possa rimuovere dalla visualizzazione della dashboard il modulo di statistiche allarmi	RF59-DE	NI
TS-59	Verificare che l'amministratore possa aggiungere alla visualizzazione della dashboard il modulo di informazioni Utente	RF60-DE	NI
TS-60	Verificare che l'amministratore possa rimuovere dalla visualizzazione della dashboard il modulo di informazioni Utente	RF61-DE	NI
TS-61	Verificare che l'amministratore possa aggiungere alla visualizzazione della dashboard il modulo di analisi clima	RF62-DE	NI
TS-62	Verificare che l'amministratore possa rimuovere dalla visualizzazione della dashboard il modulo di analisi clima	RF63-DE	NI
TS-63	Verificare che l'amministratore possa aggiungere alla visualizzazione della dashboard il modulo di analisi consumi	RF64-DE	NI
TS-64	Verificare che l'amministratore possa rimuovere dalla visualizzazione della dashboard il modulo di analisi consumi	RF65-DE	NI
TS-65	Verificare che l'utente possa risolvere un allarme	RF66-OB	S
TS-66	Verificare che l'utente possa visualizzare le analytics	RF67-OB	S
TS-67	Verificare che l'utente, visualizzando le analytics, possa visualizzare l'elenco dei suggerimenti risparmio energetico	RF68-OB	S
TS-68	Verificare che l'utente, visualizzando l'elenco dei suggerimenti risparmio energetico, possa visualizzare un suggerimento risparmio energetico	RF69-OB	S
TS-69	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato al consumo energetico	RF70-OB	S
TS-70	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato alle anomalie dell'impianto	RF71-OB	S
TS-71	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato al rilevamento presenze	RF72-OB	S
TS-72	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato alla presenza prolungata nello stesso ambiente	RF73-OB	S
TS-73	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato alle variazioni di temperatura	RF74-OB	S
TS-74	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato agli allarmi inviati e risolti	RF75-OB	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-75	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato alla frequenza degli allarmi	RF76-OP	S
TS-76	Verificare che l'utente, visualizzando le analytics, possa visualizzare il grafico dedicato alla frequenza delle cadute	RF77-OP	S
TS-77	Verificare che l'utente possa visualizzare un appartamento del Sistema	RF78-OB	S
TS-78	Verificare che l'utente, visualizzando un appartamento del Sistema, possa visualizzarne il nome	RF79-OB	S
TS-79	Verificare che l'utente, visualizzando un appartamento del Sistema, possa visualizzarne la mappa degli allarmi	RF80-OP	S
TS-80	Verificare che l'utente, visualizzando un appartamento del Sistema, possa visualizzarne le stanze	RF81-OB	S
TS-81	Verificare che l'utente possa visualizzare una stanza di un appartamento nel dettaglio	RF82-OB	S
TS-82	Verificare che l'amministratore, visualizzando una stanza nel dettaglio, possa visualizzarne il nome	RF83-OP	S
TS-83	Verificare che l'amministratore, visualizzando una stanza nel dettaglio, possa visualizzarne l'elenco dei dispositivi	RF84-OP	S
TS-84	Verificare che l'amministratore, visualizzando l'elenco dei dispositivi di una stanza, possa visualizzare un singolo dispositivo di tipo: <ul style="list-style-type: none"> • termostato; • sensore di caduta; • sensore di presenza; • punto luce; • pulsante di allarme; • porta di ingresso; • tapparella. 	RF85-OB	S
TS-85	Verificare che l'amministratore, visualizzando un dispositivo, possa visualizzarne il nome	RF86-OB	S
TS-86	Verificare che l'amministratore, visualizzando un dispositivo, possa visualizzarne lo stato	RF87-OB	S
TS-87	Verificare che l'amministratore, visualizzando un dispositivo, possa visualizzarne le azioni eseguibili	RF88-OP	S
TS-88	Verificare che l'amministratore possa abilitare un appartamento nel Sistema	RF89-OB	S
TS-89	Verificare che l'amministratore possa disabilitare un appartamento nel Sistema	RF90-OB	S
TS-90	Verificare che l'amministratore possa creare un allarme nel Sistema	RF91-OB	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-91	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare un appartamento	RF92-OB	S
TS-92	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare un sensore	RF93-OB	S
TS-93	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare un livello di priorità tra: <ul style="list-style-type: none"> • priorità 1 (bianco); • priorità 2 (verde); • priorità 3 (arancione); • priorità 4 (rosso) 	RF94-OB	S
TS-94	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare una soglia di scatto	RF95-OB	S
TS-95	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare un orario di attivazione	RF96-OB	S
TS-96	Verificare che l'amministratore, nella creazione di un allarme, possa selezionare un orario di disattivazione	RF97-OB	S
TS-97	Verificare che l'amministratore riceva un errore se durante la creazione di un allarme non ha selezionato alcun sensore	RF98-OB	S
TS-98	Verificare che l'amministratore riceva un errore se durante la creazione di un allarme non ha selezionato alcun livello di priorità	RF99-OB	S
TS-99	Verificare che l'amministratore riceva un errore se durante la creazione di un allarme non ha selezionato alcuna soglia di scatto	RF100-OB	S
TS-100	Verificare che l'amministratore possa modificare la priorità di un allarme del Sistema	RF101-OB	S
TS-101	Verificare che l'amministratore possa modificare la soglia di scatto di un allarme del Sistema	RF102-OB	S
TS-102	Verificare che l'amministratore possa modificare l'orario di attivazione di un allarme del Sistema	RF103-OB	S
TS-103	Verificare che l'amministratore possa modificare l'orario di disattivazione di un allarme del Sistema	RF104-OB	S
TS-104	Verificare che l'amministratore possa abilitare un allarme nel Sistema	RF105-OB	S
TS-105	Verificare che l'amministratore possa disabilitare un allarme nel Sistema	RF106-OB	S
TS-106	Verificare che l'amministratore possa eliminare un allarme dal Sistema	RF107-OB	S
TS-107	Verificare che l'amministratore possa visualizzare le notifiche inviate dal Sistema	RF108-OP	S
TS-108	Verificare che l'amministratore, visualizzando le notifiche inviate dal Sistema, possa visualizzare una notifica nel dettaglio	RF109-OP	S
TS-109	Verificare che l'amministratore, visualizzando una notifica nel dettaglio possa visualizzarne il titolo	RF110-OP	S

Codice	Descrizione	Requisito di riferimento	Stato del test
TS-110	Verificare che l'amministratore, visualizzando una notifica nel dettaglio possa visualizzarne il tempo trascorso dall'invio	RF111-OP	S

Tabella 13: Tabella dei test di sistema

4.2 Test di accettazione

Codice	Descrizione	Stato del test
TA-1	Verificare che il prodotto dia la possibilità all'utente non autenticato di essere riconosciuto tramite username e password	S
TA-2	Verificare che il prodotto dia la possibilità all'utente non registrato di autenticarsi tramite username e password temporanea, forniti dall'amministratore	S
TA-3	Verificare che il prodotto dia la possibilità, all'utente autenticato tramite password temporanea, di cambiare password	S
TA-4	Verificare che il prodotto dia la possibilità all'utente amministratore di gestire l'account MyVimar collegato all'impianto	S
TA-5	Verificare che il prodotto dia la possibilità all'utente amministratore di visualizzare l'elenco degli utenti	S
TA-6	Verificare che il prodotto dia la possibilità all'utente amministratore di creare utenti Operatore Sanitario	S
TA-7	Verificare che il prodotto dia la possibilità all'utente amministratore di eliminare utenti Operatore Sanitario	S
TA-8	Verificare che il prodotto dia la possibilità all'utente amministratore di generare una password temporanea per gli utenti Operatore Sanitario	S
TA-9	Verificare che il prodotto dia la possibilità all'utente di visualizzare la lista dei reparti	S
TA-10	Verificare che il prodotto dia la possibilità all'amministratore di creare un reparto	S
TA-11	Verificare che il prodotto dia la possibilità all'amministratore di eliminare un reparto	S
TA-12	Verificare che il prodotto dia la possibilità all'amministratore di assegnare un reparto ad un utente Operatore Sanitario	S
TA-13	Verificare che il prodotto dia la possibilità all'amministratore di rimuovere l'assegnazione di un reparto ad un utente Operatore Sanitario	S
TA-14	Verificare che il prodotto dia la possibilità all'amministratore di assegnare un reparto ad un appartamento	S
TA-15	Verificare che il prodotto dia la possibilità all'amministratore di rimuovere l'assegnazione di un reparto ad un appartamento	S
TA-16	Verificare che il prodotto dia la possibilità all'utente di visualizzare la dashboard	S
TA-17	Verificare che il prodotto dia la possibilità all'utente di visualizzare il modulo gestione allarmi nella dashboard	S
TA-18	Verificare che il prodotto dia la possibilità all'utente di visualizzare il modulo statistiche allarmi nella dashboard	S

Codice	Descrizione	Stato del test
TA-19	Verificare che il prodotto dia la possibilità all'amministratore di aggiungere il modulo informazioni utente alla dashboard	NI
TA-20	Verificare che il prodotto dia la possibilità all'amministratore di aggiungere il modulo analisi clima alla dashboard	NI
TA-21	Verificare che il prodotto dia la possibilità all'amministratore di aggiungere il modulo analisi consumi alla dashboard	NI
TA-22	Verificare che il prodotto dia la possibilità all'amministratore di rimuovere il modulo informazioni utente dalla dashboard	NI
TA-23	Verificare che il prodotto dia la possibilità all'amministratore di rimuovere il modulo analisi clima dalla dashboard	NI
TA-24	Verificare che il prodotto dia la possibilità all'amministratore di rimuovere il modulo analisi consumi dalla dashboard	NI
TA-25	Verificare che il prodotto dia la possibilità all'utente di visualizzare il modulo informazioni utente nella dashboard	S
TA-26	Verificare che il prodotto dia la possibilità all'utente di visualizzare il modulo analisi clima nella dashboard	S
TA-27	Verificare che il prodotto dia la possibilità all'utente di visualizzare il modulo analisi consumi nella dashboard	S
TA-28	Verificare che il prodotto dia la possibilità all'utente di risolvere un allarme	S
TA-29	Verificare che il prodotto dia la possibilità all'utente di visualizzare le analytics	S
TA-30	Verificare che il prodotto dia la possibilità all'utente di visualizzare i consigli di risparmio energetico	S
TA-31	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato al consumo energetico	S
TA-32	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato alle anomalie dell'impianto	S
TA-33	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato al rilevamento presenze	S
TA-34	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato alla presenza prolungata nello stesso ambiente	S
TA-35	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato alle variazioni di temperatura	S
TA-36	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato agli allarmi inviati e risolti	S
TA-37	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato alla frequenza degli allarmi	S
TA-38	Verificare che il prodotto dia la possibilità all'utente di visualizzare il grafico dedicato alla frequenza delle cadute	S
TA-39	Verificare che il prodotto dia la possibilità all'utente di visualizzare un appartamento	S
TA-40	Verificare che il prodotto dia la possibilità all'utente di visualizzare la mappa degli allarmi di un appartamento	S
TA-41	Verificare che il prodotto dia la possibilità all'utente di visualizzare l'elenco di stanze di un appartamento	S
TA-42	Verificare che il prodotto dia la possibilità all'utente di visualizzare l'elenco dei dispositivi di una stanza	S
TA-43	Verificare che il prodotto dia la possibilità all'amministratore di abilitare un appartamento	S

Codice	Descrizione	Stato del test
TA-44	Verificare che il prodotto dia la possibilità all'amministratore di disabilitare un appartamento	S
TA-45	Verificare che il prodotto dia la possibilità all'amministratore di creare un allarme	S
TA-46	Verificare che il prodotto dia la possibilità all'amministratore di modificare un allarme	S
TA-47	Verificare che il prodotto dia la possibilità all'amministratore di abilitare un allarme	S
TA-48	Verificare che il prodotto dia la possibilità all'amministratore di disabilitare un allarme	S
TA-49	Verificare che il prodotto dia la possibilità all'utente di ricevere notifiche dal Sistema	S
TA-50	Verificare che il prodotto dia la possibilità all'utente di visualizzare le notifiche ricevute dal Sistema	S

Tabella 14: Tabella dei test di accettazione

4.3 Test di integrazione

Codice	Descrizione	Risultato atteso	Stato del test
TI-1	Contesto <i>BreadcrumbService</i> ; caso 'breadcrumb construction e navigation tracking': verifica che breadcrumbs\$ emette array corretto, re-emette on NavigationEnd, gestisce routes multi-level ignoring empty segments and building proper paths.	Esito atteso: breadcrumbs per routes annidate costruiti correttamente, no double-slash, multi-segment paths concatenati.	S
TI-2	Contesto <i>InternalAuthService</i> ; consolidated summary covering login flows, session management, refresh token handling, concurrent restore, logout and user stream: verifica che il servizio gestisce login POST con credenziali, esegue first-login con temp password, aggiorna token via refresh, ripristina sessione dai cookie, evita richieste refresh duplicate con operazioni concorrenti, resetta stato al logout, espone stream utente. Tutti i flussi richiedono withCredentials true e metodo POST.	Esito atteso: login imposta session.username e token; first-login accetta temp password e nuova password; refresh aggiorna accessToken; restore concorrente single-request; logout pulisce token/role; currentUser inizialmente null; tutti gli endpoint con POST e withCredentials true; logout chiama /auth/logout.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-3	Contesto <i>AnalyticsComponent</i> ; consolidated summary covering component creation, apartment loading, chart fetching and metric retrieval: verifica che il componente sia valorizzato, carica il primo appartamento di default invocando <code>getAllApartments</code> e <code>getAnalytics('1')</code> , aggiorna grafici al cambio appartamento con <code>getAnalytics('2')</code> , restituisce undefined se metrica non trovata, trova la metrica corretta con <code>getChartByMetric('plant-anomalies')</code> .	Esito atteso: component valorizzato; <code>mockAnalyticsService.getAllApartments</code> invocato; <code>mockAnalyticsService.getAnalytics</code> invocato con '1' al caricamento e con '2' al cambio; <code>getChartByMetric('plant-anomalies')</code> restituisce metrica con <code>metric:'plant-anomalies'</code> ; <code>getChartByMetric('unknown')</code> restituisce undefined; feature tests RF61-RF62-RF63 verificano stesso flusso con stub API.	S
TI-4	Contesto <i>MainLayoutComponent</i> ; caso 'carica navItems, gestisce MyVimar e logout': verifica che <code>mockNavService.getNavItems</code> viene invocato e <code>component.navItems</code> è caricato; mostra warning MyVimar per admin; logout reindirizza a <code>/auth/login</code> .	Esito atteso: <code>mockNavService.getNavItems</code> viene invocato; <code>component.navItems.length</code> è maggiore di 0; il valore di <code>component.showVimarAssociationWarning</code> è appropriato; <code>navigateSpy</code> viene invocato con <code>['/auth/login']</code> al logout.	S
TI-5	Contesto <i>MainLayoutComponent</i> ; caso 'reindirizza al login senza sessione e apre pannello notifiche': verifica che senza sessione utente reindirizza a <code>/auth/login</code> ; le nuove notifiche realtime aprono il pannello.	Esito atteso: <code>component.navItems</code> coincide con <code>[]</code> ; <code>navigateSpy</code> viene invocato con <code>['/auth/login']</code> ; il valore di <code>component.isNotificationPanelOpen</code> diventa true al ricevimento notifica.	S
TI-6	Contesto <i>MainLayout</i> feature integration; caso 'RF46-RF47-RF48: carica navItems da ruolo utente, gestisce login redirect e notifiche realtime': verifica che <code>navServiceStub.getNavItems</code> viene invocato con <code>UserRole.AMMINISTRATORE</code> , mostra <code>navItems.length > 0</code> e warning MyVimar, reindirizza a <code>/auth/login</code> senza sessione, apre pannello notifiche su evento realtime.	Esito atteso: <code>navServiceStub.getNavItems</code> invocato con admin role; <code>component.navItems > 0</code> ; <code>showVimarAssociationWarning</code> true per admin; <code>navigateSpy</code> con <code>['/auth/login']</code> senza sessione; <code>isNotificationPanelOpen</code> toggled true su notifica realtime.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-7	Contesto device-taxonomy; consolidated summary covering device type resolution, endpoint labeling and dynamic discovery: verifica che il servizio risolve type thermostat esplicitamente e da endpoint sfeType ambiguo, inferisce entrance door da nome italiano, scenario devices da nome, fallback unknown; getEndpointLabel restituisce label leggibile per SFE_Cmd_* noti, fallback humanized per sconosciuti, tracks runtime discovery in getDiscoveredUnmappedSfeCodes, restituisce label dispositivo leggibile.	Esito atteso: thermostat resolved explicit e inferito; entrance_door da nome italiano; alarm_button da scenario; unknown fallback; getEndpointLabel('SFE_Cmd_ChangeOverMode')='Comando cambio modalità HVAC'; unknown SFE code fallback umanizzato; getDeviceTypeLabel(PRESENCE_SENSOR)='Sensore presenza'; unknown type fallback.	S
TI-8	Contesto <i>ElapsedTimePipe</i> ; consolidated summary covering time formatting for past and future dates in compact format: verifica che la pipe ritorna valore originale per date non valide, mostra 'tra poco' per timestamp futuri, restituisce formato compatto secondi/minuti/ore/giorni per intervalli passati con suffisso 'fa'.	Esito atteso: input invalido ritorna originale; timestamp futuro='tra poco'; 30s fa < 60s; 1m fa 60-3600s; 1h fa 3600-86400s; 1g fa >= 86400s; 3h fa e 1h 12m fa con ore/minuti; 5g fa per giorni.	S
TI-9	Contesto notification-title util integration; consolidated summary covering fallback handling, priority mapping, enum validation and truncation: verifica che la funzione restituisce fallback per nomi non-string o vuoti, normalizza spazi, mappa priorità WHITE/GREEN/ORANGE/RED a simboli unicode, accetta priorità enum string case-insensitive, restituisce fallback per priorità sconosciute, tronca e appende ellipsis se eccede maxLength.	Esito atteso: fallback per undefined/null/empty/invalid names; priority mapping WHITE=i, GREEN=bullet, ORANGE=!, RED=triangle; enum string e numeric supported; truncation con ellipsis; no error on small maxLength.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-10	Contesto <i>NotificationApiService</i> ; consolidated summary for notification history fetching covering managed/unmanaged endpoints, cold observable behavior, session validation: verifica che il servizio chiama GET /managed e /unmanaged endpoints per storia notifiche, mappa payload verso NotificationEvent[], emette una volta e completa (non buffered), cold observable produce coppie di request per subscriber, restituisce [] se no sessione attiva senza HTTP calls.	Esito atteso: managed/unmanaged GET invocati; result mappato; emissions=1 completed=true; cold observable behavior confirm; no session=no HTTP.	S
TI-11	Contesto <i>AlarmApiService</i> ; consolidated summary for comprehensive CRUD and event management covering rules, alarms, resolution tracking: verifica che il servizio supporta GET /alarm-rules e GET /alarm-rules/:id, POST /alarm-rules, PUT /alarm-rules/:id, DELETE /alarm-rules/:id, GET /alarm-events/unmanaged e GET /alarm-events/managed per allarmi attivi/risolti con pagination (userId/limit/offset), GET /alarm-events/:id per dettaglio, PATCH /alarm-events/resolve per gestione. Input validation strict su numeric params, charset speciali handled in ID encoding, empty list handling e HTTP error propagation.	Esito atteso: tutti gli endpoint supportati; GET/POST/PUT/DELETE/PATCH corretti; numeric validation userId/limit/offset; special chars in ID; empty list no error; HTTP errors propagated.	S
TI-12	Contesto <i>AlarmManagementRefreshService</i> ; consolidated summary for hot observable refresh request management: verifica che il servizio espone observable hot per requestRefresh(), emette su richiesta esplicita, non bufferizza refresh precedenti per nuovi subscriber (hot behavior), refreshPromise resettato post-resolve.	Esito atteso: hot observable behavior; refreshPromise cleared after resolve; single request per subscriber; no previous emissions buffering.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-13	Contesto <i>AlarmStateService</i> ; consolidated summary for realtime alarm and notification state management covering initialization, add/update/remove operations, snapshot handling, unread counts: verifica che il servizio inzializza stato vuoto (alarms=[], notifications=[]), aggiunge/aggiorna allarmi via onAlarmTriggered preservando ID, rimuove via onAlarmResolved, antepone notifiche (più recente first), deduplica per notificationId, removeNotification e clearNotifications, setActiveAlarms imposta snapshot, emette conteggi activeAlarm-sCount/unreadNotificationsCount.	Esito atteso: init empty state; add/update preserve, no duplication; remove target only; notifications prepended; deduplicate by ID; snapshot handling merge; counters emitted correctly.	S
TI-14	Contesto <i>EventSubscriptionService</i> ; consolidated summary part 1 - socket initialization and connection management: verifica che il servizio crea connessione socket via socketIoFactory su API_BASE_URL normalizzato (rtrim '/api' + '/ws'), risolve namespace ws correttamente, aggiorna stato ConnectionStatus su connect/reconnect/disconnect, riesegue join room precedenti al reconnect, supporta room join/leave con deduplicazione.	Esito atteso: socket created on init; API_BASE_URL normalized; ws namespace resolved; connection status updated; rejoin on reconnect; dedup join/leave.	S
TI-15	Contesto <i>EventSubscriptionService</i> ; consolidated summary part 2 - realtime events bootstrap and error handling: verifica che il servizio bootstrap room da plant/all al login se cache ward vuoto, evita bootstrap ripetuto stessa sessione, non interrompe realtime se bootstrap fallisce (fallback), riallinea room subscription senza page reload, ignora malformed events, disconnette socket alla destroy, su logout emette leave room.	Esito atteso: bootstrap plant/all on login; single bootstrap per session; no error break; refresh ward subscription; malformed ignored; cleanup on destroy; logout emits leave.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-16	Contesto <i>EventSubscriptionService</i> ; caso 'crea la connessione socket in initialize usando API_BASE_URL normalizzato': verifica che socketIoFactoryMock viene invocato con 'http://api.example.local/ws'.	Esito atteso: socketIoFactoryMock viene invocato con 'http://api.example.local/ws'.	S
TI-17	Contesto <i>EventSubscriptionService</i> ; caso 'risolve il namespace websocket ws quando API_BASE_URL termina con api': verifica che socketIoFactoryMock viene invocato con 'http://localhost/ws'.	Esito atteso: socketIoFactoryMock viene invocato con 'http://localhost/ws'.	S
TI-18	Contesto <i>EventSubscriptionService</i> ; caso 'aggiorna lo stato connessione su connect, reconnect attempt e disconnect': verifica che statuses coincide con [ConnectionStatus.DISCONNECTED, ConnectionStatus.CONNECTED, ConnectionStatus.RECONNECTING, Co....	Esito atteso: statuses coincide con [ConnectionStatus.DISCONNECTED, ConnectionStatus.CONNECTED, ConnectionStatus.RECONNECTING, Co....	S
TI-19	Contesto <i>EventSubscriptionService</i> ; caso 'esegue join e leave delle room con deduplicazione': verifica che fakeSocket.emit viene invocato con 'join-ward' e fakeSocket.emit viene invocato con 'leave-ward'.	Esito atteso: fakeSocket.emit viene invocato con 'join-ward'; fakeSocket.emit viene invocato con 'leave-ward'; il valore di joinWardCalls.length è 2; il valore di leaveWardCalls.length è 1.	S
TI-20	Contesto <i>EventSubscriptionService</i> ; caso 'riesegue il join delle room già sottoscritte al connect': verifica che il valore di joinCallsBeforeReconnect è 1 e il valore di joinCallsAfterReconnect.length è 2.	Esito atteso: il valore di joinCallsBeforeReconnect è 1; il valore di joinCallsAfterReconnect.length è 2; joinCallsAfterReconnect.at(-1) coincide con ['join-ward', 'ward-a'].	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-21	Contesto <i>EventSubscriptionService</i> ; caso 'bootstrap delle room da plant all al login quando cache ward è vuota': verifica che <code>httpClientSpy.get</code> viene invocato con <code>'http://api.example.local/plant/all'</code> e <code>joinWardCalls</code> coincide con <code>[['join-ward', '21'], ['join-ward', '22'],]</code> .	Esito atteso: <code>httpClientSpy.get</code> viene invocato con <code>'http://api.example.local/plant/all'</code> ; <code>joinWardCalls</code> coincide con <code>[['join-ward', '21'], ['join-ward', '22'],]</code> .	S
TI-22	Contesto <i>EventSubscriptionService</i> ; caso 'non interrompe il realtime quando bootstrap plant all fallisce': verifica che <code>httpClientSpy.get</code> viene invocato 1 volta e <code>alarmStateSpy.onAlarmTriggered</code> viene invocato con <code>expect.objectContaining({ id: 'active-alarm-bootstrap-fallback', })</code> .	Esito atteso: <code>httpClientSpy.get</code> viene invocato 1 volta; <code>alarmStateSpy.onAlarmTriggered</code> viene invocato con <code>expect.objectContaining({ id: 'active-alarm-bootstrap-fallback', })</code> ; <code>alarmStateSpy.onNotificationReceived</code> viene invocato con <code>expect.objectContaining({ notificationId: 'alarm-triggered-active-alarm-bootstrap-fallback', })</code> .	S
TI-23	Contesto <i>EventSubscriptionService</i> ; caso 'evita bootstrap ripetuto per lo stesso utente nella stessa sessione': verifica che <code>httpClientSpy.get</code> viene invocato 1 volta.	Esito atteso: <code>httpClientSpy.get</code> viene invocato 1 volta.	S
TI-24	Contesto <i>EventSubscriptionService</i> ; caso ' <i>refreshWardRoomSubscription</i> riallinea le room senza reload pagina': verifica che <code>httpClientSpy.get</code> viene invocato 2 volte e <code>fakeSocket.emit</code> viene invocato con 'join-ward'.	Esito atteso: <code>httpClientSpy.get</code> viene invocato 2 volte; <code>fakeSocket.emit</code> viene invocato con 'join-ward'; <code>fakeSocket.emit</code> viene invocato con 'leave-ward'.	S
TI-25	Contesto <i>EventSubscriptionService</i> ; caso 'ignora eventi raw malformati e payload backend non validi': verifica che <code>alarmStateSpy.onAlarmTriggered</code> non viene invocato e <code>alarmStateSpy.onAlarmResolved</code> non viene invocato.	Esito atteso: <code>alarmStateSpy.onAlarmTriggered</code> non viene invocato; <code>alarmStateSpy.onAlarmResolved</code> non viene invocato; <code>alarmStateSpy.onNotificationReceived</code> non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-26	Contesto <i>EventSubscriptionService</i> ; caso 'usa location.origin quando API_BASE_URL non è configurato': verifica che socketIoFactoryMock viene invocato con '\$global-This.location.origin}/ws'.	Esito atteso: socketIoFactoryMock viene invocato con '\$global-This.location.origin}/ws'.	S
TI-27	Contesto <i>EventSubscriptionService</i> ; caso 'bootstrap accetta risposta wrappata in data[]': verifica che joinWardCalls coincide con [['join-ward', '31'], ['join-ward', '32'],].	Esito atteso: joinWardCalls coincide con [['join-ward', '31'], ['join-ward', '32'],].	S
TI-28	Contesto <i>EventSubscriptionService</i> ; caso 'bootstrap accetta risposta wrappata in plants[]': verifica che joinWardCalls coincide con [['join-ward', '41'], ['join-ward', '42'],].	Esito atteso: joinWardCalls coincide con [['join-ward', '41'], ['join-ward', '42'],].	S
TI-29	Contesto <i>EventSubscriptionService</i> ; caso 'su logout emette leave delle room ': verifica che fakeSocket.emit viene invocato con 'leave-ward'.	Esito atteso: fakeSocket.emit viene invocato con 'leave-ward'.	S
TI-30	Contesto <i>EventSubscriptionService</i> ; caso 'disconnette il socket e rimuove i listener alla distruzione': verifica che fakeSocket.removeAllListeners viene invocato e fakeSocket.disconnect viene invocato.	Esito atteso: fakeSocket.removeAllListeners viene invocato; fakeSocket.disconnect viene invocato.	S
TI-31	Contesto <i>RealtimeAlarmEventNormalizerService</i> ; consolidated summary for backend payload normalization: verifica che parseBackendTriggeredPayload estrae alarmEventId/wardId/alarmRuleId da payload nativo, accetta wardId numerico espresso come stringa, parseBackendResolvedPayload estrae alarmEventId e wardId opzionale.	Esito atteso: triggered payload parsed correctly; wardId numerico string accepted; resolved payload extracted; optional wardId handled.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-32	Contesto <i>WardRealtimeCacheService</i> ; consolidated summary for per-user ward cache management: verifica che <code>mergeWardIds</code> deduplica e normalizza valori stringa/numero, mantiene cache separata per utente, <code>clearWardIds</code> rimuove solo utente target, recupera cache vuota per JSON non valido.	Esito atteso: dedup and normalize string/number; separate cache per user; selective clear; fallback empty cache.	S
TI-33	Contesto <i>WardSocketRoomCoordinatorService</i> ; consolidated summary for room subscription lifecycle management: verifica che <code>activateUser</code> carica <code>wardIds</code> da cache e propone join, <code>requestJoinRoom</code> deduplica e persiste su cache, <code>deactivateUser</code> rimuove room runtime e pulisce cache, <code>switch</code> utente restituisce stanza da lasciare e cache del nuovo utente.	Esito atteso: activate loads and proposes; join deduplicated and cached; deactivate cleans; switch manages transitions.	S
TI-34	Contesto <i>AlarmConfigFormFieldOrchestratorHelper</i> ; consolidated summary for form field orchestration in edit/create mode: verifica che <code>applyModeState</code> in edit disabilita campi non modificabili (<code>name/plantId/deviceId/datapointId</code>), <code>resolveDeviceSelection</code> ritorna datapoint leggibili solo e auto-selezione se singolo, <code>resolveDatapointSelection</code> normalizza id e restituisce datapoint atteso.	Esito atteso: edit mode disabled fields correct; readable datapoints only; auto-select on single; ID normalized.	S
TI-35	Contesto <i>AlarmConfigFormValidationHelper</i> ; consolidated summary for form validation in create and edit modes with enum and numeric datapoint types: verifica che create mode con enum forza operatore uguale e valida threshold su enum, con numerico applica validazione numerica pattern, edit mode applica stesso vincolo operatore enum, validazione numerica, senza metadati mantiene solo operatore corrente.	Esito atteso: create enum forces = operator, numeric pattern validation; edit mode same; unsupported operator error.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-36	Contesto <i>AlarmRuleFormMapper</i> ; consolidated summary for form value mapping from API model to form representation: verifica che il mapper mappa operatori >= come GREATER_THAN_OR_EQUAL e <= come LESS_THAN_OR_EQUAL, fallisce su operatori sconosciuti.	Esito atteso: >= mapped GREATER_THAN_OR_EQUAL; <= mapped LESS_THAN_OR_EQUAL; unknown operators throw.	S
TI-37	Contesto <i>AlarmRuleRequestMapper</i> ; consolidated summary for request payload serialization covering create, update and toggle operations: verifica che toUpdateRequest serializza operatore >= senza degradarlo, toCreateRequest serializza e normalizza soglia booleana in on/off, fallisce se thresholdValue vuoto o thresholdOperator null, toToggleRequest fallisce se name vuoto e normalizza operatori con padding dal backend.	Esito atteso: >= preserved in serialize; <= preserved; boolean <-> on/off normalization; empty/null validation; padding normalization.	S
TI-38	Contesto <i>AlarmConfigFormStateService</i> ; caso 'ensurePlantsLoaded in create mode carica e deduplica gli impianti': verifica che service.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3'] e service.plantsLoadError() è null.	Esito atteso: service.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3']; service.plantsLoadError() è null.	S
TI-39	Contesto <i>AlarmConfigFormStateService</i> ; caso 'ensurePlantsLoaded in edit mode non esegue chiamate': verifica che wardApiStub.getAvailablePlants non viene invocato e wardApiStub.getWards non viene invocato.	Esito atteso: wardApiStub.getAvailablePlants non viene invocato; wardApiStub.getWards non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-40	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>loadDeviceOptionsByPlant</i> ritorna opzioni dispositivi e pulisce loading state': verifica che <i>apartmentApiStub.loadApartmentViewForPlantId</i> viene invocato con 'plant-1' e result coincide con options.	Esito atteso: <i>apartmentApiStub.loadApartmentViewForPlantId</i> viene invocato con 'plant-1'; result coincide con options; il valore di <i>service.isDevicesLoading()</i> è false; <i>service.devicesLoadError()</i> è null.	S
TI-41	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>loadDeviceOptionsByPlant</i> in errore restituisce lista vuota e imposta messaggio': verifica che result coincide con [] e il valore di <i>service.isDevicesLoading()</i> è false.	Esito atteso: result coincide con []; il valore di <i>service.isDevicesLoading()</i> è false; il valore di <i>service.devicesLoadError()</i> è 'Errore durante il caricamento dei dispositivi.'	S
TI-42	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>resolveDatapointForEdit</i> usa il catalogo impianti e restituisce il datapoint trovato': verifica che <i>apartmentApiStub.getAllPlants</i> viene invocato 1 volta e <i>datapointExtractionStub.findDatapointByDeviceAndDatapointId</i> viene invocato con [{ id: 'plant-1', name: 'Apt', rooms: [] }].	Esito atteso: <i>apartmentApiStub.getAllPlants</i> viene invocato 1 volta; <i>datapointExtractionStub.findDatapointByDeviceAndDatapointId</i> viene invocato con [{ id: 'plant-1', name: 'Apt', rooms: [] }]; result coincide con <i>expectedDatapoint</i> .	S
TI-43	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>resolveDatapointForEdit</i> ritorna null se il catalogo impianti fallisce': verifica che result è null.	Esito atteso: result è null.	S
TI-44	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>inizializza stato con alarms vuoti e nessun errore</i> ': verifica che <i>await firstValueFrom(service.alarms\$)</i> coincide con [] e <i>await firstValueFrom(service.error\$)</i> è null.	Esito atteso: <i>await firstValueFrom(service.alarms\$)</i> coincide con []; <i>await firstValueFrom(service.error\$)</i> è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-45	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>loadAlarmRules</i> aggiorna stato locale quando API risponde con successo': verifica che <i>apiStub.getAlarmRules</i> viene invocato 1 volta e <i>alarms</i> coincide con [alarmA, alarmB].	Esito atteso: <i>apiStub.getAlarmRules</i> viene invocato 1 volta; <i>alarms</i> coincide con [alarmA, alarmB]; <i>alarms</i> ha lunghezza 2; <i>await firstValueFrom(service.error\$)</i> è null.	S
TI-46	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>loadAlarmRules</i> espone messaggio errore e pulisce <i>error</i> precedente al <i>retry</i> ': verifica che il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante il caricamento degli allarmi.' e <i>await firstValueFrom(service.error\$)</i> è null.	Esito atteso: il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante il caricamento degli allarmi.'; <i>await firstValueFrom(service.error\$)</i> è null; <i>await firstValueFrom(service.alarms\$)</i> coincide con [alarmA].	S
TI-47	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>getAlarmRuleById</i> inoltra il valore di API in caso di successo': verifica che <i>apiStub.getAlarmRule</i> viene invocato con 'alarm-1' e <i>result</i> coincide con alarmA.	Esito atteso: <i>apiStub.getAlarmRule</i> viene invocato con 'alarm-1'; <i>result</i> coincide con alarmA.	S
TI-48	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>getAlarmRuleById</i> in errore imposta messaggio e completa senza emissioni': verifica che il valore di <i>emitted</i> è false e il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante il recupero dell\{\}'allarme?.	Esito atteso: il valore di <i>emitted</i> è false; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante il recupero dell\{\}'allarme?.	S
TI-49	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>createAlarmRule</i> con form invalido non chiama API e imposta errore di validazione': verifica che il valore di <i>emitted</i> è false e <i>apiStub.createAlarmRule</i> non viene invocato.	Esito atteso: il valore di <i>emitted</i> è false; <i>apiStub.createAlarmRule</i> non viene invocato; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme?.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-50	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>createAlarmRule</i> in errore API imposta messaggio e non altera lo stato locale': verifica che il valore di <i>emitted</i> è false e il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante la creazione dell\{\}'allarme.'	Esito atteso: il valore di <i>emitted</i> è false; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante la creazione dell\{\}'allarme.'; <i>await firstValueFrom(service.alarms\$)</i> coincide con [alarmA]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TI-51	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>createAlarmRule</i> pulisce l'errore precedente quando una creazione successiva va a buon fine': verifica che il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme.' e <i>await firstValueFrom(service.error\$)</i> è null.	Esito atteso: il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme.'; <i>await firstValueFrom(service.error\$)</i> è null; <i>await firstValueFrom(service.alarms\$)</i> coincide con [alarmB].	S
TI-52	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>updateAlarmRule</i> mantiene il nome originale nel payload update e sostituisce l'allarme nello stato': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1' e <i>await firstValueFrom(service.alarms\$)</i> coincide con [updatedAlarm, alarmB].	Esito atteso: <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1'; <i>await firstValueFrom(service.alarms\$)</i> coincide con [updatedAlarm, alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TI-53	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>updateAlarmRule</i> con form invalido non chiama API e imposta errore': verifica che il valore di <i>emitted</i> è false e <i>apiStub.updateAlarmRule</i> non viene invocato.	Esito atteso: il valore di <i>emitted</i> è false; <i>apiStub.updateAlarmRule</i> non viene invocato; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per l\{\}'aggiornamento dell\{\}'allarme.'	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-54	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l'aggiornamento dell'allarme.'.	Esito atteso: il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l'aggiornamento dell'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA, alarmB]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TI-55	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> senza allarme locale usa il nome del form nel payload': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'missing-id'.	Esito atteso: <i>apiStub.updateAlarmRule</i> viene invocato con 'missing-id'.	S
TI-56	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> sostituisce la riga locale anche se il backend ritorna un nuovo id': verifica che alarms coincide con [updatedWithNewId, alarmB] e il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false.	Esito atteso: alarms coincide con [updatedWithNewId, alarmB]; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1-v2')</code> è true; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TI-57	Contesto <i>AlarmConfigStateService</i> ; caso <i>'toggleEnabled</i> aggiorna enabled costruendo payload dai dati locali': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1' e await <code>firstValueFrom(service.alarms\$)</code> coincide con [toggledAlarm].	Esito atteso: <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [toggledAlarm]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TI-58	Contesto <i>AlarmConfigStateService</i> ; caso <i>'toggleEnabled</i> su allarme assente non chiama API e imposta errore': verifica che il valore di emitted è false e <i>apiStub.updateAlarmRule</i> non viene invocato.	Esito atteso: il valore di emitted è false; <i>apiStub.updateAlarmRule</i> non viene invocato; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Allarme non trovato nello stato locale.'; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-59	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>toggleEnabled</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante la modifica dello stato dell\{\}'allarme.'.	Esito atteso: il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante la modifica dello stato dell\{\}'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TI-60	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>toggleEnabled</i> aggiorna subito lo stato locale anche se il backend ritorna un nuovo id': verifica che alarms coincide con [toggledWithNewId, alarmB] e il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false.	Esito atteso: alarms coincide con [toggledWithNewId, alarmB]; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1-v2')</code> è true.	S
TI-61	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> rimuove l'allarme locale quando API conferma': verifica che il valore di emitted è true e <i>apiStub.deleteAlarmRule</i> viene invocato con 'alarm-1'.	Esito atteso: il valore di emitted è true; <i>apiStub.deleteAlarmRule</i> viene invocato con 'alarm-1'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TI-62	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'.	Esito atteso: il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA, alarmB]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TI-63	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> pulisce errore precedente quando un delete successivo riesce': verifica che il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.' e await <code>firstValueFrom(service.error\$)</code> è null.	Esito atteso: il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'; await <code>firstValueFrom(service.error\$)</code> è null; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-64	Contesto <i>DeviceDatapointExtractionService</i> ; caso <i>'extractDeviceOptions crea label stanza-dispositivo e normalizza enum duplicati'</i> : verifica che options ha lunghezza 1 e il valore di options[0].id è 'dev-1'.	Esito atteso: options ha lunghezza 1; il valore di options[0].id è 'dev-1'; il valore di options[0].label è 'Soggiorno - Sensore UWB'; options[0].datapoints[0].enum coincide con ['Off', 'On'].	S
TI-65	Contesto <i>DeviceDatapointExtractionService</i> ; caso <i>'findReadableDatapoints ritorna solo datapoint leggibili'</i> : verifica che datapoints.map((datapoint) => datapoint.id) coincide con ['dp-1', 'dp-2'] e il valore di datapoints.every((datapoint) => datapoint.readable) è true.	Esito atteso: datapoints.map((datapoint) => datapoint.id) coincide con ['dp-1', 'dp-2']; il valore di datapoints.every((datapoint) => datapoint.readable) è true.	S
TI-66	Contesto <i>DeviceDatapointExtractionService</i> ; caso <i>'findReadableDatapoints ritorna lista vuota se il device non esiste'</i> : verifica che datapoints coincide con [].	Esito atteso: datapoints coincide con [].	S
TI-67	Contesto <i>DeviceDatapointExtractionService</i> ; caso <i>'getAllowedOperators limita a uguale quando il datapoint ha enum'</i> : verifica che <i>service.getAllowedOperators(enumDatapoint)</i> coincide con [ThresholdOperator.EQUAL_TO] e <i>service.getAllowedOperators(numericDatapoint)</i> coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	Esito atteso: <i>service.getAllowedOperators(enumDatapoint)</i> coincide con [ThresholdOperator.EQUAL_TO]; <i>service.getAllowedOperators(numericDatapoint)</i> coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	S
TI-68	Contesto <i>DeviceDatapointExtractionService</i> ; caso <i>'isThresholdValueValid valida enum case-insensitive e soglie numeriche'</i> : verifica che il valore di <i>service.isThresholdValueValid(enumDatapoint, 'on')</i> è true e il valore di <i>service.isThresholdValueValid(enumDatapoint, 'Unknown')</i> è false.	Esito atteso: il valore di <i>service.isThresholdValueValid(enumDatapoint, 'on')</i> è true; il valore di <i>service.isThresholdValueValid(enumDatapoint, 'Unknown')</i> è false; il valore di <i>service.isThresholdValueValid(numericDatapoint, 'on')</i> è true; il valore di <i>service.isThresholdValueValid(numericDatapoint, 'Unknown')</i> è false.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-69	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findDatapointByDeviceAndDatapointId trova e normalizza il datapoint dal catalogo plant': verifica che datapoint non è null e il valore di datapoint?.id è 'dp-lookup'.	Esito atteso: datapoint non è null; il valore di datapoint?.id è 'dp-lookup'; datapoint?.enum coincide con ['Off', 'On'].	S
TI-70	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findDatapointByDeviceAndDatapointId ritorna null se non trova corrispondenze': verifica che datapoint è null.	Esito atteso: datapoint è null.	S
TI-71	Contesto AlarmHistory feature integration; caso 'inizializza pagina e service' : verifica che component è valorizzato e alarmHistoryServiceStub.initialize viene invocato 1 volta.	Esito atteso: component è valorizzato; alarmHistoryServiceStub.initialize viene invocato 1 volta.	S
TI-72	Contesto AlarmHistory feature integration; caso 'ordina righe per chiusura desc e applica fallback handler' : verifica che il valore di rows[0].id è 'a-2' e il valore di rows[0].handlerUsername è 'sconosciuto'.	Esito atteso: il valore di rows[0].id è 'a-2'; il valore di rows[0].handlerUsername è 'sconosciuto'; il valore di component.currentPage() è 2.	S
TI-73	Contesto AlarmHistory feature integration; caso 'delega paginazione next/previous al service' : verifica che alarmHistoryServiceStub.nextPage viene invocato 1 volta e alarmHistoryServiceStub.previousPage viene invocato 1 volta.	Esito atteso: alarmHistoryServiceStub.nextPage viene invocato 1 volta; alarmHistoryServiceStub.previousPage viene invocato 1 volta.	S
TI-74	Contesto <i>AlarmHistoryService</i> ; caso 'initialize carica la prima pagina con offset 0' : verifica che alarmApiStub.getResolvedAlarms viene invocato con 99.	Esito atteso: alarmApiStub.getResolvedAlarms viene invocato con 99.	S
TI-75	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con lista vuota non imposta canGoNext' : verifica che il valore di vm.alarms.length è 0 e vm.canGoNext è falsy.	Esito atteso: il valore di vm.alarms.length è 0; vm.canGoNext è falsy.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-76	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con pageLimit+1 elementi imposta canGoNext e fa lo slice': verifica che vm.canGoNext è valorizzato e il valore di vm.alarms.length è 6.	Esito atteso: vm.canGoNext è valorizzato; il valore di vm.alarms.length è 6.	S
TI-77	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con userId non numerico non chiama API': verifica che alarmApiStub.getResolvedAlarms non viene invocato.	Esito atteso: alarmApiStub.getResolvedAlarms non viene invocato.	S
TI-78	Contesto <i>AlarmHistoryService</i> ; caso 'initialize non propaga eccezioni al chiamante': verifica che () => service.initialize() non genera l'errore atteso.	Esito atteso: () => service.initialize() non genera l'errore atteso.	S
TI-79	Contesto <i>AlarmHistoryService</i> ; caso 'nextPage avanza alla pagina successiva': verifica che il valore di vm.currentPage è 2 e il valore di vm.pageOffset è 6.	Esito atteso: il valore di vm.currentPage è 2; il valore di vm.pageOffset è 6; vm.canGoPrevious è valorizzato; vm.alarms.map((a) => a.id) coincide con ['resolved-10'].	S
TI-80	Contesto <i>AlarmHistoryService</i> ; caso 'nextPage non fa nulla se canGoNext è false': verifica che alarmApiStub.getResolvedAlarms viene invocato 1 volta.	Esito atteso: alarmApiStub.getResolvedAlarms viene invocato 1 volta.	S
TI-81	Contesto <i>AlarmHistoryService</i> ; caso 'pagina successiva con meno di pageLimit elementi imposta canGoNext a false': verifica che vm.canGoNext è falsy e vm.alarms.map((a) => a.id) coincide con ['resolved-10', 'resolved-11'].	Esito atteso: vm.canGoNext è falsy; vm.alarms.map((a) => a.id) coincide con ['resolved-10', 'resolved-11'].	S
TI-82	Contesto <i>AlarmHistoryService</i> ; caso 'previousPage torna alla pagina precedente': verifica che il valore di vm.currentPage è 1 e il valore di vm.pageOffset è 0.	Esito atteso: il valore di vm.currentPage è 1; il valore di vm.pageOffset è 0; vm.canGoPrevious è falsy; l'asserzione alarmApiStub.getResolvedAlarms toHaveBeenNthCalledWith 1 risulta soddisfatta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-83	Nel contesto <i>AlarmHistoryService</i> , il caso 'previousPage non fa nulla se l'offset è già a zero' verifica che <i>alarmApiStub.getResolvedAlarms</i> viene invocato 1 volta.	Esito atteso: <i>alarmApiStub.getResolvedAlarms</i> viene invocato 1 volta.	S
TI-84	Contesto <i>AlarmManagementService</i> ; consolidated summary for RF44 initialization and pagination: verifica che initialize non carica automaticamente, admin role mostra tutti allarmi senza filtri, OSS usa endpoint unmanaged con userId, error handling con fallback messaggi, userId non-numeric blocks API call, nextPage incrementa offset con limit 6 quando pagina piena, previousPage decrementa fino a 0, pagina con <6 elementi imposta canGoNext=false, nextPage/previousPage non agiscono oltre limiti.	Esito atteso: lazy init on demand; admin no-filter GET; OSS unmanaged endpoint; errors mapped to vm state; userId validation; pagination with limit 6; offset tracking; flag management.	S
TI-85	Contesto <i>AlarmManagementService</i> ; consolidated summary for RF44 resolveAlarm and state management: verifica che resolveAlarm segue success flow (resolving true/false), VM reflects state transitions before/after resolve, local rows cleared e backend snapshot reload, automatic page down on empty non-first page, concurrent resolve requests tracked to completion, resolveError cleared on new resolve attempt, error path with network timeout, userId validation blocks resolve API.	Esito atteso: resolving state toggled; VM aligned to backend post-resolve; pagination auto-adjust; concurrent requests joined; error clear on retry; timeout handled; userId validated.	S
TI-86	Contesto <i>AnalyticsApiService</i> ; caso 'should be created': verifica che service è valorizzato.	Esito atteso: service è valorizzato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-87	Contesto <i>AnalyticsApiService</i> - <i>getAllApartments</i> ; caso 'should fetch all apartments and map them correctly' : verifica che il valore di <code>apartments.length</code> è 2 e <code>apartments[0]</code> coincide con <code>{ id: '1', name: 'Apartment 1' }</code> .	Esito atteso: il valore di <code>apartments.length</code> è 2; <code>apartments[0]</code> coincide con <code>{ id: '1', name: 'Apartment 1' }</code> ; l'asserzione <code>apartments[0].toHaveProperty('extraField')</code> risulta non soddisfatta; il valore di <code>req.request.method</code> è 'GET'.	S
TI-88	Contesto <i>AnalyticsApiService</i> - <i>getAnalytics</i> ; caso 'should fetch analytics and map the complex DTO structure' : verifica che il valore di <code>result.apartmentId</code> è <code>apartmentId</code> e il valore di <code>result.analyticsInfo.length</code> è 1.	Esito atteso: il valore di <code>result.apartmentId</code> è <code>apartmentId</code> ; il valore di <code>result.analyticsInfo.length</code> è 1; il valore di <code>info.title</code> è 'Test Chart'; <code>info.suggestions.messages</code> coincide con ['Suggestion 1'].	S
TI-89	Contesto <i>ApartmentApiService</i> ; caso 'RF83-OP abilita un appartamento nel Sistema' : verifica che il valore di <code>request.request.method</code> è 'PATCH' e <code>request.request.body</code> coincide con <code>{}</code> .	Esito atteso: il valore di <code>request.request.method</code> è 'PATCH'; <code>request.request.body</code> coincide con <code>{}</code> .	S
TI-90	Contesto <i>ApartmentApiService</i> ; caso 'RF84-OP disabilita un appartamento nel Sistema' : verifica che il valore di <code>request.request.method</code> è 'PATCH' e <code>request.request.body</code> coincide con <code>{}</code> .	Esito atteso: il valore di <code>request.request.method</code> è 'PATCH'; <code>request.request.body</code> coincide con <code>{}</code> .	S
TI-91	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>getAvailableApartments</i> ordina per nome e deduplica plant' : verifica che il valore di <code>request.request.method</code> è 'GET' e <code>apartments</code> coincide con <code>[{ id: 'p1', name: 'Alfa' }, { id: 'p2', name: 'Zeta' },]</code> .	Esito atteso: il valore di <code>request.request.method</code> è 'GET'; <code>apartments</code> coincide con <code>[{ id: 'p1', name: 'Alfa' }, { id: 'p2', name: 'Zeta' },]</code> .	S
TI-92	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>getAllPlants</i> supporta risposta wrappata in data[]' : verifica che <code>plants</code> ha lunghezza 2 e il valore di <code>plants[1].name</code> è 'p10'.	Esito atteso: <code>plants</code> ha lunghezza 2; il valore di <code>plants[1].name</code> è 'p10'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-93	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>getCurrentApartment</i> usa <i>activePlantId</i> salvato e mappa <i>apartment view</i> ': verifica che il valore di <i>apartment.id</i> è 'p2' e <i>apartment.rooms</i> ha lunghezza 1.	Esito atteso: il valore di <i>apartment.id</i> è 'p2'; <i>apartment.rooms</i> ha lunghezza 1; <i>apartment.rooms[0].devices</i> ha lunghezza 1; il valore di <i>apartment.rooms[0].devices[0].id</i> è 'd-main'.	S
TI-94	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>getCurrentApartment</i> fallisce se non esistono <i>plant disponibili</i> ': verifica che dopo rigetto, <i>apartmentPromise</i> genera l'errore atteso.	Esito atteso: dopo rigetto, <i>apartmentPromise</i> genera l'errore atteso.	S
TI-95	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>setActivePlantId</i> non emette evento se <i>plant</i> invariato': verifica che <i>dispatchSpy</i> non viene invocato e il valore di <i>global-This.localStorage.getItem('activePlantId')</i> è 'p1'.	Esito atteso: <i>dispatchSpy</i> non viene invocato; il valore di <i>global-This.localStorage.getItem('activePlantId')</i> è 'p1'.	S
TI-96	Contesto <i>ApartmentApiService</i> ; caso 'RF72-OBL <i>getAllPlants</i> sincronizza <i>wardIds</i> nel <i>realtime cache</i> con utente loggato': verifica che <i>wardRealtimeCacheStub.mergeWardIds</i> viene invocato con 'u-1'.	Esito atteso: <i>wardRealtimeCacheStub.mergeWardIds</i> viene invocato con 'u-1'.	S
TI-97	Contesto <i>NavService</i> ; caso 'viene creato': verifica che <i>service</i> è valorizzato.	Esito atteso: <i>service</i> è valorizzato.	S
TI-98	Contesto <i>NavService</i> ; caso 'restituisce tutti i <i>NavItems</i> per <i>AMMINISTRATORE</i> ': verifica che <i>items.length</i> è maggiore di 0 e il valore di <i>items.some(i => i.label === "Dashboard")</i> è true.	Esito atteso: <i>items.length</i> è maggiore di 0; il valore di <i>items.some(i => i.label === "Dashboard")</i> è true; il valore di <i>items.some(i => i.route.includes('dashboard'))</i> è true; il valore di <i>items.some(i => i.label === "Configurazione Allarmi")</i> è true.	S
TI-99	Contesto <i>NavService</i> ; caso 'restituisce solo i <i>NavItems</i> filtrati per <i>OPERATORE SANITARIO</i> ': verifica che <i>OSSIItems.length</i> è maggiore di 0 e il valore di <i>roleReqItems.length</i> è 0.	Esito atteso: <i>OSSIItems.length</i> è maggiore di 0; il valore di <i>roleReqItems.length</i> è 0; il valore di <i>items.some(i => i.label === "Dashboard")</i> è true; il valore di <i>items.some(i => i.route.includes('dashboard'))</i> è true.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-100	Contesto <i>DeviceApiService</i> ; consolidated summary for RF79-RF81 covering device data retrieval, filtering, deduplication and write operations: verifica che <code>getWritableEndpointRows</code> GET /plant/id, mappa datapoint per stanza, POST /api/device con payload <code>datapointId</code> , <code>value</code> , propaga errori write, <code>resolveActivePlantId</code> GET /plant/all se mancante, esclude energy-measure devices, deduplica per nome mantenendo priorità, <code>getCurrentValuePointsByDeviceIds</code> GET valori, continua merge se un device call fallisce, error handling per timeout.	Esito atteso: room mapping by id; datapoint array; POST write correct; error propagation; plant resolution; device filtering; deduplication priority; value map retrieval; resilience on partial failures.	S
TI-101	Contesto <i>NotificationService</i> ; caso 'espone vm iniziale vuoto quando storico e stream in-session sono vuoti': verifica che vm coincide con { notifications: [], unreadCount: 0, }.	Esito atteso: vm coincide con { notifications: [], unreadCount: 0, }.	S
TI-102	Contesto <i>NotificationService</i> ; caso 'combina storico e in-session, deduplica per notificationId e ordina per sentAt desc': verifica che il valore di vm.unreadCount è 2 e vm.notifications coincide con [sharedInSession, inSessionOnly, historyOlder].	Esito atteso: il valore di vm.unreadCount è 2; vm.notifications coincide con [sharedInSession, inSessionOnly, historyOlder]; vm.notifications ha lunghezza 3.	S
TI-103	Contesto <i>NotificationService</i> ; caso 'in errore sullo storico applica fallback a [] e mantiene i dati in-session': verifica che vm.notifications coincide con [inSessionOnly] e il valore di vm.unreadCount è 1.	Esito atteso: vm.notifications coincide con [inSessionOnly]; il valore di vm.unreadCount è 1.	S
TI-104	Contesto <i>NotificationService</i> ; caso 'propaga nuovi unreadCount senza alterare la lista notifiche': verifica che vm.notifications coincide con [historyOlder] e il valore di vm.unreadCount è 5.	Esito atteso: vm.notifications coincide con [historyOlder]; il valore di vm.unreadCount è 5.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-105	Contesto <i>NotificationService</i> ; caso 'aggiorna il vm quando arrivano nuove notifiche in-session dopo la prima emissione': verifica che <code>vm.notifications</code> coincide con <code>[inSessionOnly, historyOlder]</code> e il valore di <code>vm.unreadCount</code> è 0.	Esito atteso: <code>vm.notifications</code> coincide con <code>[inSessionOnly, historyOlder]</code> ; il valore di <code>vm.unreadCount</code> è 0.	S
TI-106	Contesto <i>NotificationService</i> ; caso 'deduplica tenendo la versione in-session quando <code>notificationId</code> è presente anche nello storico': verifica che <code>vm.notifications</code> coincide con <code>[sharedInSession]</code> e <code>vm.notifications</code> ha lunghezza 1.	Esito atteso: <code>vm.notifications</code> coincide con <code>[sharedInSession]</code> ; <code>vm.notifications</code> ha lunghezza 1.	S
TI-107	Contesto <i>NotificationService</i> ; caso 'gestisce timestamp non validi senza lanciare errori e li ordina in fondo': verifica che <code>vm.notifications</code> coincide con <code>[historyOlder, invalidTimestampNotification]</code> .	Esito atteso: <code>vm.notifications</code> coincide con <code>[historyOlder, invalidTimestampNotification]</code> .	S
TI-108	Contesto <i>NotificationService</i> ; caso 'riusa la stessa pipeline vm con più subscriber senza richiamare di nuovo lo storico': verifica che <code>vmA</code> coincide con <code>vmB</code> e <code>notificationApiStub.getNotificationsHistory</code> viene invocato 1 volta.	Esito atteso: <code>vmA</code> coincide con <code>vmB</code> ; <code>notificationApiStub.getNotificationsHistory</code> viene invocato 1 volta.	S
TI-109	Contesto <i>NotificationService</i> ; caso 'mantiene uno snapshot coerente per subscriber tardivi senza ricaricare lo storico': verifica che <code>vmLateSubscriber.notifications</code> coincide con <code>[inSessionOnly, historyOlder]</code> e <code>notificationApiStub.getNotificationsHistory</code> viene invocato 1 volta.	Esito atteso: <code>vmLateSubscriber.notifications</code> coincide con <code>[inSessionOnly, historyOlder]</code> ; <code>notificationApiStub.getNotificationsHistory</code> viene invocato 1 volta.	S
TI-110	Contesto <i>NotificationService</i> ; caso 'non muta gli array sorgente ricevuti da API e <i>AlarmStateService</i> ': verifica che <code>historicList</code> coincide con <code>[sharedHistoric, historyOlder]</code> e <code>inSessionList</code> coincide con <code>[sharedInSession, inSessionOnly]</code> .	Esito atteso: <code>historicList</code> coincide con <code>[sharedHistoric, historyOlder]</code> ; <code>inSessionList</code> coincide con <code>[sharedInSession, inSessionOnly]</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-111	Contesto <i>NotificationService</i> ; caso 'deduplica anche duplicati interni alla stessa sorgente mantenendo l'ultimo elemento': verifica che <code>vm.notifications</code> coincide con <code>[duplicatedSessionB, duplicatedHistoryB]</code> e <code>vm.notifications</code> ha lunghezza 2.	Esito atteso: <code>vm.notifications</code> coincide con <code>[duplicatedSessionB, duplicatedHistoryB]</code> ; <code>vm.notifications</code> ha lunghezza 2.	S
TI-112	Contesto <i>NotificationService</i> ; caso 'mantiene coerenza: unreadCount segue lo stream dedicato anche quando la deduplica riduce la lista': verifica che <code>vm.notifications</code> ha lunghezza 1 e il valore di <code>vm.unreadCount</code> è 5.	Esito atteso: <code>vm.notifications</code> ha lunghezza 1; il valore di <code>vm.unreadCount</code> è 5.	S
TI-113	Contesto <i>NotificationService</i> ; caso 'filtra le notifiche cancellate localmente dall'archivio unificato': verifica che <code>vm.notifications.map((notification) => notification.notificationId)</code> coincide con <code>['session-1', 'shared-id',]</code> .	Esito atteso: <code>vm.notifications.map((notification) => notification.notificationId)</code> coincide con <code>['session-1', 'shared-id',]</code> .	S
TI-114	Contesto <i>NotificationService</i> ; caso ' <i>removeNotification</i> delega <i>dismiss</i> + <i>remove</i> allo stato <i>realtime</i> ': verifica che <code>alarmStateStub.dismissNotification</code> viene invocato con 'n-100' e <code>alarmStateStub.removeNotification</code> viene invocato con 'n-100'.	Esito atteso: <code>alarmStateStub.dismissNotification</code> viene invocato con 'n-100'; <code>alarmStateStub.removeNotification</code> viene invocato con 'n-100'.	S
TI-115	Contesto <i>NotificationService</i> ; caso ' <i>clearAllNotifications</i> delega <i>dismiss</i> multiplo + <i>clear</i> <i>realtime</i> ': verifica che <code>alarmStateStub.dismissNotifications</code> viene invocato con <code>[historyOlder.notificationId, inSessionOnly.notificationId,]</code> e <code>alarmStateStub.clearNotifications</code> viene invocato 1 volta.	Esito atteso: <code>alarmStateStub.dismissNotifications</code> viene invocato con <code>[historyOlder.notificationId, inSessionOnly.notificationId,]</code> ; <code>alarmStateStub.clearNotifications</code> viene invocato 1 volta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-116	Contesto <i>UserApiService</i> ; consolidated summary for user CRUD and password decoding: verifica che getUsers GET /users, createUser POST /users con dto e decodifica tempPassword base64 al result, deleteUser DELETE /users/:id con id numerico, decodeTempPassword decodifica/mantiene base64 valida, empty string preserve, invalid base64 mantiene originale, atob exception fallback, whitespace trim per decode, spazi marginali gestiti.	Esito atteso: GET users list; POST create returns decoded tempPassword; DELETE removes; base64 decoding works; invalid inputs preserved; exception handling fallback; whitespace handling correct.	S
TI-117	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>assignOperator</i> esegue operazione e poi ricarica i wards': verifica che result è undefined e <i>apiStub.assignOperatorToWard</i> viene invocato con 1.	Esito atteso: result è undefined; <i>apiStub.assignOperatorToWard</i> viene invocato con 1; l'asserzione <i>apiStub.getWards</i> <i>toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> viene invocato con 1.	S
TI-118	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>removeOperator</i> in errore espone errore testo da <i>HttpErrorResponse.error</i> ': verifica che il valore di emitted è false e <i>storeStub.setError</i> viene invocato con 'Operatore non assegnato'.	Esito atteso: il valore di emitted è false; <i>storeStub.setError</i> viene invocato con 'Operatore non assegnato'; <i>storeStub.setError</i> viene invocato 1 volta; <i>apiStub.getWards</i> non viene invocato.	S
TI-119	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>getAvailablePlantsForWard</i> filtra i plant già assegnati in qualunque ward': verifica che result coincide con [{ id: '203', name: 'App. 203' }] e l'asserzione <i>apiStub.getAvailablePlants</i> <i>toHaveBeenCalledOnce</i> risulta soddisfatta.	Esito atteso: result coincide con [{ id: '203', name: 'App. 203' }]; l'asserzione <i>apiStub.getAvailablePlants</i> <i>toHaveBeenCalledOnce</i> risulta soddisfatta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-120	Contesto <i>AssignmentOperationsService</i> ; caso 'dopo rimozione un appartamento torna disponibile': verifica che result coincide con [{ id: '202', name: 'App. 202' }] e result coincide con [{ id: '201', name: 'App. 201' }, { id: '202', name: 'App. 202' },].	Esito atteso: result coincide con [{ id: '202', name: 'App. 202' }]; result coincide con [{ id: '201', name: 'App. 201' }, { id: '202', name: 'App. 202' },].	S
TI-121	Contesto <i>AssignmentOperationsService</i> ; caso 'dopo eliminazione reparto, gli appartamenti del reparto eliminato tornano disponibili': verifica che result coincide con [{ id: '203', name: 'App. 203' }] e result coincide con [{ id: '201', name: 'App. 201' }, { id: '203', name: 'App. 203' },].	Esito atteso: result coincide con [{ id: '203', name: 'App. 203' }]; result coincide con [{ id: '201', name: 'App. 201' }, { id: '203', name: 'App. 203' },].	S
TI-122	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>assignPlant</i> in errore usa <i>HttpErrorResponse.message</i> quando non c'è error string': verifica che messageArg.length è maggiore di 0 e apiStub.getWards non viene invocato.	Esito atteso: messageArg.length è maggiore di 0; apiStub.getWards non viene invocato; storeStub.setWards non viene invocato.	S
TI-123	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>removePlant</i> in errore non-http usa fallback message': verifica che storeStub.setError viene invocato con 'Operazione di assegnazione non riuscita.' e storeStub.setError viene invocato 1 volta.	Esito atteso: storeStub.setError viene invocato con 'Operazione di assegnazione non riuscita.'; storeStub.setError viene invocato 1 volta; apiStub.getWards non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-124	Contesto <i>WardApiService</i> ; consolidated summary for RF73-RF76 covering ward and relationship management endpoints: verifica che getWards GET /api/wards, getOperatorsByWardId GET /api/wards-users-relationships/:wardId, getAvailableOperators GET /users, getPlantsByWardId GET /api/wards-plants-relationships/:wardId, getAvailablePlants GET /api/plant/all, createWard POST /api/wards con dto, updateWard PUT /api/wards/:id, deleteWard DELETE /api/wards/:id, assignOperatorToWard POST /api/wards-users-relationships con wardId userId, removeOperatorFromWard DELETE /api/wards-users-relationships/:wardId/:userId, assignPlantToWard POST /api/wards-plants-relationships, removePlantFromWard DELETE /api/wards-plants-relationships/:plantId, mapping compatibility.	Esito atteso: GET /wards lista; GET operators by ward; GET available operators; GET plants by ward; GET all available plants; POST create returns ward; PUT update preserves data; DELETE removes ward; POST relationship created; DELETE relationship removed; assignment endpoints work; backward compatibility maintained.	S
TI-125	Contesto <i>WardOperationsService</i> ; caso ' <i>loadWards</i> aggiorna store e termina senza valore': verifica che result è undefined e l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta.	Esito atteso: result è undefined; l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> viene invocato con 1; <i>apiStub.getOperatorsByWardId</i> viene invocato con 1.	S
TI-126	Contesto <i>WardOperationsService</i> ; caso ' <i>loadWards</i> con lista vuota non richiama le <i>relationship</i> API': verifica che l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta e <i>apiStub.getPlantsByWardId</i> non viene invocato.	Esito atteso: l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> non viene invocato; <i>apiStub.getOperatorsByWardId</i> non viene invocato; <i>storeStub.setWards</i> viene invocato con [].	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-127	Contesto <i>WardOperationsService</i> ; caso ' <i>createWard</i> con 409 imposta messaggio funzionale': verifica che il valore di <i>emitted</i> è false e <i>storeStub.addWard</i> non viene invocato.	Esito atteso: il valore di <i>emitted</i> è false; <i>storeStub.addWard</i> non viene invocato; <i>storeStub.setError</i> viene invocato con 'Esiste già un reparto con questo nome.'; <i>storeStub.setError</i> viene invocato 1 volta.	S
TI-128	Contesto <i>WardOperationsService</i> ; caso ' <i>createWard</i> normalizza payload minimale mantenendo il modello <i>Ward</i> coerente': verifica che <i>storeStub.addWard</i> viene invocato con { id: 11, name: 'Oncologia', apartments: [], operators: [], } e <i>storeStub.setLoading</i> viene invocato con false.	Esito atteso: <i>storeStub.addWard</i> viene invocato con { id: 11, name: 'Oncologia', apartments: [], operators: [], }; <i>storeStub.setLoading</i> viene invocato con false.	S
TI-129	Contesto <i>WardOperationsService</i> ; caso ' <i>createWard</i> preserva apartments/operators dallo snapshot se il backend non li restituisce': verifica che <i>storeStub.addWard</i> viene invocato con { id: 1, name: 'Cardiologia nuova', apartments: ward.apartments, operators: ward.operators, }.	Esito atteso: <i>storeStub.addWard</i> viene invocato con { id: 1, name: 'Cardiologia nuova', apartments: ward.apartments, operators: ward.operators, }.	S
TI-130	Contesto <i>WardOperationsService</i> ; caso ' <i>updateWard</i> sostituisce il ward nello store': verifica che <i>apiStub.updateWard</i> viene invocato con 1 e <i>storeStub.replaceWard</i> viene invocato con updatedWard.	Esito atteso: <i>apiStub.updateWard</i> viene invocato con 1; <i>storeStub.replaceWard</i> viene invocato con updatedWard; <i>storeStub.replaceWard</i> viene invocato 1 volta; <i>storeStub.setLoading</i> viene invocato con false.	S
TI-131	Contesto <i>WardOperationsService</i> ; caso ' <i>updateWard</i> preserva apartments/operators quando backend restituisce solo id e name': verifica che <i>storeStub.replaceWard</i> viene invocato con { id: 1, name: 'Cardiologia X', apartments: ward.apartments, operators: ward.operators, } e <i>storeStub.setLoading</i> viene invocato con false.	Esito atteso: <i>storeStub.replaceWard</i> viene invocato con { id: 1, name: 'Cardiologia X', apartments: ward.apartments, operators: ward.operators, }; <i>storeStub.setLoading</i> viene invocato con false.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-132	Contesto <i>WardOperationsService</i> ; caso ' updateWard gestisce errore HTTP con messaggio testuale del server': verifica che <i>storeStub.setError</i> viene invocato con 'update failed' e <i>storeStub.replaceWard</i> non viene invocato.	Esito atteso: <i>storeStub.setError</i> viene invocato con 'update failed'; <i>storeStub.replaceWard</i> non viene invocato.	S
TI-133	Contesto <i>WardOperationsService</i> ; caso ' deleteWard rimuove il ward dallo store': verifica che <i>apiStub.deleteWard</i> viene invocato con 1 e <i>storeStub.removeWard</i> viene invocato con 1.	Esito atteso: <i>apiStub.deleteWard</i> viene invocato con 1; <i>storeStub.removeWard</i> viene invocato con 1; <i>storeStub.removeWard</i> viene invocato 1 volta; <i>storeStub.setLoading</i> viene invocato con false.	S
TI-134	Contesto <i>WardOperationsService</i> ; caso ' deleteWard usa il fallback generico quando l'errore non è strutturato': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.' e <i>storeStub.removeWard</i> non viene invocato.	Esito atteso: <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'; <i>storeStub.removeWard</i> non viene invocato.	S
TI-135	Contesto <i>WardOperationsService</i> ; caso ' createWard usa il fallback generico quando l'errore non è strutturato': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'.	Esito atteso: <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'.	S
TI-136	Contesto <i>WardOperationsService</i> ; caso ' in caso di errore generico usa il fallback message': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.' e <i>storeStub.setError</i> viene invocato 1 volta.	Esito atteso: <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'; <i>storeStub.setError</i> viene invocato 1 volta; <i>storeStub.setWards</i> non viene invocato.	S
TI-137	Contesto <i>WardOperationsService</i> ; caso ' loadWards normalizza apartments dal dto ': verifica che <i>storeStub.setWards</i> viene invocato con [{ ...hydratedWard, apartments: [{ id: '101', name: 'App. 101' }], },].	Esito atteso: <i>storeStub.setWards</i> viene invocato con [{ ...hydratedWard, apartments: [{ id: '101', name: 'App. 101' }], },].	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-138	Contesto <i>WardStore</i> ; caso 'inizializza stato vuoto' : verifica che <code>await firstValueFrom(store.wards\$)</code> coincide con <code>[]</code> e il valore di <code>await firstValueFrom(store.isLoading\$)</code> è <code>false</code> .	Esito atteso: <code>await firstValueFrom(store.wards\$)</code> coincide con <code>[]</code> ; il valore di <code>await firstValueFrom(store.isLoading\$)</code> è <code>false</code> ; <code>await firstValueFrom(store.error\$)</code> è <code>null</code> .	S
TI-139	Contesto <i>WardStore</i> ; caso 'setWards sovrascrive i wards' : verifica che <code>wards</code> coincide con <code>[wardA, wardB]</code> e <code>wards</code> ha lunghezza 2.	Esito atteso: <code>wards</code> coincide con <code>[wardA, wardB]</code> ; <code>wards</code> ha lunghezza 2.	S
TI-140	Contesto <i>WardStore</i> ; caso 'addWard aggiunge il reparto e resetta error/loading' : verifica che <code>wards</code> coincide con <code>[wardA]</code> e <code>wards</code> ha lunghezza 1.	Esito atteso: <code>wards</code> coincide con <code>[wardA]</code> ; <code>wards</code> ha lunghezza 1; il valore di <code>await firstValueFrom(store.isLoading\$)</code> è <code>false</code> ; <code>await firstValueFrom(store.error\$)</code> è <code>null</code> .	S
TI-141	Contesto <i>WardStore</i> ; caso 'replaceWard aggiorna il reparto target' : verifica che <code>wards</code> coincide con <code>[wardA, renamed]</code> e <code>wards</code> ha lunghezza 2.	Esito atteso: <code>wards</code> coincide con <code>[wardA, renamed]</code> ; <code>wards</code> ha lunghezza 2.	S
TI-142	Contesto <i>WardStore</i> ; caso 'removeWard elimina il reparto target' : verifica che <code>wards</code> coincide con <code>[wardB]</code> e <code>wards</code> ha lunghezza 1.	Esito atteso: <code>wards</code> coincide con <code>[wardB]</code> ; <code>wards</code> ha lunghezza 1.	S
TI-143	Contesto <i>WardStore</i> ; caso 'patchPlant aggiorna solo l'appartamento richiesto' : verifica che <code>wards</code> ha lunghezza 2 e <code>wards[0].apartments[0]</code> coincide con <code>wardA.apartments[0]</code> .	Esito atteso: <code>wards</code> ha lunghezza 2; <code>wards[0].apartments[0]</code> coincide con <code>wardA.apartments[0]</code> ; <code>wards[1].apartments[0]</code> coincide con <code>{ id: '102', name: 'App. 102A' }</code> .	S
TI-144	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'mostra lo stato iniziale non attivo' : verifica che il valore di <code>button.getAttribute('aria-pressed')</code> è <code>false</code> e <code>component.trackClass()</code> contiene <code>'border-slate-500'</code> .	Esito atteso: il valore di <code>button.getAttribute('aria-pressed')</code> è <code>false</code> ; <code>component.trackClass()</code> contiene <code>'border-slate-500'</code> ; <code>component.thumbClass()</code> contiene <code>'translate-x-0.5'</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-145	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'mostra lo stato attivo e la classe corretta': verifica che il valore di <code>button.getAttribute('aria-pressed')</code> è 'true' e <code>component.trackClass()</code> contiene 'border-yellow-700'.	Esito atteso: il valore di <code>button.getAttribute('aria-pressed')</code> è 'true'; <code>component.trackClass()</code> contiene 'border-yellow-700'; <code>component.thumbClass()</code> contiene 'translate-x-4'.	S
TI-146	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'emette il nuovo valore quando viene toggolato': verifica che <code>emitSpy</code> viene invocato con true e <code>emitSpy</code> viene invocato 1 volta.	Esito atteso: <code>emitSpy</code> viene invocato con true; <code>emitSpy</code> viene invocato 1 volta.	S
TI-147	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'non emette quando è disabilitato': verifica che <code>emitSpy</code> non viene invocato e il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled</code> è true.	Esito atteso: <code>emitSpy</code> non viene invocato; il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled</code> è true.	S
TI-148	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'usa aria-label custom quando fornita': verifica che il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).getAttribute('aria-label')</code> è 'Attiva allarme'.	Esito atteso: il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).getAttribute('aria-label')</code> è 'Attiva allarme'.	S
TI-149	Contesto <i>AlarmConfigFormComponent</i> ; caso 'crea il componente': verifica che <code>component</code> è valorizzato.	Esito atteso: <code>component</code> è valorizzato.	S
TI-150	Contesto <i>AlarmConfigFormComponent</i> ; caso 'buildForm applica i validatori richiesti': verifica che il valore di <code>component.form.controls.plantId.invalid</code> è true e il valore di <code>component.form.controls.deviceId.invalid</code> è true.	Esito atteso: il valore di <code>component.form.controls.plantId.invalid</code> è true; il valore di <code>component.form.controls.deviceId.invalid</code> è true; il valore di <code>component.form.controls.datapointId.invalid</code> è true; il valore di <code>component.form.controls.priority.invalid</code> è true.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-151	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in create mode mantiene il dispositivo bloccato finché non viene selezionato un plant': verifica che il valore di <code>component.form.controls.deviceId.disabled</code> è true e <code>apartmentApiStub.loadApartmentViewForPlantId</code> viene invocato con 'plant-1'.	Esito atteso: il valore di <code>component.form.controls.deviceId.disabled</code> è true; <code>apartmentApiStub.loadApartmentViewForPlantId</code> viene invocato con 'plant-1'; il valore di <code>component.form.controls.deviceId.enabled</code> è true; il valore di <code>component.deviceOptions()[0]?.id</code> è 'sensor-1'.	S
TI-152	Contesto <i>AlarmConfigFormComponent</i> ; caso 'mostra solo datapoint leggibili dopo la selezione dispositivo': verifica che <code>component.datapointOptions().map((datapoint) => datapoint.id)</code> coincide con ['dp-readable-1', 'dp-readable-2',] e il valore di <code>component.datapointOptions().every((datapoint) => datapoint.readable)</code> è true.	Esito atteso: <code>component.datapointOptions().map((datapoint) => datapoint.id)</code> coincide con ['dp-readable-1', 'dp-readable-2',]; il valore di <code>component.datapointOptions().every((datapoint) => datapoint.readable)</code> è true.	S
TI-153	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in create mode non mostra il campo posizione read-only': verifica che <code>positionInput</code> è null.	Esito atteso: <code>positionInput</code> è null.	S
TI-154	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode mostra il campo posizione read-only valorizzato': verifica che <code>positionInput</code> non è null e il valore di <code>positionInput?.value</code> è 'Appartamento 2 - Ingresso - Sensore porta'.	Esito atteso: <code>positionInput</code> non è null; il valore di <code>positionInput?.value</code> è 'Appartamento 2 - Ingresso - Sensore porta'.	S
TI-155	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode non mostra il campo datapoint nel form': verifica che <code>datapointInput</code> è null.	Esito atteso: <code>datapointInput</code> è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-156	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode mostra thresholdOperator come campo readonly': verifica che thresholdOperatorReadOnly non è null e il valore di thresholdOperatorReadOnly?.value è ThresholdOperator.EQUAL_TO.	Esito atteso: thresholdOperatorReadOnly non è null; il valore di thresholdOperatorReadOnly?.value è ThresholdOperator.EQUAL_TO; thresholdOperatorSelect è null.	S
TI-157	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode limita gli operatori al datapoint risolto e resetta operatori non supportati': verifica che component.thresholdOperatorOptions() coincide con [ThresholdOperator.EQUAL_TO] e component.form.controls.thresholdOperator.value è null.	Esito atteso: component.thresholdOperatorOptions() coincide con [ThresholdOperator.EQUAL_TO]; component.form.controls.thresholdOperator.value è null.	S
TI-158	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode valida thresholdValue in base al datapoint risolto': verifica che il valore di component.form.controls.thresholdValue.errors?.['invalidEnumThreshold'] è true.	Esito atteso: il valore di component.form.controls.thresholdValue.errors?.['invalidEnumThreshold'] è true.	S
TI-159	Contesto <i>AlarmConfigFormComponent</i> ; caso 'onSubmit emette submittedForm in create mode con form valido': verifica che emitSpy viene invocato con validFormValue e emitSpy viene invocato 1 volta.	Esito atteso: emitSpy viene invocato con validFormValue; emitSpy viene invocato 1 volta.	S
TI-160	Contesto <i>AlarmConfigFormComponent</i> ; caso 'onSubmit non invia se il form è invalido': verifica che emitSpy non viene invocato.	Esito atteso: emitSpy non viene invocato.	S
TI-161	Contesto <i>AlarmConfigFormComponent</i> ; caso 'onCancel emette evento cancelled': verifica che emitSpy viene invocato 1 volta.	Esito atteso: emitSpy viene invocato 1 volta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-162	Contesto <i>AlarmConfigFormComponent</i> ; caso 'espone opzioni enum per priorità e operatore': verifica che <code>component.priorityOptions</code> coincide con [<code>AlarmPriority.WHITE</code> , <code>AlarmPriority.GREEN</code> , <code>AlarmPriority.ORANGE</code> , <code>AlarmPriority.RED</code> ,] e <code>component.thresholdOperatorOptions()</code> coincide con [<code>ThresholdOperator.GREATER_THAN</code> , <code>ThresholdOperator.GREATER_THAN_OR_EQUAL</code> , <code>ThresholdOperator.L...</code>	Esito atteso: <code>component.priorityOptions</code> coincide con [<code>AlarmPriority.WHITE</code> , <code>AlarmPriority.GREEN</code> , <code>AlarmPriority.ORANGE</code> , <code>AlarmPriority.RED</code> ,]; <code>component.thresholdOperatorOptions()</code> coincide con [<code>ThresholdOperator.GREATER_THAN</code> , <code>ThresholdOperator.GREATER_THAN_OR_EQUAL</code> , <code>ThresholdOperator.L...</code>	S
TI-163	Contesto <i>AlarmConfigFormComponent</i> ; caso 'limita operatori e valore soglia quando il datapoint ha enum': verifica che <code>component.thresholdOperatorOptions()</code> coincide con [<code>ThresholdOperator.EQUAL_TO</code>] e il valore di <code>component.form.controls.thresholdOperator</code> è <code>true</code> .	Esito atteso: <code>component.thresholdOperatorOptions()</code> coincide con [<code>ThresholdOperator.EQUAL_TO</code>]; il valore di <code>component.form.controls.thresholdOperator.invalid</code> è <code>true</code> ; il valore di <code>component.form.controls.thresholdValue.invalid</code> è <code>true</code> ; il valore di <code>component.form.controls.thresholdValue.errors?.['invalidEnumThresholdOperator']</code> è <code>true</code> .	S
TI-164	Contesto <i>AlarmConfigFormComponent</i> ; caso 'accetta valore soglia booleano quando operatore è uguale': verifica che il valore di <code>component.form.controls.thresholdOperator</code> è <code>true</code> e il valore di <code>component.form.controls.thresholdValue</code> è <code>true</code> .	Esito atteso: il valore di <code>component.form.controls.thresholdOperator.valid</code> è <code>true</code> ; il valore di <code>component.form.controls.thresholdValue.valid</code> è <code>true</code> .	S
TI-165	Contesto <i>AlarmConfigFormComponent</i> ; caso 'carica impianti da /plant/all deduplicando e ordinando per nome': verifica che <code>component.plants().map((plant) => plant.id)</code> coincide con ['plant-1', 'plant-2', 'plant-3'] e <code>component.plantsLoadError()</code> è <code>null</code> .	Esito atteso: <code>component.plants().map((plant) => plant.id)</code> coincide con ['plant-1', 'plant-2', 'plant-3']; <code>component.plantsLoadError()</code> è <code>null</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-166	Contesto <i>AlarmConfigFormComponent</i> ; caso 'se /plant/all fallisce usa fallback available + plants assegnati ai reparti': verifica che component.plants().map((plant) => plant.id) coincide con ['plant-7', 'plant-8', 'plant-9'] e component.plantsLoadError() è null.	Esito atteso: component.plants().map((plant) => plant.id) coincide con ['plant-7', 'plant-8', 'plant-9']; component.plantsLoadError() è null; wardApi- Stub.getPlantsByWardId viene invocato 1 volta.	S
TI-167	Contesto <i>AlarmConfigFormComponent</i> ; caso 'se il plant viene deselezionato resetta opzioni dispositivo e blocca il campo dispositivo': verifica che il valore di component.deviceOptions()[0]?.id è 'sensor-1' e il valore di compo- nent.form.controls.deviceId.enabled è true.	Esito atteso: il valore di component.deviceOptions()[0]?.id è 'sensor-1'; il valore di compo- nent.form.controls.deviceId.enabled è true; component.deviceOptions() coincide con []; il valore di compo- nent.form.controls.deviceId.value è ''.	S
TI-168	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode non consente di modificare il dispositivo associato': verifica che il valore di compo- nent.form.controls.deviceId.disabled è true e component.deviceOptions() coincide con [{ id: 'sensor-9', label: 'sensor-9', datapoints: [] }].	Esito atteso: il valore di compo- nent.form.controls.deviceId.disabled è true; component.deviceOptions() coincide con [{ id: 'sensor-9', label: 'sensor-9', datapoints: [] }].	S
TI-169	Contesto <i>AlarmConfigFormComponent</i> ; caso 'imposta errore quando il caricamento dispositivi fallisce': verifica che il valore di component.devicesLoadError() è 'Errore durante il caricamento dei dispositivi.' e il valore di compo- nent.form.controls.deviceId.disabled è true.	Esito atteso: il valore di component.devicesLoadError() è 'Errore durante il caricamento dei dispositivi.'; il valore di compo- nent.form.controls.deviceId.disabled è true.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-170	Contesto <i>AlarmConfigForm</i> feature integration; caso 'RF88-OBL create mode carica impianti e abilita selezione dispositivo': verifica che <code>component.plants().map((plant) => plant.id)</code> coincide con ['plant-1', 'plant-2'] e il valore di <code>component.form.controls.deviceId.disabled</code> è true.	Esito atteso: <code>component.plants().map((plant) => plant.id)</code> coincide con ['plant-1', 'plant-2']; il valore di <code>component.form.controls.deviceId.disabled</code> è true; <code>apartmentApiStub.loadApartmentViewForPlantId</code> viene invocato con 'plant-1'; il valore di <code>component.form.controls.deviceId.enabled</code> è true.	S
TI-171	Contesto <i>AlarmConfigForm</i> feature integration; caso 'RF89-OBL selezione datapoint enum limita operatore e valida threshold': verifica che <code>component.thresholdOperatorOptions()</code> coincide con [ThresholdOperator.EQUAL_TO] e il valore di <code>component.form.controls.thresholdOperator.errors?['unsupportedOperator']</code> è true.	Esito atteso: <code>component.thresholdOperatorOptions()</code> coincide con [ThresholdOperator.EQUAL_TO]; il valore di <code>component.form.controls.thresholdOperator.errors?['unsupportedOperator']</code> è true; il valore di <code>component.form.controls.thresholdValue.errors?['invalidEnumThreshold']</code> è true.	S
TI-172	Contesto <i>AlarmConfigForm</i> feature integration; caso 'RF91-OBL edit mode blocca campi vincolati e mantiene nome originale su submit': verifica che il valore di <code>component.form.controls.name.disabled</code> è true e il valore di <code>component.form.controls.deviceId.disabled</code> è true.	Esito atteso: il valore di <code>component.form.controls.name.disabled</code> è true; il valore di <code>component.form.controls.deviceId.disabled</code> è true; il valore di <code>component.form.controls.datapointId.disabled</code> è true; <code>emitSpy</code> viene invocato 1 volta.	S
TI-173	Contesto <i>AlarmConfigForm</i> feature integration; caso 'RF92-OBL annulla form emettendo evento cancelled': verifica che <code>cancelSpy</code> viene invocato 1 volta.	Esito atteso: <code>cancelSpy</code> viene invocato 1 volta.	S
TI-174	Contesto <i>AlarmConfigPageComponent</i> ; caso 'crea il componente': verifica che <code>component</code> è valorizzato.	Esito atteso: <code>component</code> è valorizzato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-175	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>ngOnInit</i> carica gli allarmi e aggiorna le righe tabellari': verifica che <i>stateServiceStub.loadAlarmRules</i> viene invocato 1 volta e il valore di <i>component.rows().length</i> è 1.	Esito atteso: <i>stateServiceStub.loadAlarmRules</i> viene invocato 1 volta; il valore di <i>component.rows().length</i> è 1; il valore di <i>component.error()</i> è 'Errore test'.	S
TI-176	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onCreateNew</i> apre la modale in create mode': verifica che il valore di <i>component.isModalOpen()</i> è true e <i>component.editingRule()</i> è null.	Esito atteso: il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null; il valore di <i>component.modalTitle()</i> è 'Creazione allarme'.	S
TI-177	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onEdit</i> apre la modale in edit mode con regola selezionata': verifica che il valore di <i>component.isModalOpen()</i> è true e il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	Esito atteso: il valore di <i>component.isModalOpen()</i> è true; il valore di <i>component.editingRule()?.id</i> è 'alarm-1'; il valore di <i>component.modalTitle()</i> è 'Modifica allarme'.	S
TI-178	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onEdit</i> ignora id non trovato': verifica che il valore di <i>component.isModalOpen()</i> è false e <i>component.editingRule()</i> è null.	Esito atteso: il valore di <i>component.isModalOpen()</i> è false; <i>component.editingRule()</i> è null.	S
TI-179	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in create mode invoca create e chiude modale': verifica che <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> e il valore di <i>component.isModalOpen()</i> è false.	Esito atteso: <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> ; il valore di <i>component.isModalOpen()</i> è false.	S
TI-180	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in create mode mantiene aperta la modale se non arriva alcun esito': verifica che <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> e il valore di <i>component.isModalOpen()</i> è true.	Esito atteso: <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> ; il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-181	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in edit mode invoca update e chiude modale': verifica che <i>stateServiceStub.updateAlarmRule</i> viene invocato con 'alarm-1' e il valore di <i>component.isModalOpen()</i> è false.	Esito atteso: <i>stateServiceStub.updateAlarmRule</i> viene invocato con 'alarm-1'; il valore di <i>component.isModalOpen()</i> è false.	S
TI-182	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in edit mode mantiene aperta la modale se non arriva alcun esito': verifica che il valore di <i>component.isModalOpen()</i> è true e il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	Esito atteso: il valore di <i>component.isModalOpen()</i> è true; il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	S
TI-183	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onToggleEnabled</i> invoca <i>toggleEnabled</i> con lo stato richiesto': verifica che <i>stateServiceStub.toggleEnabled</i> viene invocato con 'alarm-1'.	Esito atteso: <i>stateServiceStub.toggleEnabled</i> viene invocato con 'alarm-1'.	S
TI-184	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onToggleEnabled</i> blocca i click ripetuti finché la richiesta è in corso': verifica che il valore di <i>component.pendingToggleRuleId()</i> è 'alarm-1' e <i>stateServiceStub.toggleEnabled</i> viene invocato 1 volta.	Esito atteso: il valore di <i>component.pendingToggleRuleId()</i> è 'alarm-1'; <i>stateServiceStub.toggleEnabled</i> viene invocato 1 volta; <i>component.pendingToggleRuleId()</i> è null; <i>stateServiceStub.toggleEnabled</i> viene invocato 2 volte.	S
TI-185	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDelete</i> apre la modale di conferma senza chiamare subito la delete': verifica che <i>component.pendingDelete()</i> coincide con { id: 'alarm-1', name: 'Temperatura alta' } e <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	Esito atteso: <i>component.pendingDelete()</i> coincide con { id: 'alarm-1', name: 'Temperatura alta' }; <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-186	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDeleteConfirmed</i> invoca <i>deleteAlarmRule</i> con id e chiude la modale di conferma': verifica che <i>stateServiceStub.deleteAlarmRule</i> viene invocato con 'alarm-1' e <i>component.pendingDelete()</i> è null.	Esito atteso: <i>stateServiceStub.deleteAlarmRule</i> viene invocato con 'alarm-1'; <i>component.pendingDelete()</i> è null.	S
TI-187	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDeleteCancelled</i> chiude la modale di conferma senza eliminare': verifica che <i>component.pendingDelete()</i> è null e <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	Esito atteso: <i>component.pendingDelete()</i> è null; <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	S
TI-188	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>renderizza la modale di conferma cancellazione quando richiesta</i> ': verifica che <i>confirmDialog</i> non è null e content contiene 'Confermi l\{\}'eliminazione della soglia "Temperatura alta"?.	Esito atteso: <i>confirmDialog</i> non è null; content contiene 'Confermi l\{\}'eliminazione della soglia "Temperatura alta"?.	S
TI-189	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onModalClosed</i> chiude modale e resetta <i>editingRule</i> ': verifica che il valore di <i>component.isModalOpen()</i> è false e <i>component.editingRule()</i> è null.	Esito atteso: il valore di <i>component.isModalOpen()</i> è false; <i>component.editingRule()</i> è null.	S
TI-190	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onCreateNew</i> resetta l'eventuale regola in edit e apre la modale in create': verifica che il valore di <i>component.isModalOpen()</i> è true e <i>component.editingRule()</i> è null.	Esito atteso: il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null.	S
TI-191	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>renderizza la tabella con le righe e i pulsanti azione</i> ': verifica che il valore di <i>tableRows.length</i> è 1 e content contiene 'Temperatura alta'.	Esito atteso: il valore di <i>tableRows.length</i> è 1; content contiene 'Temperatura alta'; content contiene 'Posizione'; content contiene 'Appartamento 1 - Soggiorno - Sensore porta'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-192	Contesto <i>AlarmConfigPageComponent</i> ; caso 'renderizza messaggio vuoto quando non ci sono regole': verifica che content contiene 'Nessuna regola configurata.'	Esito atteso: content contiene 'Nessuna regola configurata.'	S
TI-193	Contesto <i>AlarmConfigPageComponent</i> ; caso 'renderizza il form in modale quando la modale è aperta': verifica che modalForm non è null.	Esito atteso: modalForm non è null.	S
TI-194	Contesto AlarmConfiguration feature integration; caso 'RF85-OBL inizializza pagina e carica regole allarme': verifica che component è valorizzato e stateServiceStub.loadAlarmRules viene invocato 1 volta.	Esito atteso: component è valorizzato; stateServiceStub.loadAlarmRules viene invocato 1 volta.	S
TI-195	Contesto AlarmConfiguration feature integration; caso 'RF86-OBL create mode invia form al servizio e chiude modale': verifica che stateServiceStub.createAlarmRule viene invocato con formValue e il valore di component.isModalOpen() è false.	Esito atteso: stateServiceStub.createAlarmRule viene invocato con formValue; il valore di component.isModalOpen() è false.	S
TI-196	Contesto AlarmConfiguration feature integration; caso 'RF87-OBL edit mode mantiene modale aperta se update non emette esito': verifica che il valore di component.isModalOpen() è true e il valore di component.editingRule()?.id è 'alarm-1'.	Esito atteso: il valore di component.isModalOpen() è true; il valore di component.editingRule()?.id è 'alarm-1'.	S
TI-197	Contesto <i>AlarmPageManagementComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-198	Contesto <i>AlarmPageManagementComponent</i> ; caso <i>'onResolve delega a facade.resolveAlarm'</i> : verifica che <i>alarmManagementStub.resolveAlarm</i> viene invocato con <i>'active-2'</i> e <i>alarmManagementStub.resolveAlarm</i> viene invocato 1 volta.	Esito atteso: <i>alarmManagementStub.resolveAlarm</i> viene invocato con <i>'active-2'</i> ; <i>alarmManagementStub.resolveAlarm</i> viene invocato 1 volta.	S
TI-199	Contesto <i>AlarmPageManagementComponent</i> ; caso <i>'renderizza stato lista vuota quando non ci sono allarmi'</i> : verifica che <i>nativeElement.querySelector('.alarm-management__empty')?.textContent</i> contiene <i>'Nessun allarme attivo al momento.'</i> e il valore di <i>nativeElement.querySelectorAll('tbody tr').length</i> è 0.	Esito atteso: <i>nativeElement.querySelector('.alarm-management__empty')?.textContent</i> contiene <i>'Nessun allarme attivo al momento.'</i> ; il valore di <i>nativeElement.querySelectorAll('tbody tr').length</i> è 0.	S
TI-200	Contesto <i>AlarmPageManagementComponent</i> ; caso <i>'renderizza errore e stato resolving quando presenti nel vm'</i> : verifica che <i>nativeElement.querySelector('.alarm-management__error')?.textContent</i> contiene <i>'Errore durante la risoluzione'</i> e <i>nativeElement.querySelector('.alarm-management__status')?.textContent</i> contiene <i>'Risoluzione allarme in corso...'</i> .	Esito atteso: <i>nativeElement.querySelector('.alarm-management__error')?.textContent</i> contiene <i>'Errore durante la risoluzione'</i> ; <i>nativeElement.querySelector('.alarm-management__status')?.textContent</i> contiene <i>'Risoluzione allarme in corso...'</i> .	S
TI-201	Contesto <i>AlarmPageManagementComponent</i> ; caso <i>'renderizza tabella con una riga per ogni allarme'</i> : verifica che il valore di <i>rows.length</i> è 2 e <i>nativeElement.textContent</i> contiene <i>'Priorità'</i> .	Esito atteso: il valore di <i>rows.length</i> è 2; <i>nativeElement.textContent</i> contiene <i>'Priorità'</i> ; <i>nativeElement.textContent</i> contiene <i>'Corridoio Nord'</i> ; <i>nativeElement.textContent</i> contiene <i>'mock-elapsed:2026-03-24T10:00:00.000Z'</i> .	S
TI-202	Contesto <i>AlarmPageManagementComponent</i> ; caso <i>'nasconde la colonna gestore per operatore sanitario'</i> : verifica che <i>nativeElement.textContent</i> non contiene <i>'Gestore'</i> e <i>nativeElement.textContent</i> non contiene <i>'oss_1'</i> .	Esito atteso: <i>nativeElement.textContent</i> non contiene <i>'Gestore'</i> ; <i>nativeElement.textContent</i> non contiene <i>'oss_1'</i> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-203	Contesto <i>AlarmPageManagementComponent</i> ; caso 'disabilita solo il bottone associato al resolvingId corrente': verifica che il valore di <code>firstButton?.disabled</code> è true e il valore di <code>secondButton?.disabled</code> è false.	Esito atteso: il valore di <code>firstButton?.disabled</code> è true; il valore di <code>secondButton?.disabled</code> è false.	S
TI-204	Contesto <i>AlarmPageManagementComponent</i> ; caso 'renderizza paginazione coerente con vm e delega i click ai metodi del service': verifica che <code>nativeElement.querySelector('.care-pagination__status')?.textContent</code> contiene 'Pagina 3' e il valore di <code>previousButton?.disabled</code> è false.	Esito atteso: <code>nativeElement.querySelector('.care-pagination__status')?.textContent</code> contiene 'Pagina 3'; il valore di <code>previousButton?.disabled</code> è false; il valore di <code>nextButton?.disabled</code> è true; <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	S
TI-205	Contesto <i>AlarmPageManagementComponent</i> ; caso 'click su GESTISCI propaga resolve verso facade': verifica che <code>alarmManagementStub.resolveAlarm</code> viene invocato con 'active-1' e <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	Esito atteso: <code>alarmManagementStub.resolveAlarm</code> viene invocato con 'active-1'; <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	S
TI-206	Contesto <i>AlarmPageManagementComponent</i> ; caso 'mantiene la riga visibile se l'allarme è già gestito e disabilita l'azione': verifica che il valore di <code>rows.length</code> è 1 e <code>nativeElement.textContent</code> contiene 'Non da gestire'.	Esito atteso: il valore di <code>rows.length</code> è 1; <code>nativeElement.textContent</code> contiene 'Non da gestire'; <code>managedButton</code> non è null; il valore di <code>managedButton?.disabled</code> è true.	S
TI-207	Contesto <i>AlarmPageManagementComponent</i> ; caso 'onNextPage e onPreviousPage delegano al service': verifica che <code>alarmManagementStub.nextPage</code> viene invocato 1 volta e <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	Esito atteso: <code>alarmManagementStub.nextPage</code> viene invocato 1 volta; <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-208	Contesto <i>AlarmPageManagementComponent</i> ; caso 'quando arriva <i>refreshRequested</i> reinizializza la pagina': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta e <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	Esito atteso: <i>alarmManagementStub.initialize</i> viene invocato 1 volta; <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	S
TI-209	Contesto <i>AlarmPageManagementComponent</i> ; caso 'dopo <i>destroy</i> non risponde più agli eventi di <i>refresh</i> ': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta.	Esito atteso: <i>alarmManagementStub.initialize</i> viene invocato 1 volta.	S
TI-210	Contesto <i>AlarmManagement</i> feature integration; caso 'RF49-OBL gestisce il flusso completo: <i>render</i> lista, <i>resolve</i> event e aggiornamento <i>vm</i> ': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta e il valore di <i>resolveButtons.length</i> è 2.	Esito atteso: <i>alarmManagementStub.initialize</i> viene invocato 1 volta; il valore di <i>resolveButtons.length</i> è 2; <i>nativeElement.textContent</i> contiene 'Allarmi attivi'; <i>nativeElement.textContent</i> contiene 'mock-elapsed:2026-03-24T10:00:00.000Z'.	S
TI-211	Contesto <i>AlarmManagement</i> feature integration; caso 'RF60-OBL riesegue <i>initialize</i> quando arriva una richiesta di <i>refresh</i> esterna': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta e <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	Esito atteso: <i>alarmManagementStub.initialize</i> viene invocato 1 volta; <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	S
TI-212	Contesto <i>AlarmFrequencyChartComponent</i> ; caso 'espone metadati <i>chart</i> consistenti': verifica che il valore di <i>component.chartType</i> è 'bar' e <i>component.description.length</i> è maggiore di 0.	Esito atteso: il valore di <i>component.chartType</i> è 'bar'; <i>component.description.length</i> è maggiore di 0.	S
TI-213	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive <i>lg:w-[70%]</i> quando i suggerimenti sono attivi.	Esito atteso: l'elemento <code>[data-test="chart-container"]</code> include la classe <code>lg:w-[70%]</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-214	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S
TI-215	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TI-216	Contesto <i>AlarmFrequencyChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-217	Contesto <i>AlarmFrequencyChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-218	Contesto <i>AlarmsSentResolved-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	Esito atteso: il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-219	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi' : verifica che il contenitore del grafico applica la classe responsive <code>lg:w-[70%]</code> quando i suggerimenti sono attivi.	Esito atteso: l'elemento <code>[data-test="chart-container"]</code> include la classe <code>lg:w-[70%]</code> .	S
TI-220	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti' : verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento <code>[data-test="suggestions-sidebar"]</code> è presente e non risulta null.	S
TI-221	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi' : verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi <code>[data-test="suggestion-item"]</code> sono esattamente 2, in linea con i messaggi forniti.	S
TI-222	Contesto <i>AlarmsSentResolved-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza' : verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S
TI-223	Contesto <i>AlarmsSentResolved-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM' : verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (<code>.lg:w-[30%]</code>) è assente e la query restituisce null.	S
TI-224	Contesto <i>EnergyConsumption-ChartComponent</i> ; caso 'espone metadati chart consistenti' : verifica che il valore di <code>component.chartType</code> è <code>'line'</code> e <code>component.description.length</code> è maggiore di 0.	Esito atteso: il valore di <code>component.chartType</code> è <code>'line'</code> ; <code>component.description.length</code> è maggiore di 0.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-225	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	Esito atteso: l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TI-226	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S
TI-227	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TI-228	Contesto <i>EnergyConsumption-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-229	Contesto <i>EnergyConsumption-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-230	Contesto <i>FallFrequencyChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di <code>component.chartType</code> è 'bar' e <code>component.description.length</code> è maggiore di 0.	Esito atteso: il valore di <code>component.chartType</code> è 'bar'; <code>component.description.length</code> è maggiore di 0.	S
TI-231	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive <code>lg:w-[70%]</code> quando i suggerimenti sono attivi.	Esito atteso: l'elemento <code>[data-test="chart-container"]</code> include la classe <code>lg:w-[70%]</code> .	S
TI-232	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento <code>[data-test="suggestions-sidebar"]</code> è presente e non risulta null.	S
TI-233	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi <code>[data-test="suggestion-item"]</code> sono esattamente 2, in linea con i messaggi forniti.	S
TI-234	Contesto <i>FallFrequencyChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-235	Contesto <i>FallFrequencyChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-236	Contesto <i>PlantAnomaliesChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	Esito atteso: il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S
TI-237	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	Esito atteso: l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TI-238	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S
TI-239	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-240	Contesto <i>PlantAnomaliesChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-241	Contesto <i>PlantAnomaliesChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-242	Contesto <i>PresenceDetection-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	Esito atteso: il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S
TI-243	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	Esito atteso: l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TI-244	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-245	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TI-246	Contesto <i>PresenceDetection-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-247	Contesto <i>PresenceDetection-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-248	Contesto <i>ProlongedPresence-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	Esito atteso: il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S
TI-249	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	Esito atteso: l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TI-250	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-251	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TI-252	Contesto <i>ProlongedPresence-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-253	Contesto <i>ProlongedPresence-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-254	Contesto <i>TemperatureVariation-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'line' e component.description.length è maggiore di 0.	Esito atteso: il valore di component.chartType è 'line'; component.description.length è maggiore di 0.	S
TI-255	Contesto <i>TemperatureVariation-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	Esito atteso: l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TI-256	Contesto <i>TemperatureVariation-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	Esito atteso: l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-257	Contesto <i>TemperatureVariation-sChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	Esito atteso: gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TI-258	Contesto <i>TemperatureVariation-sChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	Esito atteso: è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TI-259	Contesto <i>TemperatureVariation-sChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	Esito atteso: la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TI-260	Contesto <i>ApartmentMonitorComponent</i> ; caso 'crea il componente e sincronizza activeApartmentId dal feed apartment\$': verifica che component è valorizzato e il valore di component.activeApartmentId è 'ap-1'.	Esito atteso: component è valorizzato; il valore di component.activeApartmentId è 'ap-1'; apartmentApiStub.getCurrentApartment viene invocato.	S
TI-261	Contesto <i>ApartmentMonitorComponent</i> ; caso 'onApartmentSelected ignora id vuoto o uguale': verifica che apartmentApiStub.setActivePlantId non viene invocato.	Esito atteso: apartmentApiStub.setActivePlantId non viene invocato.	S
TI-262	Contesto <i>ApartmentMonitorComponent</i> ; caso 'onApartmentSelected cambia appartamento, resetta errore e attiva refresh': verifica che apartmentApiStub.setActivePlantId viene invocato con 'ap-2' e il valore di component.error è ''.	Esito atteso: apartmentApiStub.setActivePlantId viene invocato con 'ap-2'; il valore di component.error è ''.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-263	Contesto <i>ApartmentMonitorComponent</i> ; caso 'refresh realtime azzerare errore e ricaricare apartment\$': verifica che il valore di component.error è " e apartmentApi-Stub.get <code>CurrentApartment</code> .mock.calls.length è maggiore di callsBefore.	Esito atteso: il valore di component.error è "; apartmentApi-Stub.get <code>CurrentApartment</code> .mock.calls.length è maggiore di callsBefore.	S
TI-264	Contesto ApartmentMonitor feature integration; caso 'RF72-OBL inzializza appartamento attivo dal feed e prepara la vista': verifica che component è valorizzato e apartmentApi-Stub.get <code>CurrentApartment</code> viene invocato.	Esito atteso: component è valorizzato; apartmentApi-Stub.get <code>CurrentApartment</code> viene invocato; il valore di component.activeApartmentId è 'ap-1'.	S
TI-265	Contesto ApartmentMonitor feature integration; caso 'RF73-OBL cambio appartamento aggiorna plant attivo e resetta errore': verifica che apartmentApi-Stub.setActivePlantId viene invocato con 'ap-2' e il valore di component.error è ".	Esito atteso: apartmentApi-Stub.setActivePlantId viene invocato con 'ap-2'; il valore di component.error è ".	S
TI-266	Contesto ApartmentMonitor feature integration; caso 'RF74-OBL evento realtime forza refresh del caricamento appartamento': verifica che apartmentApi-Stub.get <code>CurrentApartment</code> .mock.calls.length è maggiore di beforeCalls.	Esito atteso: apartmentApi-Stub.get <code>CurrentApartment</code> .mock.calls.length è maggiore di beforeCalls.	S
TI-267	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe essere creato': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-268	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe inzializzare il servizio al caricamento (ngOnInit)': verifica che mockAlarmService.initialize viene invocato.	Esito atteso: mockAlarmService.initialize viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-269	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe mostrare lo stato di caricamento se vm è null': verifica che <code>compiled.textContent</code> contiene 'Caricamento allarmi in corso'.	Esito atteso: <code>compiled.textContent</code> contiene 'Caricamento allarmi in corso'.	S
TI-270	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe mostrare il messaggio "Nessun allarme" se la lista è vuota': verifica che <code>compiled.textContent</code> contiene 'Nessun allarme attivo al momento'.	Esito atteso: <code>compiled.textContent</code> contiene 'Nessun allarme attivo al momento'.	S
TI-271	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe chiamare <code>resolveAlarm</code> quando <code>onResolve</code> viene invocato': verifica che <code>mockAlarmService.resolveAlarm</code> viene invocato con <code>alarmId</code> .	Esito atteso: <code>mockAlarmService.resolveAlarm</code> viene invocato con <code>alarmId</code> .	S
TI-272	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe ricaricare i dati quando il <code>refresh service</code> emette': verifica che <code>mockAlarmService.initialize</code> viene invocato.	Esito atteso: <code>mockAlarmService.initialize</code> viene invocato.	S
TI-273	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe calcolare le righe correttamente tramite il presenter': verifica che <code>mockPresenter.toRows</code> viene invocato con <code>alarms</code> .	Esito atteso: <code>mockPresenter.toRows</code> viene invocato con <code>alarms</code> .	S
TI-274	Contesto <i>AlarmWidget</i> feature integration; caso 'RF44-OBL <code>init widget</code> invoca <code>initialize service</code> ': verifica che component è valorizzato e <code>alarmManagementStub.initialize</code> viene invocato.	Esito atteso: component è valorizzato; <code>alarmManagementStub.initialize</code> viene invocato.	S
TI-275	Contesto <i>AlarmWidget</i> feature integration; caso 'RF44-OBL click azione allarme delega <code>resolveAlarm</code> ': verifica che <code>alarmManagementStub.resolveAlarm</code> viene invocato con 'a-1'.	Esito atteso: <code>alarmManagementStub.resolveAlarm</code> viene invocato con 'a-1'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-276	Contesto <i>AlarmWidget</i> feature integration; caso 'RF44-OBL refresh esterno reinizializza la tabella': verifica che <code>alarmManagementStub.initialize.mock.calls.length</code> è maggiore di <code>callsBefore</code> .	Esito atteso: <code>alarmManagementStub.initialize.mock.calls.length</code> è maggiore di <code>callsBefore</code> .	S
TI-277	Contesto <i>AnalyticsComponent</i> ; caso 'viene creato': verifica che <code>component</code> è valorizzato.	Esito atteso: <code>component</code> è valorizzato.	S
TI-278	Contesto <i>AnalyticsComponent</i> ; caso 'carica il primo appartamento di default': verifica che <code>mockAnalyticsService.getAllApartments</code> viene invocato e <code>mockAnalyticsService.getAnalytics</code> viene invocato con '1'.	Esito atteso: <code>mockAnalyticsService.getAllApartments</code> viene invocato; <code>mockAnalyticsService.getAnalytics</code> viene invocato con '1'.	S
TI-279	Contesto <i>AnalyticsComponent</i> ; caso 'aggiorna grafici al cambio appartamento': verifica che <code>mockAnalyticsService.getAnalytics</code> viene invocato con '2'.	Esito atteso: <code>mockAnalyticsService.getAnalytics</code> viene invocato con '2'.	S
TI-280	Contesto <i>AnalyticsComponent</i> ; caso 'should return undefined if metric is not found': verifica che <code>result</code> è undefined.	Esito atteso: <code>result</code> è undefined.	S
TI-281	Contesto <i>AnalyticsComponent</i> ; caso 'should find the correct metric in <code>getChartByMetric</code> ': verifica che <code>result</code> è definito e il valore di <code>result?.metric</code> è 'plant-anomalies'.	Esito atteso: <code>result</code> è definito; il valore di <code>result?.metric</code> è 'plant-anomalies'.	S
TI-282	Contesto <i>AnalyticsWidget</i> feature integration; caso 'RF45-OBL inzializza widget e carica analytics del primo appartamento': verifica che <code>analyticsApiStub.getAllApartments</code> viene invocato 1 volta e <code>analyticsApiStub.getAnalytics</code> viene invocato con 'plant-1'.	Esito atteso: <code>analyticsApiStub.getAllApartments</code> viene invocato 1 volta; <code>analyticsApiStub.getAnalytics</code> viene invocato con 'plant-1'.	S
TI-283	Contesto <i>AnalyticsWidget</i> feature integration; caso 'RF45-OBL cambio appartamento aggiorna richiesta analytics': verifica che <code>analyticsApiStub.getAnalytics</code> viene invocato con 'plant-2'.	Esito atteso: <code>analyticsApiStub.getAnalytics</code> viene invocato con 'plant-2'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-284	Contesto <i>AnalyticsWidget</i> feature integration; caso 'RF45-OBL renderizza overview appartamento selezionato ': verifica che overview non è null e <code>fixture.nativeElement.textContent</code> contiene 'Appartamento Monitorato'.	Esito atteso: overview non è null; <code>fixture.nativeElement.textContent</code> contiene 'Appartamento Monitorato'.	S
TI-285	Contesto Dashboard feature integration; caso 'RF43-OBL renderizza la dashboard ': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-286	Contesto Dashboard feature integration; caso 'RF44-OBL visualizza widget allarmi ': verifica che <code>fixture.debugElement.query(By.css('app-alarms-widget'))</code> non è null.	Esito atteso: <code>fixture.debugElement.query(By.css('app-alarms-widget'))</code> non è null.	S
TI-287	Contesto Dashboard feature integration; caso 'RF45-OBL visualizza widget analytics ': verifica che <code>fixture.debugElement.query(By.css('app-analytics-widget'))</code> non è null.	Esito atteso: <code>fixture.debugElement.query(By.css('app-analytics-widget'))</code> non è null.	S
TI-288	Contesto <i>PlantOverviewComponent</i> ; caso '1. Dovrebbe mostrare il nome dell'impianto ': verifica che <code>html.textContent</code> contiene 'Casa Test'.	Esito atteso: <code>html.textContent</code> contiene 'Casa Test'.	S
TI-289	Contesto <i>PlantOverviewComponent</i> ; caso '2. Dovrebbe mostrare i nomi delle stanze ': verifica che <code>html.textContent</code> contiene 'Soggiorno' e <code>html.textContent</code> contiene 'Camera'.	Esito atteso: <code>html.textContent</code> contiene 'Soggiorno'; <code>html.textContent</code> contiene 'Camera'.	S
TI-290	Contesto <i>PlantOverviewComponent</i> ; caso '3. Dovrebbe gestire il caso senza dispositivi ': verifica che <code>html.textContent</code> contiene 'Nessun dispositivo configurato'.	Esito atteso: <code>html.textContent</code> contiene 'Nessun dispositivo configurato'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-291	Contesto <i>PlantOverviewComponent</i> ; caso '4. Non dovrebbe mostrare nulla se apartment è null': verifica che <code>html.querySelector('header')</code> è null.	Esito atteso: <code>html.querySelector('header')</code> è null.	S
TI-292	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>groupRows creates one group for same room id and trackBy returns stable keys</i> ': verifica che il valore di <code>groups.length</code> è 1 e il valore di <code>component.trackByGroup(0, { roomId: 'room-1', roomName: ", rows: [] })</code> è 'room-1'.	Esito atteso: il valore di <code>groups.length</code> è 1; il valore di <code>component.trackByGroup(0, { roomId: 'room-1', roomName: ", rows: [] })</code> è 'room-1'; <code>component.trackByRow(0, mappedRow)</code> contiene 'room-1::device-1::dp-cmd-1'.	S
TI-293	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>getCurrentValue resolves exact datapoint match</i> ': verifica che il valore di <code>exact</code> è 'Off'.	Esito atteso: il valore di <code>exact</code> è 'Off'.	S
TI-294	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>getCurrentValue uses first available value when exact and semantic matches are missing</i> ': verifica che il valore di <code>component.getCurrentValue(fallbackRow)</code> è 'FallbackValue'.	Esito atteso: il valore di <code>component.getCurrentValue(fallbackRow)</code> è 'FallbackValue'.	S
TI-295	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>selected values default to first enum and can be changed</i> ': verifica che il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Cool' e il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Heat'.	Esito atteso: il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Cool'; il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Heat'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-296	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>onWriteRow success toggles executing state and exposes success feedback</i> ': verifica che il valore di <i>component.isRowExecuting(mappedRow)</i> è true e <i>deviceApi-Mock.writeDatapointValue</i> viene invocato con { datapointId: 'dp-cmd-1', value: 'Heat', }.	Esito atteso: il valore di <i>component.isRowExecuting(mappedRow)</i> è true; <i>deviceApi-Mock.writeDatapointValue</i> viene invocato con { datapointId: 'dp-cmd-1', value: 'Heat', }; il valore di <i>component.isRowExecuting(mappedRow)</i> è false; il valore di <i>component.getRowFeedback(mappedRow)?.type</i> è 'success'.	S
TI-297	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>onWriteRow ignores duplicate invocation while row is already executing</i> ': verifica che <i>deviceApi-Mock.writeDatapointValue</i> viene invocato 1 volta.	Esito atteso: <i>deviceApi-Mock.writeDatapointValue</i> viene invocato 1 volta.	S
TI-298	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>onWriteRow with no selectable enum value does not call API</i> ': verifica che <i>deviceApi-Mock.writeDatapointValue</i> non viene invocato.	Esito atteso: <i>deviceApi-Mock.writeDatapointValue</i> non viene invocato.	S
TI-299	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>onWriteRow error path sets error feedback and clears executing state</i> ': verifica che il valore di <i>component.isRowExecuting(mappedRow)</i> è false e il valore di <i>component.getRowFeedback(mappedRow)?.type</i> è 'error'.	Esito atteso: il valore di <i>component.isRowExecuting(mappedRow)</i> è false; il valore di <i>component.getRowFeedback(mappedRow)?.type</i> è 'error'.	S
TI-300	Contesto <i>EndpointTableComponent</i> deep integration; caso ' <i>feedback is auto-cleared after timeout</i> ': verifica che il valore di <i>component.getRowFeedback(mappedRow)?.type</i> è 'success' e <i>component.getRowFeedback(mappedRow)</i> è null.	Esito atteso: il valore di <i>component.getRowFeedback(mappedRow)?.type</i> è 'success'; <i>component.getRowFeedback(mappedRow)</i> è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-301	Contesto <i>EndpointTableComponent</i> deep integration; caso 'active-plant-changed event resets transient state and triggers a refresh' : verifica che il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Cool' e il valore di <code>component.loadError</code> è "".	Esito atteso: il valore di <code>component.getSelectedValue(mappedRow)</code> è 'Cool'; il valore di <code>component.loadError</code> è ""; <code>deviceApi-Mock.getWritableEndpointRows.mock.calls.length</code> è maggiore di <code>callsBefore</code> .	S
TI-302	Contesto <i>EndpointTableComponent</i> deep integration; caso 'polling periodically reloads current values' : verifica che <code>deviceApi-Mock.getCurrentValuePointsByDeviceIds.mock.calls.length</code> è maggiore di <code>callsBefore</code> .	Esito atteso: <code>deviceApi-Mock.getCurrentValuePointsByDeviceIds.mock.calls.length</code> è maggiore di <code>callsBefore</code> .	S
TI-303	Contesto <i>EndpointTableComponent</i> deep integration; caso 'load current values error branch keeps component usable and returns fallback dash' : verifica che il valore di <code>component.getCurrentValue(mappedRow)</code> è '-'	Esito atteso: il valore di <code>component.getCurrentValue(mappedRow)</code> è '-'.	S
TI-304	Contesto <i>EndpointTableComponent</i> ; caso 'creates component and loads writable endpoints' : verifica che component è valorizzato e <code>deviceApi-Mock.getWritableEndpointRows</code> viene invocato 1 volta.	Esito atteso: component è valorizzato; <code>deviceApi-Mock.getWritableEndpointRows</code> viene invocato 1 volta; <code>deviceApi-Mock.getCurrentValuePointsByDeviceIds</code> viene invocato 1 volta.	S
TI-305	Contesto <i>EndpointTableComponent</i> ; caso 'renders mapped endpoint label for known sfeType' : verifica che il valore di <code>component.getEndpointLabel(mappedRow)</code> è 'Comando cambio modalità HVAC'.	Esito atteso: il valore di <code>component.getEndpointLabel(mappedRow)</code> è 'Comando cambio modalità HVAC'.	S
TI-306	Contesto <i>EndpointTableComponent</i> ; caso 'falls back to a runtime humanized label when sfeType is unknown' : verifica che il valore di <code>component.getEndpointLabel(unknownRow)</code> è 'Comando future feature'.	Esito atteso: il valore di <code>component.getEndpointLabel(unknownRow)</code> è 'Comando future feature'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-307	Contesto <i>EndpointTableComponent</i> ; caso 'returns semantic current value by matching Cmd endpoint to State datapoint': verifica che il valore di <code>component.getCurrentValue(mappedRow)</code> è 'Heat'.	Esito atteso: il valore di <code>component.getCurrentValue(mappedRow)</code> è 'Heat'.	S
TI-308	Contesto <i>EndpointTableComponent</i> ; caso 'returns fallback when current datapoint value is unavailable': verifica che il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'	Esito atteso: il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'.	S
TI-309	Contesto DeviceInteraction feature integration; caso 'RF79-OBL carica endpoint scrivibili e valori correnti alla init': verifica che component è valorizzato e <code>deviceApiStub.getWritableEndpointRows</code> viene invocato 1 volta.	Esito atteso: component è valorizzato; <code>deviceApiStub.getWritableEndpointRows</code> viene invocato 1 volta; <code>deviceApiStub.getCurrentValuePointsByDeviceIds</code> viene invocato 1 volta.	S
TI-310	Contesto DeviceInteraction feature integration; caso 'RF80-OBL espone label endpoint leggibili per sfeType noti e ignoti': verifica che il valore di <code>component.getEndpointLabel(mappedRow)</code> è 'Comando cambio modalità HVAC' e il valore di <code>component.getEndpointLabel(unknownRow)</code> è 'Comando future feature'.	Esito atteso: il valore di <code>component.getEndpointLabel(mappedRow)</code> è 'Comando cambio modalità HVAC'; il valore di <code>component.getEndpointLabel(unknownRow)</code> è 'Comando future feature'.	S
TI-311	Contesto DeviceInteraction feature integration; caso 'RF81-OBL risolve valore corrente dal datapoint state corrispondente': verifica che il valore di <code>component.getCurrentValue(mappedRow)</code> è 'Heat' e il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'.	Esito atteso: il valore di <code>component.getCurrentValue(mappedRow)</code> è 'Heat'; il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'.	S
TI-312	Contesto <i>MyVimarPageComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-313	Contesto <i>MyVimarPageComponent</i> ; caso 'carica lo stato account in <i>ngOnInit</i> ': verifica che <i>serviceStub.getLinkedAccount</i> viene invocato 1 volta.	Esito atteso: <i>serviceStub.getLinkedAccount</i> viene invocato 1 volta.	S
TI-314	Contesto <i>MyVimarPageComponent</i> ; caso 'delegates <i>onLinkAccount</i> al service': verifica che <i>serviceStub.initiateOAuth</i> viene invocato 1 volta.	Esito atteso: <i>serviceStub.initiateOAuth</i> viene invocato 1 volta.	S
TI-315	Contesto <i>MyVimarPageComponent</i> ; caso 'onUnlinkAccount richiama il service e attiva refresh': verifica che <i>serviceStub.unlinkAccount</i> viene invocato 1 volta e <i>serviceStub.getLinkedAccount</i> viene invocato 2 volte.	Esito atteso: <i>serviceStub.unlinkAccount</i> viene invocato 1 volta; <i>serviceStub.getLinkedAccount</i> viene invocato 2 volte.	S
TI-316	Contesto <i>MyVimarPageComponent</i> ; caso 'imposta errore quando <i>unlinkAccount fallisce</i> ': verifica che <i>component.error</i> contiene 'Errore durante la rimozione' e il valore di <i>component.isLoading</i> è false.	Esito atteso: <i>component.error</i> contiene 'Errore durante la rimozione'; il valore di <i>component.isLoading</i> è false.	S
TI-317	Contesto <i>MyVimarIntegration</i> feature integration; caso 'RF11-OBL carica stato account in init': verifica che <i>component</i> è valorizzato e <i>serviceStub.getLinkedAccount</i> viene invocato 1 volta.	Esito atteso: <i>component</i> è valorizzato; <i>serviceStub.getLinkedAccount</i> viene invocato 1 volta.	S
TI-318	Contesto <i>MyVimarIntegration</i> feature integration; caso 'RF12-OBL unlink account esegue refresh stato': verifica che <i>serviceStub.unlinkAccount</i> viene invocato 1 volta e <i>serviceStub.getLinkedAccount</i> viene invocato 2 volte.	Esito atteso: <i>serviceStub.unlinkAccount</i> viene invocato 1 volta; <i>serviceStub.getLinkedAccount</i> viene invocato 2 volte.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-319	Contesto MyVimarIntegration feature integration; caso 'RF13-OBL fallback errore quando unlink fallisce': verifica che consoleErrorSpy viene invocato e component.error contiene 'Errore durante la rimozione'.	Esito atteso: consoleErrorSpy viene invocato; component.error contiene 'Errore durante la rimozione'; il valore di component.isLoading è false.	S
TI-320	Contesto <i>NotificationPageComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-321	Contesto <i>NotificationPageComponent</i> ; caso 'renderizza uno stato vuoto accessibile quando non ci sono notifiche': verifica che emptyState è valorizzato e emptyState.textContent contiene 'Nessuna notifica disponibile'.	Esito atteso: emptyState è valorizzato; emptyState.textContent contiene 'Nessuna notifica disponibile'; items ha lunghezza 0.	S
TI-322	Contesto <i>NotificationPageComponent</i> ; caso 'renderizza la lista notifiche e passa i dati ai figli': verifica che summary.textContent contiene '2 notifiche non lette' e items ha lunghezza 2.	Esito atteso: summary.textContent contiene '2 notifiche non lette'; items ha lunghezza 2; items[0].componentInstance.notification() coincide con notificationA; items[1].componentInstance.notification() coincide con notificationB.	S
TI-323	Contesto <i>NotificationPageComponent</i> ; caso 'usa il singolare nel riepilogo quando unreadCount è 1': verifica che summary.textContent contiene '1 notifica non letta'.	Esito atteso: summary.textContent contiene '1 notifica non letta'.	S
TI-324	Contesto <i>NotificationPageComponent</i> ; caso 'aggiorna il DOM quando vm\$ emette un nuovo snapshot coerente': verifica che items ha lunghezza 1 e items ha lunghezza 2.	Esito atteso: items ha lunghezza 1; items ha lunghezza 2.	S
TI-325	Contesto <i>NotificationPageComponent</i> ; caso 'invoca <i>clearAllNotifications</i> quando si clicca Cancella tutte': verifica che notificationServiceStub. <i>clearAllNotifications</i> viene invocato con [notificationA, notificationB,].	Esito atteso: notificationServiceStub. <i>clearAllNotifications</i> viene invocato con [notificationA, notificationB,].	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-326	Contesto <i>NotificationPageComponent</i> ; caso 'evidenzia leggermente la notifica richiesta da query param focus': verifica che il valore di <code>items[0].componentInstance.isHighlighted()</code> è false e il valore di <code>items[1].componentInstance.isHighlighted()</code> è true.	Esito atteso: il valore di <code>items[0].componentInstance.isHighlighted()</code> è false; il valore di <code>items[1].componentInstance.isHighlighted()</code> è true.	S
TI-327	Contesto Notification feature integration; caso 'RF102-OPL visualizza stato vuoto quando non ci sono notifiche': verifica che component è valorizzato e emptyState è valorizzato.	Esito atteso: component è valorizzato; emptyState è valorizzato; emptyState.textContent contiene 'Nessuna notifica disponibile'.	S
TI-328	Contesto Notification feature integration; caso 'RF103-OPL renderizza lista notifiche e clear-all sul servizio': verifica che items ha lunghezza 2 e notificationServiceStub.clearAllNotifications viene invocato con [notificationA, notificationB].	Esito atteso: items ha lunghezza 2; notificationServiceStub.clearAllNotifications viene invocato con [notificationA, notificationB].	S
TI-329	Contesto Notification feature integration; caso 'RF104-OPL focus query param evidenzia la notifica corretta': verifica che il valore di <code>items[0].componentInstance.isHighlighted()</code> è false e il valore di <code>items[1].componentInstance.isHighlighted()</code> è true.	Esito atteso: il valore di <code>items[0].componentInstance.isHighlighted()</code> è false; il valore di <code>items[1].componentInstance.isHighlighted()</code> è true.	S
TI-330	Contesto <i>NotificationTopbarPanel</i> feature integration; caso 'RF105-OPL stato vuoto mostra messaggio e nasconde clear-all': verifica che empty?.textContent contiene 'Nessuna notifica disponibile' e clearAll è null.	Esito atteso: empty?.textContent contiene 'Nessuna notifica disponibile'; clearAll è null.	S
TI-331	Contesto <i>NotificationTopbarPanel</i> feature integration; caso 'RF103-OPL mostra max 6 notifiche e clear-all emette evento': verifica che il valore di items.length è 6 e clearSpy viene invocato 1 volta.	Esito atteso: il valore di items.length è 6; clearSpy viene invocato 1 volta.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-332	Contesto <i>NotificationTopbarPanel</i> feature integration; caso 'RF104-OPL click su notifica e remove emettono gli id corretti': verifica che <i>selectedSpy</i> viene invocato con 'n-2' e <i>removeSpy</i> viene invocato con 'n-2'.	Esito atteso: <i>selectedSpy</i> viene invocato con 'n-2'; <i>removeSpy</i> viene invocato con 'n-2'.	S
TI-333	Contesto <i>NotificationTopbarPanel</i> feature integration; caso 'RF102-OPL applica parsing priorità e titolo visualizzato': verifica che <i>component.getPriority(notifications[0])</i> non è null e il valore di <i>component.getDisplayTitle(notifications[0])</i> è 'Allarme antincendio'.	Esito atteso: <i>component.getPriority(notifications[0])</i> non è null; il valore di <i>component.getDisplayTitle(notifications[0])</i> è 'Allarme antincendio'; il valore di <i>component.getPrioritySymbol(component.getPriority(notifications[0]))</i> è '!'. <i>component.getPriority(notifications[0])</i> non è null e il valore di <i>component.getDisplayTitle(notifications[0])</i> è 'Allarme antincendio'.	S
TI-334	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga a first-access quando isFirstAccess è true': verifica che il valore di <i>component.isLoading</i> è false e <i>component.errorType</i> è null.	Esito atteso: il valore di <i>component.isLoading</i> è false; <i>component.errorType</i> è null; <i>routerMock.navigate</i> viene invocato con ['/auth/first-access'].	S
TI-335	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga a dashboard quando isFirstAccess è false': verifica che <i>routerMock.navigate</i> viene invocato con ['/dashboard'].	Esito atteso: <i>routerMock.navigate</i> viene invocato con ['/dashboard'].	S
TI-336	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga su returnUrl quando presente': verifica che <i>routerMock.navigateByUrl</i> viene invocato con '/vimar-link' e <i>routerMock.navigate</i> non viene invocato con ['/dashboard'].	Esito atteso: <i>routerMock.navigateByUrl</i> viene invocato con '/vimar-link'; <i>routerMock.navigate</i> non viene invocato con ['/dashboard'].	S
TI-337	Contesto <i>AuthBaseComponent</i> ; caso 'handleError imposta USERNAME_OR_PASSWORD_WRONG su 400 e 401': verifica che il valore di <i>component.errorType</i> è <i>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</i> .	Esito atteso: il valore di <i>component.errorType</i> è <i>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</i> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-338	Contesto <i>AuthBaseComponent</i> ; caso ' <i>handleError</i> imposta <code>NEW_PASSWORD_EQUALS_TEMP</code> su 409': verifica che il valore di <code>component.errorType</code> è <code>AuthErrorType.NEW_PASSWORD_EQUALS_TEMP</code> .	Esito atteso: il valore di <code>component.errorType</code> è <code>AuthErrorType.NEW_PASSWORD_EQUALS_TEMP</code> .	S
TI-339	Contesto <i>AuthBaseComponent</i> ; caso ' <i>handleError</i> usa <code>fallback</code> <code>USERNAME_OR_PASSWORD_WRONG</code> per errori non <code>Http</code> ': verifica che il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</code> e il valore di <code>component.isLoading</code> è <code>false</code> .	Esito atteso: il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</code> ; il valore di <code>component.isLoading</code> è <code>false</code> .	S
TI-340	Contesto <i>AuthBaseComponent</i> ; caso ' <i>espone il messaggio per credenziali errate</i> ': verifica che il valore di <code>component.loginErrorMessage</code> è <code>'Utente non trovato: username o password errati.'</code> .	Esito atteso: il valore di <code>component.loginErrorMessage</code> è <code>'Utente non trovato: username o password errati.'</code> .	S
TI-341	Contesto <i>LoginComponent</i> ; caso ' <i>crea il componente</i> ': verifica che <code>component</code> è valorizzato.	Esito atteso: <code>component</code> è valorizzato.	S
TI-342	Contesto <i>LoginComponent</i> ; caso ' <i>non invoca login se il form non rispetta i validator</i> ': verifica che <code>markAllAsTouchedSpy</code> viene invocato e <code>authServiceMock.login</code> non viene invocato.	Esito atteso: <code>markAllAsTouchedSpy</code> viene invocato; <code>authServiceMock.login</code> non viene invocato.	S
TI-343	Contesto <i>LoginComponent</i> ; caso ' <i>invoca login e naviga su dashboard se autenticazione ok</i> ': verifica che <code>authServiceMock.login</code> viene invocato con <code>'mrossi'</code> e <code>routerMock.navigate</code> viene invocato con <code>['/dashboard']</code> .	Esito atteso: <code>authServiceMock.login</code> viene invocato con <code>'mrossi'</code> ; <code>routerMock.navigate</code> viene invocato con <code>['/dashboard']</code> ; <code>component.errorType</code> è <code>null</code> ; il valore di <code>component.isLoading</code> è <code>false</code> .	S
TI-344	Contesto <i>LoginComponent</i> ; caso ' <i>invoca login anche con valori presenti solo nel DOM (autofill browser)</i> ': verifica che <code>authServiceMock.login</code> viene invocato con <code>'mrossi'</code> .	Esito atteso: <code>authServiceMock.login</code> viene invocato con <code>'mrossi'</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-345	Contesto <i>LoginComponent</i> ; caso 'invoca login e naviga su returnUrl se presente': verifica che authServiceMock.login viene invocato con 'mrossi' e routerMock.navigateByUrl viene invocato con '/vimar-link'.	Esito atteso: authServiceMock.login viene invocato con 'mrossi'; routerMock.navigateByUrl viene invocato con '/vimar-link'.	S
TI-346	Contesto <i>LoginComponent</i> ; caso 'imposta errore USERNAME_OR_PASSWORD_WRONG su 401': verifica che il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG e il valore di component.isLoading è false.	Esito atteso: il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG; il valore di component.isLoading è false.	S
TI-347	Contesto <i>LoginComponent</i> ; caso 'espone il messaggio già al primo tentativo fallito': verifica che il valore di component.isLoading è false e il valore di component.loginErrorMessage è 'Utente non trovato: username o password errati.'.	Esito atteso: il valore di component.isLoading è false; il valore di component.loginErrorMessage è 'Utente non trovato: username o password errati.'.	S
TI-348	Contesto <i>LoginComponent</i> ; caso 'renderizza il testo errore quando errorType è valorizzato': verifica che content contiene 'Utente non trovato: username o password errati.'.	Esito atteso: content contiene 'Utente non trovato: username o password errati.'.	S
TI-349	Contesto UserAuth feature integration; caso 'RF1-OBL login valido reindirizza in dashboard': verifica che authServiceStub.login viene invocato con 'mrossi' e routerStub.navigate viene invocato con ['/dashboard'].	Esito atteso: authServiceStub.login viene invocato con 'mrossi'; routerStub.navigate viene invocato con ['/dashboard']; component.errorType è null.	S
TI-350	Contesto UserAuth feature integration; caso 'RF2-OBL login invalido espone errore credenziali': verifica che il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG e component.loginErrorMessage contiene 'username o password errati'.	Esito atteso: il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG; component.loginErrorMessage contiene 'username o password errati'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-351	Contesto <i>UserAuth</i> feature integration; caso 'RF3-OBL form invalido non invoca autenticazione': verifica che <i>AuthServiceStub.login</i> non viene invocato.	Esito atteso: <i>AuthServiceStub.login</i> non viene invocato.	S
TI-352	Contesto <i>UserCreatedDialogComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-353	Contesto <i>UserCreatedDialogComponent</i> ; caso 'mostra nome utente e password temporanea': verifica che <i>textContent</i> contiene 'Mario' e <i>textContent</i> contiene 'mrossi'.	Esito atteso: <i>textContent</i> contiene 'Mario'; <i>textContent</i> contiene 'mrossi'; <i>textContent</i> contiene 'TempPass123'.	S
TI-354	Contesto <i>UserCreatedDialogComponent</i> ; caso 'closeDialog emette evento closed': verifica che <i>closedSpy</i> viene invocato 1 volta.	Esito atteso: <i>closedSpy</i> viene invocato 1 volta.	S
TI-355	Contesto <i>UserCreatedDialogComponent</i> ; caso 'click sul bottone chiudi invoca closeDialog': verifica che <i>closeSpy</i> viene invocato 1 volta.	Esito atteso: <i>closeSpy</i> viene invocato 1 volta.	S
TI-356	Contesto <i>UserListComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-357	Contesto <i>UserListComponent</i> ; caso 'onDelete apre il dialog di conferma': verifica che <i>component.pendingDeleteUser()</i> coincide con <i>user</i> .	Esito atteso: <i>component.pendingDeleteUser()</i> coincide con <i>user</i> .	S
TI-358	Contesto <i>UserListComponent</i> ; caso 'onDeleteConfirm emette deleteUser con id numerico': verifica che <i>emitSpy</i> viene invocato con 1 e <i>emitSpy</i> viene invocato 1 volta.	Esito atteso: <i>emitSpy</i> viene invocato con 1; <i>emitSpy</i> viene invocato 1 volta; <i>component.pendingDeleteUser()</i> è null.	S
TI-359	Contesto <i>UserListComponent</i> ; caso 'onDeleteCancel non emette deleteUser e chiude il dialog': verifica che <i>emitSpy</i> non viene invocato e <i>component.pendingDeleteUser()</i> è null.	Esito atteso: <i>emitSpy</i> non viene invocato; <i>component.pendingDeleteUser()</i> è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-360	Contesto UserManagement feature integration; caso 'RF16-OBL inizializza pagina e carica elenco utenti': verifica che component è valorizzato e <i>getUsersStub</i> viene invocato 1 volta.	Esito atteso: component è valorizzato; <i>getUsersStub</i> viene invocato 1 volta; il valore di <i>component.iscreateFormOpen()</i> è false.	S
TI-361	Contesto UserManagement feature integration; caso 'RF17-OBL submit creazione utente aggiorna stato e aggiorna la lista': verifica che <i>createUserStub</i> viene invocato con { name: 'Mario', surname: 'Rossi', username: 'mrossi', } e il valore di <i>component.createdResponse()?.tempPassword</i> è 'TempPass123'.	Esito atteso: <i>createUserStub</i> viene invocato con { name: 'Mario', surname: 'Rossi', username: 'mrossi', }; il valore di <i>component.createdResponse()?.tempPassword</i> è 'TempPass123'; <i>getUsersStub</i> viene invocato 2 volte.	S
TI-362	Contesto UserManagement feature integration; caso 'RF18-OBL errore 409 in creazione espone errore username in uso': verifica che il valore di <i>component.formError()</i> è <i>UserManagementErrorType.USERNAME_ALREADY_IN_USE</i> .	Esito atteso: il valore di <i>component.formError()</i> è <i>UserManagementErrorType.USERNAME_ALREADY_IN_USE</i> .	S
TI-363	Contesto <i>AssignWardDialogComponent</i> ; caso 'should create': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-364	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe resettare plantId a null in <i>ngOnInit</i> ': verifica che <i>component.form.controls.plantId.value</i> è null.	Esito atteso: <i>component.form.controls.plantId.value</i> è null.	S
TI-365	Contesto <i>AssignWardDialogComponent</i> ; caso 'non dovrebbe emettere submitted se il form è invalido': verifica che il valore di <i>component.form.invalid</i> è true e il valore di <i>component.form.controls.plantId.touched</i> è true.	Esito atteso: il valore di <i>component.form.invalid</i> è true; il valore di <i>component.form.controls.plantId.touched</i> è true; <i>submittedSpy</i> non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-366	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe emettere submitted con plantId quando il form è valido': verifica che submittedSpy viene invocato con { plantId : '101' } e submittedSpy viene invocato 1 volta.	Esito atteso: submittedSpy viene invocato con { plantId : '101' }; submittedSpy viene invocato 1 volta.	S
TI-367	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe emettere cancelled quando <i>onCancel</i> viene chiamato': verifica che cancelledSpy viene invocato 1 volta.	Esito atteso: cancelledSpy viene invocato 1 volta.	S
TI-368	Contesto <i>WardManagementPageComponent</i> ; caso 'should create ': verifica che component è valorizzato.	Esito atteso: component è valorizzato.	S
TI-369	Contesto <i>WardManagementPageComponent</i> ; caso 'ngOnInit dovrebbe chiamare loadWards e sincronizzare snapshots/error ': verifica che storeStub.loadWards viene invocato 1 volta e component.wardsSnapshot() coincide con [].	Esito atteso: storeStub.loadWards viene invocato 1 volta; component.wardsSnapshot() coincide con []; component.snackbarMessage() è null; component.wardsSnapshot() ha lunghezza 2.	S
TI-370	Contesto <i>WardManagementPageComponent</i> ; caso 'quando wards diventa vuoto resetta selezioni e step mobile ': verifica che component.selectedWardId() è null e component.selectedApartmentId() è null.	Esito atteso: component.selectedWardId() è null; component.selectedApartmentId() è null; il valore di component.mobileStep() è 'wards'.	S
TI-371	Contesto <i>WardManagementPageComponent</i> ; caso 'selectedApartment si riallinea al ward selezionato': verifica che component.selectedApartment() coincide con { id : '101', name : 'App. 101' } e component.selectedApartment() è null.	Esito atteso: component.selectedApartment() coincide con { id : '101', name : 'App. 101' }; component.selectedApartment() è null.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-372	Contesto <i>WardManagementPage-Component</i> ; caso <i>'ngOnDestroy'</i> dovrebbe interrompere aggiornamenti da stream': verifica che il valore di <code>component.snackbarMessage()</code> è 'Errore iniziale'.	Esito atteso: il valore di <code>component.snackbarMessage()</code> è 'Errore iniziale'.	S
TI-373	Contesto <i>WardManagementPage-Component</i> ; caso <i>'dismissSnackbar'</i> dovrebbe azzerare il messaggio': verifica che <code>component.snackbarMessage()</code> è null.	Esito atteso: <code>component.snackbarMessage()</code> è null.	S
TI-374	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onCreateWard'</i> dovrebbe aprire dialog in create mode': verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.wardDialogMode()</code> è 'create'.	Esito atteso: il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.wardDialogMode()</code> è 'create'.	S
TI-375	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onCreateWardSubmit'</i> dovrebbe delegare a store e chiudere dialog': verifica che <code>storeStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> e <code>storeStub.createWard</code> viene invocato 1 volta.	Esito atteso: <code>storeStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> ; <code>storeStub.createWard</code> viene invocato 1 volta; il valore di <code>component.wardDialogMode()</code> è 'closed'.	S
TI-376	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onEditWard'</i> e <i>'onEditWardSubmit'</i> dovrebbero aggiornare il reparto selezionato': verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.wardDialogMode()</code> è 'edit'.	Esito atteso: il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.wardDialogMode()</code> è 'edit'; <code>storeStub.updateWard</code> viene invocato con 1; il valore di <code>component.wardDialogMode()</code> è 'closed'.	S
TI-377	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onEditWardSubmit'</i> non dovrebbe chiamare store se <code>selectedWard</code> è null': verifica che <code>storeStub.updateWard</code> non viene invocato.	Esito atteso: <code>storeStub.updateWard</code> non viene invocato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-378	Contesto <i>WardManagementPage-Component</i> ; caso ' <i>onCloseWardDialog</i> dovrebbe chiudere dialog e pulire selezione': verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.wardDialogMode()</code> è <code>'closed'</code> .	Esito atteso: il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.wardDialogMode()</code> è <code>'closed'</code> .	S
TI-379	Contesto <i>WardManagementPage-Component</i> ; caso ' <i>onCloseAssignOperatorDialog</i> e <i>onCloseAssignPlantDialog</i> ripuliscono stato di caricamento e lookup': verifica che <code>component.operatorWardId()</code> è <code>null</code> e <code>component.availableOperatorsFromFetch()</code> è <code>null</code> .	Esito atteso: <code>component.operatorWardId()</code> è <code>null</code> ; <code>component.availableOperatorsFromFetch()</code> è <code>null</code> ; il valore di <code>component.isLoadingAvailableOperators()</code> è <code>false</code> ; <code>component.plantWardId()</code> è <code>null</code> .	S
TI-380	Contesto <i>WardManagementPage-Component</i> ; caso ' <i>selectWard</i> dovrebbe aggiornare ward attivo e step mobile': verifica che il valore di <code>component.selectedWardId()</code> è <code>2</code> e il valore di <code>component.mobileStep()</code> è <code>'apartments'</code> .	Esito atteso: il valore di <code>component.selectedWardId()</code> è <code>2</code> ; il valore di <code>component.mobileStep()</code> è <code>'apartments'</code> .	S
TI-381	Contesto <i>WardManagementPage-Component</i> ; caso ' <i>selectWard</i> sul ward già selezionato mantiene la selezione dell'appartamento': verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.selectedApartmentId()</code> è <code>'101'</code> .	Esito atteso: il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.selectedApartmentId()</code> è <code>'101'</code> ; il valore di <code>component.mobileStep()</code> è <code>'apartments'</code> .	S
TI-382	Contesto <i>WardManagementPage-Component</i> ; caso ' <i>selectApartment</i> e step helpers dovrebbero aggiornare lo stato ui': verifica che il valore di <code>component.selectedApartmentId()</code> è <code>'101'</code> e il valore di <code>component.mobileStep()</code> è <code>'wards'</code> .	Esito atteso: il valore di <code>component.selectedApartmentId()</code> è <code>'101'</code> ; il valore di <code>component.mobileStep()</code> è <code>'wards'</code> ; il valore di <code>component.mobileStep()</code> è <code>'apartments'</code> ; il valore di <code>component.mobileStep()</code> è <code>'operators'</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-383	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onEditSelectedWard non dovrebbe fare nulla senza ward selezionato'</i> : verifica che <code>component.selectedWardId()</code> è null e il valore di <code>component.wardDialogMode()</code> è <i>'closed'</i> .	Esito atteso: <code>component.selectedWardId()</code> è null; il valore di <code>component.wardDialogMode()</code> è <i>'closed'</i> .	S
TI-384	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onAssignOperatorSubmit non dovrebbe chiamare store senza wardId'</i> : verifica che <code>storeStub.assignOperator</code> non viene invocato.	Esito atteso: <code>storeStub.assignOperator</code> non viene invocato.	S
TI-385	Contesto <i>WardManagementPage-Component</i> ; caso <i>'flow plant dovrebbe impostare wardId, submit e chiudere'</i> : verifica che il valore di <code>component.plantWardId()</code> è 2 e <code>storeStub.getAvailablePlantsForWard</code> viene invocato con 2.	Esito atteso: il valore di <code>component.plantWardId()</code> è 2; <code>storeStub.getAvailablePlantsForWard</code> viene invocato con 2; <code>component.availablePlants()</code> coincide con <code>{ id: '103', name: 'App. 103' }</code> ; <code>storeStub.assignPlant</code> viene invocato con 2.	S
TI-386	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onAssignPlantSubmit non dovrebbe chiamare store senza wardId'</i> : verifica che <code>storeStub.assignPlant</code> non viene invocato.	Esito atteso: <code>storeStub.assignPlant</code> non viene invocato.	S
TI-387	Contesto <i>WardManagementPage-Component</i> ; caso <i>'confirmState + onConfirmDialogConfirmed dovrebbero gestire i 3 rami'</i> : verifica che <code>component.confirmState()</code> coincide con <code>{ kind: 'delete-ward', wardId: 1 }</code> e <code>storeStub.deleteWard</code> viene invocato con 1.	Esito atteso: <code>component.confirmState()</code> coincide con <code>{ kind: 'delete-ward', wardId: 1 }</code> ; <code>storeStub.deleteWard</code> viene invocato con 1; <code>component.confirmState()</code> è null; <code>component.confirmState()</code> coincide con <code>{ kind: 'remove-operator', wardId: 1, userId: 2, }</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-388	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onConfirmDialogConfirmed non dovrebbe fare nulla con stato nullo'</i> : verifica che <i>storeStub.deleteWard</i> non viene invocato e <i>storeStub.removeOperator</i> non viene invocato.	Esito atteso: <i>storeStub.deleteWard</i> non viene invocato; <i>storeStub.removeOperator</i> non viene invocato; <i>storeStub.removePlant</i> non viene invocato.	S
TI-389	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onConfirmDialogCancelled dovrebbe pulire confirmState'</i> : verifica che <i>component.confirmState()</i> è null.	Esito atteso: <i>component.confirmState()</i> è null.	S
TI-390	Contesto <i>WardManagementPage-Component</i> ; caso <i>'getMessage dovrebbe restituire il testo corretto per ogni stato'</i> : verifica che il valore di <i>component.getMessage()</i> è <i>'Confermi questa operazione?'</i> e il valore di <i>component.getLabel()</i> è <i>'Conferma'</i> .	Esito atteso: il valore di <i>component.getMessage()</i> è <i>'Confermi questa operazione?'</i> ; il valore di <i>component.getLabel()</i> è <i>'Conferma'</i> ; il valore di <i>component.getMessage()</i> è <i>'Confermi l'eliminazione del reparto?'</i> ; il valore di <i>component.getMessage()</i> è <i>'Confermi la rimozione dell'operatore dal reparto?'</i> .	S
TI-391	Contesto <i>WardManagementPage-Component</i> ; caso <i>'getDisplayOperatorName dovrebbe comporre e trimmare nome e cognome'</i> : verifica che il valore di <i>component.getOperatorDisplayName('Mario', 'Rossi')</i> è <i>'Mario Rossi'</i> e il valore di <i>component.getOperatorDisplayName('Mario', '')</i> è <i>'Mario'</i> .	Esito atteso: il valore di <i>component.getOperatorDisplayName('Mario', 'Rossi')</i> è <i>'Mario Rossi'</i> ; il valore di <i>component.getOperatorDisplayName('Mario', '')</i> è <i>'Mario'</i> .	S
TI-392	Contesto <i>WardManagementPage-Component</i> ; caso <i>'availablePlants dovrebbe deduplicare e escludere quelli già assegnati al ward selezionato'</i> : verifica che <i>component.availablePlants()</i> coincide con [].	Esito atteso: <i>component.availablePlants()</i> coincide con [].	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-393	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onAssignPlant non usa fallback locale se fetch ad-hoc fallisce'</i> : verifica che <code>component.availablePlants()</code> coincide con [] e il valore di <code>component.isLoadingAvailablePlants()</code> è false.	Esito atteso: <code>component.availablePlants()</code> coincide con []; il valore di <code>component.isLoadingAvailablePlants()</code> è false.	S
TI-394	Contesto <i>WardManagement feature integration</i> ; caso <i>'RF27-OBL mantiene coerente la selezione del ward e aggiorna il nome in entrambe le sezioni'</i> : verifica che <code>wardApiStub.getWards</code> viene invocato 1 volta e <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene <code>ward1.name</code> .	Esito atteso: <code>wardApiStub.getWards</code> viene invocato 1 volta; <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene <code>ward1.name</code> ; il valore di <code>wardButtons.length</code> è 2; <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene <code>ward2.name</code> .	S
TI-395	Contesto <i>WardManagement feature integration</i> ; caso <i>'RF28-OBL aggiorna il nome del ward e sincronizza lista e pannello centrale'</i> : verifica che <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene <code>ward1.name</code> e <code>getDialog()?.textContent</code> contiene 'Modifica reparto'.	Esito atteso: <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene <code>ward1.name</code> ; <code>getDialog()?.textContent</code> contiene 'Modifica reparto'; <code>wardApiStub.updateWard</code> viene invocato con 1; <code>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')</code> contiene 'Cardiologia A'.	S
TI-396	Contesto <i>WardManagement feature integration</i> ; caso <i>'RF29-OBL dopo la creazione mantiene cliccabile il ward appena creato'</i> : verifica che <code>getDialog()?.textContent</code> contiene 'Crea reparto' e <code>wardApiStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> .	Esito atteso: <code>getDialog()?.textContent</code> contiene 'Crea reparto'; <code>wardApiStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> ; il valore di <code>wardButtons.length</code> è 3; <code>wardButtons[2].textContent</code> contiene 'Oncologia'.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-397	Contesto WardManagement feature integration; caso 'RF30-OBL assegna un operatore dal dialog e invoca il flusso di assegnazione': verifica che <i>getDialog()?.textContent</i> contiene 'Assegna operatore sanitario' e <i>wardApiStub.getAvailableOperators</i> viene invocato 1 volta.	Esito atteso: <i>getDialog()?.textContent</i> contiene 'Assegna operatore sanitario'; <i>wardApiStub.getAvailableOperators</i> viene invocato 1 volta; <i>assignOperatorDialog</i> è valorizzato; <i>wardApiStub.assignOperatorToWard</i> viene invocato con 1.	S
TI-398	Contesto WardManagement feature integration; caso 'RF31-OBL non mostra appartamenti assegnabili se fetch disponibile fallisce': verifica che <i>getDialog()?.textContent</i> contiene 'Assegna appartamento' e <i>wardApiStub.getAvailablePlants</i> viene invocato 1 volta.	Esito atteso: <i>getDialog()?.textContent</i> contiene 'Assegna appartamento'; <i>wardApiStub.getAvailablePlants</i> viene invocato 1 volta; <i>apartmentOptions</i> non contiene 'App. 102'; <i>apartmentOptions</i> non contiene 'App. 101'.	S
TI-399	Contesto WardManagement feature integration; caso 'RF32-OBL rimuove operatore e appartamento passando dal confirm dialog': verifica che <i>getDialog()?.textContent</i> contiene 'rimozione dell\{\}'operatore' e <i>wardApiStub.removeOperatorFromWard</i> viene invocato con 1.	Esito atteso: <i>getDialog()?.textContent</i> contiene 'rimozione dell\{\}'operatore'; <i>wardApiStub.removeOperatorFromWard</i> viene invocato con 1; <i>getDialog()?.textContent</i> contiene 'rimozione dell\{\}'appartamento'; <i>wardApiStub.removePlantFromWard</i> viene invocato con 1.	S
TI-400	Contesto WardManagement feature integration; caso 'RF33-OBL elimina un ward dopo conferma e aggiorna la selezione': verifica che <i>getDialog()?.textContent</i> contiene 'eliminazione del reparto' e <i>wardApiStub.deleteWard</i> viene invocato con 1.	Esito atteso: <i>getDialog()?.textContent</i> contiene 'eliminazione del reparto'; <i>wardApiStub.deleteWard</i> viene invocato con 1; il valore di <i>getWardButtons().length</i> è 1; <i>nativeElement.querySelector('section[aria-label="Dettagli reparto e appartamenti"] h2')?.textC...</i> contiene <i>ward2.name</i> .	S
TI-401	Contesto <i>Auth Integration Test</i> ; caso 'should login and set refresh cookie with real auth flow': verifica che <i>response.body</i> contiene la proprietà <i>accessToken</i> e il set-cookie header è definito.	Esito atteso: <i>response.body</i> contiene la proprietà <i>accessToken</i> ; il set-cookie header è definito; il valore di HTTP status è 201.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-402	Contesto <i>Auth Integration Test</i> ; caso 'should execute first-login end to end and persist credentials change' : verifica che la password dell'utente viene aggiornata nel database e il valore di <code>first_access</code> diventa false.	Esito atteso: la password dell'utente viene aggiornata nel database; il valore di <code>first_access</code> diventa false; il valore di HTTP status è 201.	S
TI-403	Contesto <i>Auth Integration Test</i> ; caso 'should refresh token from cookie generated by /auth/login' : verifica che il refresh endpoint ritorna un nuovo <code>accessToken</code> e utilizza il refresh cookie.	Esito atteso: il refresh endpoint ritorna un nuovo <code>accessToken</code> ; il nuovo token è valido; il valore di HTTP status è 201.	S
TI-404	Contesto <i>Auth Integration Test</i> ; caso 'should return 401 on /auth/refresh without cookie' : verifica che il refresh endpoint restituisce un errore quando il cookie non è presente.	Esito atteso: il valore di HTTP status è 401; la risposta non contiene <code>accessToken</code> .	S
TI-405	Contesto <i>Alarms Integration Test</i> ; caso 'should create alarm rule through real controller/service/repository' : verifica che il POST <code>/alarm-rules</code> crea una nuova regola di allarme nel database.	Esito atteso: <code>response.body</code> contiene la proprietà <code>name</code> con valore <code>'Too Hot'</code> ; il numero di regole nel database aumenta di 1; il valore di HTTP status è 201.	S
TI-406	Contesto <i>Alarms Integration Test</i> ; caso 'should list alarm rules' : verifica che il GET <code>/alarm-rules</code> ritorna una lista di regole di allarme.	Esito atteso: <code>response.body</code> è un array non vuoto; le regole contengono la proprietà <code>id</code> ; il valore di HTTP status è 200.	S
TI-407	Contesto <i>Alarms Integration Test</i> ; caso 'should get alarm rule by id' : verifica che il GET <code>/alarm-rules/:id</code> ritorna una specifica regola di allarme.	Esito atteso: <code>response.body</code> contiene la proprietà <code>id</code> con valore cercato; la regola contiene tutte le proprietà richieste; il valore di HTTP status è 200.	S
TI-408	Contesto <i>Alarms Integration Test</i> ; caso 'should list managed alarm events for a user' : verifica che il GET <code>/alarm-events/managed/:userId/:Ward/:offset</code> ritorna gli allarmi risolti.	Esito atteso: <code>response.body</code> è un array; tutti gli elementi hanno <code>resolution_time</code> valorizzato; il valore di HTTP status è 200.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-409	Contesto <i>Alarms Integration Test</i> ; caso 'should list unmanaged alarm events for a user' : verifica che il GET /alarm-events/unmanaged/:userId/:Ward/:offset ritorna gli allarmi attivi.	Esito atteso: response.body è un array; gli allarmi hanno resolution_time null; il valore di HTTP status è 200.	S
TI-410	Contesto <i>Alarms Integration Test</i> ; caso 'should resolve alarm event and emit resolution flow' : verifica che il PATCH /alarm-events/resolve risolve un allarme e aggiorna il database.	Esito atteso: l'allarme nel database ha resolution_time valorizzato; l'user_id e user_username vengono salvati; il valore di HTTP status è 200.	S
TI-411	Contesto <i>Alarms Integration Test</i> ; caso 'should update and delete alarm rules through real service' : verifica che il PUT e DELETE aggiornano/cancellano le regole nel database.	Esito atteso: l'aggiornamento modifica le proprietà della regola; la cancellazione rimuove la regola dal database; il valore di HTTP status è appropriato.	S
TI-412	Contesto <i>Alarms Integration Test</i> ; caso 'should execute checkAlarmRule flow and create alarm event' : verifica che il flusso di controllo regola allarme crea un evento di allarme quando la soglia viene superata.	Esito atteso: il numero di eventi allarme aumenta di 1; il nuovo evento ha i parametri corretti; il valore del alarm_rule_id corrisponde alla regola.	S
TI-413	Contesto <i>Users Integration Test</i> ; caso 'should list users using real controller/service/adapters/repository chain' : verifica che il GET /users ritorna l'elenco di tutti gli utenti.	Esito atteso: response.body è un array con 2 elementi; il primo utente contiene le proprietà id, username, surname, name; il valore di HTTP status è 200.	S
TI-414	Contesto <i>Users Integration Test</i> ; caso 'should create user and return generated tempPassword' : verifica che il POST /users crea un nuovo utente e genera una password temporanea.	Esito atteso: response.body contiene le proprietà id, username, tempPassword; il nuovo utente viene salvato nel database; il valore di HTTP status è 201.	S
TI-415	Contesto <i>Users Integration Test</i> ; caso 'should validate payload on create user' : verifica che il POST /users con payload non valido ritorna un errore di validazione.	Esito atteso: il valore di HTTP status è 400; la risposta contiene i dettagli dell'errore di validazione.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-416	Contesto <i>Wards Integration Test</i> ; caso 'should create and list wards with real service and adapters': verifica che il POST /wards crea un nuovo reparto e il GET /wards ritorna la lista aggiornata.	Esito atteso: il numero di reparti aumenta di 1; il nuovo reparto è presente nella lista dei reparti; il valore di HTTP status è 201 per POST e 200 per GET.	S
TI-417	Contesto <i>Wards Integration Test</i> ; caso 'should add and remove user from ward': verifica che il POST /wards-users-relationships aggiunge un utente al reparto e il DELETE lo rimuove.	Esito atteso: dopo POST, l'utente è associato al reparto; dopo DELETE, l'utente non è più associato al reparto; il valore di HTTP status è appropriato.	S
TI-418	Contesto <i>Wards Integration Test</i> ; caso 'should add and list plants for a ward': verifica che il POST /wards-plants-relationships aggiunge un impianto al reparto e il GET ritorna la lista.	Esito atteso: il numero di impianti associati al reparto aumenta di 1; il nuovo impianto è presente nella lista; il valore di HTTP status è appropriato.	S
TI-419	Contesto <i>Plant Integration Test</i> ; caso 'should retrieve plant by id with real repository': verifica che il GET /plant con plantid query param ritorna l'impianto corretto.	Esito atteso: response.body contiene le proprietà id, name; il valore di HTTP status è 200; l'impianto ritornato corrisponde a quello cercato.	S
TI-420	Contesto <i>Plant Integration Test</i> ; caso 'should return only available plants': verifica che il GET /plant/available ritorna solo gli impianti disponibili.	Esito atteso: response.body è un array che contiene solo impianti disponibili; il valore di HTTP status è 200.	S
TI-421	Contesto <i>Plant Integration Test</i> ; caso 'should return all plants': verifica che il GET /plant/all ritorna tutti gli impianti nel sistema.	Esito atteso: response.body è un array con 2 elementi; il valore di HTTP status è 200.	S
TI-422	Contesto <i>Plant Integration Test</i> ; caso 'should return 404 when plantid query param is missing': verifica che il GET /plant senza parametro ritorna un errore 404.	Esito atteso: il valore di HTTP status è 404; la risposta contiene un messaggio di errore.	S
TI-423	Contesto <i>Plant Integration Test</i> ; caso 'should return fallback message when no available plants are found': verifica che il GET /plant/available ritorna un messaggio fallback quando non ci sono impianti disponibili.	Esito atteso: response.body contiene il messaggio 'No available plants found'; il valore di status code nel messaggio è 202.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-424	Contesto <i>Analytics Integration Test</i> ; caso <code>'should retrieve all analytics given the plantId'</code> : verifica che il GET <code>/analytics/:plantId</code> ritorna una lista di grafici di analisi.	Esito atteso: <code>response.body</code> è un array non vuoto; i grafici contengono le proprietà <code>metric</code> , <code>unit</code> , <code>labels</code> , <code>series</code> ; il valore di HTTP status è 200.	S
TI-425	Contesto <i>Analytics Integration Test</i> ; caso <code>'should return one PlotDto per strategy'</code> : verifica che ogni strategia di analisi ritorna un grafico.	Esito atteso: il numero di grafici corrisponde al numero di strategie; ogni grafico ha proprietà <code>distinct</code> ; il valore di HTTP status è 200.	S
TI-426	Contesto <i>Analytics Integration Test</i> ; caso <code>'should return 500 when all strategies fail'</code> : verifica che quando tutte le strategie falliscono il sistema ritorna un errore 500.	Esito atteso: il valore di HTTP status è 500; la risposta contiene un messaggio di errore.	S
TI-427	Contesto <i>Analytics Integration Test</i> ; caso <code>'should return 404 when plantId query param is missing'</code> : verifica che il GET <code>/analytics</code> senza <code>plantId</code> ritorna un errore 404.	Esito atteso: il valore di HTTP status è 404.	S
TI-428	Contesto <i>Analytics Integration Test</i> ; caso <code>'should throw when suggestion service receives no suggestion from LLM port'</code> : verifica che quando l'LLM non ritorna suggerimenti il servizio genera un errore.	Esito atteso: il servizio genera un errore con messaggio appropriato.	S
TI-429	Contesto <i>Cache Integration Test</i> ; caso <code>'should process webhook and write plant cache using real cache flow'</code> : verifica che il POST <code>/cache/update</code> processa il webhook e scrive la cache dell'impianto.	Esito atteso: <code>response.body</code> contiene <code>success: true</code> ; il valore di HTTP status è 202; la cache dell'impianto viene scritta nel sistema.	S
TI-430	Contesto <i>Cache Integration Test</i> ; caso <code>'should refresh all plants on fetched.tokens event'</code> : verifica che l'evento <code>fetched.tokens</code> aggiorna la cache di tutti gli impianti.	Esito atteso: le cache di tutti gli impianti vengono aggiornate; il flusso completa correttamente.	S
TI-431	Contesto <i>Cache Integration Test</i> ; caso <code>'should reject updateCache use case when plantId is missing'</code> : verifica che il use case rifiuta la richiesta quando <code>plantId</code> manca.	Esito atteso: il sistema genera un errore con messaggio <code>'PlantId is null'</code> .	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-432	Contesto <i>Cache Integration Test</i> ; caso <code>'should handle webhook update failure with Error without crashing'</code> : verifica che gli errori nei webhook vengono gestiti senza bloccare il sistema.	Esito atteso: <code>response.body</code> contiene <code>success: true</code> ; il sistema continua a funzionare; il valore di <code>HTTP status</code> è 202.	S
TI-433	Contesto <i>Cache Integration Test</i> ; caso <code>'should skip duplicate cache sync events while one sync is already running'</code> : verifica che gli eventi di sincronizzazione duplicati vengono ignorati mentre una sincronizzazione è in corso.	Esito atteso: solo una sincronizzazione è eseguita; gli eventi duplicati non creano nuove sincronizzazioni.	S
TI-434	Contesto <i>Notifications Integration Test</i> ; caso <code>'should handle alarm.activated event and persist notification'</code> : verifica che l'evento <code>alarm.activated</code> crea e persiste una notifica nel database.	Esito atteso: la notifica viene scritta nel database; il valore di <code>HTTP status</code> è appropriato; il mock pool è invocato correttamente.	S
TI-435	Contesto <i>Notifications Integration Test</i> ; caso <code>'should handle alarm.resolved event and persist notification'</code> : verifica che l'evento <code>alarm.resolved</code> crea una notifica di risoluzione.	Esito atteso: la notifica viene scritta nel database; l'evento viene elaborato correttamente.	S
TI-436	Contesto <i>Notifications Integration Test</i> ; caso <code>'should ignore invalid alarm payload without throwing'</code> : verifica che i payload non validi vengono ignorati senza errori.	Esito atteso: il sistema non genera eccezioni; nessuna query viene eseguita per <code>payload null</code> .	S
TI-437	Contesto <i>Notifications Integration Test</i> ; caso <code>'should reject ward notification when alarmEventId is missing'</code> : verifica che la notifica viene rifiutata quando <code>alarmEventId</code> manca.	Esito atteso: il sistema genera un messaggio di errore appropriato.	S
TI-438	Contesto <i>Notifications Integration Test</i> ; caso <code>'should reject resolution notification when wardId is missing'</code> : verifica che la notifica di risoluzione viene rifiutata quando <code>wardId</code> manca.	Esito atteso: il sistema genera un messaggio di errore appropriato.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-439	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should return account status from real token service and cache repository': verifica che il GET /my-vimar/account ritorna lo stato dell'account MyVimar collegato.	Esito atteso: response.body contiene le proprietà isLinked e email; il valore di HTTP status è 200.	S
TI-440	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should prepare and authorize OAuth ticket end-to-end': verifica che il flusso OAuth completo funziona: preparazione -> autorizzazione.	Esito atteso: il POST /api/auth/prepare-oauth ritorna un ticket; il GET /api/auth/authorize reindirizza al servizio OAuth; il valore di HTTP status è 302 per il redirect.	S
TI-441	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should handle callback and persist tokens with real token flow': verifica che il callback OAuth persiste i token nel sistema.	Esito atteso: il GET /my-vimar/callback elabora il codice OAuth e persiste i token; il valore di HTTP status è 302 per il redirect.	S
TI-442	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should disconnect account by clearing token cache': verifica che il DELETE /my-vimar/account disconnette l'account e pulisce la cache.	Esito atteso: il valore di HTTP status è 200; il token viene rimosso dalla cache.	S
TI-443	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should return 401 when prepare-oauth user identity is invalid': verifica che la preparazione OAuth fallisce con identità utente non valida.	Esito atteso: il valore di HTTP status è 401; la risposta non contiene un ticket.	S
TI-444	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should return 400 when authorize redirect_url is missing': verifica che l'autorizzazione fallisce quando redirect_url manca.	Esito atteso: il valore di HTTP status è 400.	S
TI-445	Contesto <i>ApiAuthVimar Integration Test</i> ; caso 'should return 401 when authorize ticket is invalid': verifica che l'autorizzazione fallisce con ticket non valido.	Esito atteso: il valore di HTTP status è 401.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-446	Contesto <i>Subscription Integration Test</i> ; caso 'should refresh subscriptions for all plants on cache.all.updated' : verifica che l'evento <code>cache.all.updated</code> aggiorna le sottoscrizioni di tutti gli impianti.	Esito atteso: le sottoscrizioni di entrambi gli impianti vengono aggiornate; il flusso completa correttamente.	S
TI-447	Contesto <i>Subscription Integration Test</i> ; caso 'should refresh subscriptions for one plant on cache.updated' : verifica che l'evento <code>cache.updated</code> aggiorna la sottoscrizione di un singolo impianto.	Esito atteso: solo la sottoscrizione dell'impianto specificato viene aggiornata.	S
TI-448	Contesto <i>Subscription Integration Test</i> ; caso 'should execute node cron flow with real service' : verifica che il flusso cron dei nodi viene eseguito correttamente.	Esito atteso: le sottoscrizioni dei nodi vengono aggiornate; il servizio completa senza errori.	S
TI-449	Contesto <i>Subscription Integration Test</i> ; caso 'should skip duplicate full-cache refresh event while another refresh is running' : verifica che gli eventi di refresh duplicati vengono ignorati quando un refresh è in corso.	Esito atteso: solo un refresh è eseguito; gli eventi duplicati non creano nuovi refresh.	S
TI-450	Contesto <i>Subscription Integration Test</i> ; caso 'should return false when datapoint refresh fails in refreshAllSubscription' : verifica che il fallimento del refresh datapoint causa il ritorno di false.	Esito atteso: il flusso continua; il valore di ritorno è false.	S
TI-451	Contesto <i>Device Integration Test</i> ; caso 'should retrieve a device by ID and return correct response structure' : verifica che il GET <code>/device/:id</code> ritorna il dispositivo con la struttura corretta.	Esito atteso: <code>response.body</code> contiene le proprietà <code>id</code> , <code>plantId</code> , <code>name</code> , <code>type</code> , <code>datapoints</code> ; è un array di datapoint; il valore di HTTP status è 200.	S
TI-452	Contesto <i>Device Integration Test</i> ; caso 'should retrieve all devices for a plant' : verifica che il GET <code>/device/plant/:plantId</code> ritorna i dispositivi dell'impianto.	Esito atteso: <code>response.body</code> è un array di dispositivi; tutti i dispositivi appartengono al <code>plantId</code> specificato; il valore di HTTP status è 200.	S

Codice	Descrizione	Risultato atteso	Stato del test
TI-453	Contesto <i>Device Integration Test</i> ; caso 'should accept write datapoint request and return 202' : verifica che il POST /device accetta la richiesta di scrittura e ritorna 202.	Esito atteso: il valore di HTTP status è 202; la richiesta viene accodata per l'elaborazione asincrona.	S
TI-454	Contesto <i>Device Integration Test</i> ; caso 'should reject request without datapointId' : verifica che il POST /device rifiuta le richieste senza datapointId.	Esito atteso: il valore di HTTP status è 400; la risposta contiene un errore di validazione.	S
TI-455	Contesto <i>Device Integration Test</i> ; caso 'should accept webhook payload with datapoint updates' : verifica che il POST /device/update processa i dati di aggiornamento dal webhook.	Esito atteso: il valore di HTTP status è 202; i datapoint vengono aggiornati; gli allarmi vengono controllati.	S
TI-456	Contesto <i>Device Integration Test</i> ; caso 'should filter non-datapoint items from webhook payload' : verifica che il webhook filtra gli elementi non datapoint.	Esito atteso: solo i datapoint validi vengono elaborati; gli elementi non validi vengono ignorati.	S
TI-457	Contesto <i>Device Integration Test</i> ; caso 'should retrieve current values for a device' : verifica che il GET /device/:deviceId/value ritorna i valori attuali del dispositivo.	Esito atteso: response.body è un array di valori datapoint; ogni valore contiene le proprietà necessarie; il valore di HTTP status è 200.	S
TI-458	Contesto <i>Device Integration Test</i> ; caso 'should handle full device lifecycle: find -> write -> read values' : verifica che il ciclo completo di gestione del dispositivo funziona correttamente.	Esito atteso: il dispositivo viene trovato; il valore viene scritto; i valori sono letti correttamente; il valore di HTTP status è appropriato.	S
TI-459	Contesto <i>Device Integration Test</i> ; caso 'should handle multiple devices in same plant' : verifica che il sistema gestisce correttamente più dispositivi dello stesso impianto.	Esito atteso: i dispositivi mantengono l'associazione corretta con l'impianto; le operazioni non interferiscono tra loro.	S
TI-460	Contesto <i>Device Integration Test</i> ; caso 'should process webhook and trigger alarms concurrently' : verifica che il webhook processa i dati e attiva allarmi in modo concorrente.	Esito atteso: più dispositivi vengono elaborati contemporaneamente; gli allarmi vengono controllati per ogni datapoint; il flusso completa correttamente.	S

Tabella 15: Tabella dei test di integrazione

4.4 Test di unità

Codice	Descrizione	Esito atteso	Stato del test
TU-1	Contesto appConfig initialization; caso <code>'runs EventSubscriptionService initialization at bootstrap'</code> : verifica che <code>initializeSpy</code> viene invocato con <code>[]</code> .	<code>initializeSpy</code> viene invocato con <code>[]</code> .	S
TU-2	Contesto App Routes; caso <code>'auth route configurata con lazy loading'</code> : verifica che route è definito e <code>route?.loadChildren</code> è definito.	route è definito; <code>route?.loadChildren</code> è definito; il valore di <code>typeof route?.loadChildren</code> è <code>'function'</code> .	S
TU-3	Contesto App Routes; caso <code>'vimar-link route protetta da authGuard e roleGuard'</code> : verifica che <code>route?.canActivate</code> contiene <code>authGuard</code> e <code>route?.canActivate</code> contiene <code>roleGuard</code> .	<code>route?.canActivate</code> contiene <code>authGuard</code> ; <code>route?.canActivate</code> contiene <code>roleGuard</code> .	S
TU-4	Contesto App Routes; caso <code>'vimar-link route richiede ruolo AMMINISTRATORE'</code> : verifica che il valore di <code>route?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	il valore di <code>route?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	S
TU-5	Contesto App Routes; caso <code>'route "" carica il main layout'</code> : verifica che route è definito e <code>route?.canActivate</code> contiene <code>authGuard</code> .	route è definito; <code>route?.canActivate</code> contiene <code>authGuard</code> ; <code>route?.loadChildren</code> è definito.	S
TU-6	Contesto App Routes; caso <code>'route "*" fa redirect alla root'</code> : verifica che <code>wildcardRoute</code> è definito e il valore di <code>wildcardRoute?.redirectTo</code> è <code>''</code> .	<code>wildcardRoute</code> è definito; il valore di <code>wildcardRoute?.redirectTo</code> è <code>''</code> .	S
TU-7	Contesto App; caso <code>'should create the app'</code> : verifica che app è valorizzato.	app è valorizzato.	S
TU-8	Contesto App; caso <code>'should render router outlet'</code> : verifica che <code>compiled.querySelector('router-outlet')</code> è valorizzato.	<code>compiled.querySelector('router-outlet')</code> è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-9	Contesto environment; caso 'espone la configurazione base attesa': verifica che il valore di typeof environment.production è 'boolean' e il valore di environment.apiUrl.endsWith('/api') è true.	il valore di typeof environment.production è 'boolean'; il valore di environment.apiUrl.endsWith('/api') è true.	S
TU-10	Contesto Feature Routes Quick Coverage; caso 'copre alarm-configuration route e lazy component': verifica che il valore di route.path è "" e route.canActivate contiene authGuard.	il valore di route.path è ""; route.canActivate contiene authGuard; route.providers contiene <i>AlarmConfigStateService</i> ; il valore di typeof child?.loadComponent è 'function'.	S
TU-11	Contesto Feature Routes Quick Coverage; caso 'copre alarm-history route e lazy component': verifica che il valore di route.path è "" e route.canActivate contiene authGuard.	il valore di route.path è ""; route.canActivate contiene authGuard; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-12	Contesto Feature Routes Quick Coverage; caso 'copre alarm-management route e lazy component': verifica che il valore di route.path è "" e route.canActivate contiene authGuard.	il valore di route.path è ""; route.canActivate contiene authGuard; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-13	Contesto Feature Routes Quick Coverage; caso 'copre analytics route e lazy component': verifica che il valore di route.path è "" e il valore di typeof route.loadComponent è 'function'.	il valore di route.path è ""; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-14	Contesto Feature Routes Quick Coverage; caso 'copre apartment-monitor route e lazy component': verifica che il valore di route.path è "" e route.canActivate contiene authGuard.	il valore di route.path è ""; route.canActivate contiene authGuard; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-15	Contesto Feature Routes Quick Coverage; caso 'copre dashboard route e lazy component': verifica che il valore di route.path è "" e il valore di typeof route.loadComponent è 'function'.	il valore di route.path è ""; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-16	Contesto Feature Routes Quick Coverage; caso 'copre device-interaction routes e lazy component': verifica che DEVICE_INTERACTION_-ROUTES ha lunghezza 2 e route.canActivate contiene authGuard.	DEVICE_INTERACTION_-ROUTES ha lunghezza 2; route.canActivate contiene authGuard; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-17	Contesto Feature Routes Quick Coverage; caso 'copre my-vimar-integration route, provider binding e lazy component': verifica che il valore di route.path è " e providerList contiene MyVimarCloudApiFeatureService.	il valore di route.path è "; providerList contiene MyVimarCloudApiFeatureService; il valore di existingProvider?.useExisting è MyVimarCloudApiFeatureService; child?.canActivate contiene authGuard.	S
TU-18	Contesto Feature Routes Quick Coverage; caso 'copre notification route e lazy component': verifica che il valore di route.path è " e route.canActivate contiene authGuard.	il valore di route.path è "; route.canActivate contiene authGuard; il valore di typeof route.loadComponent è 'function'; loaded è valorizzato.	S
TU-19	Contesto Feature Routes Quick Coverage; caso 'copre user-management route e lazy component': verifica che il valore di route.path è " e route.canActivate contiene authGuard.	il valore di route.path è "; route.canActivate contiene authGuard; route.canActivate contiene adminGuard; il valore di typeof route.loadComponent è 'function'.	S
TU-20	Contesto Feature Routes Quick Coverage; caso 'copre ward-management route, providers e lazy component': verifica che il valore di route.path è " e route.providers contiene WardStore.	il valore di route.path è "; route.providers contiene WardStore; route.providers contiene WardOperationsService; route.providers contiene AssignmentOperationsService.	S
TU-21	Contesto adminGuard; caso 'consente sempre l accesso': verifica che il valore di result è true.	il valore di result è true.	S
TU-22	Contesto authGuard; caso 'restituisce true quando l'utente è autenticato': verifica che il valore di result è true e authService-Mock.restoreSessionFromRefresh non viene invocato.	il valore di result è true; authService-Mock.restoreSessionFromRefresh non viene invocato; routerMock.createUrlTree non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-23	Contesto authGuard; caso 'consente accesso quando restore sessione riesce' : verifica che il valore di await <code>firstValueFrom(result as ReturnType<typeof of>)</code> è true e <code>routerMock.createUrlTree</code> non viene invocato.	il valore di await <code>firstValueFrom(result as ReturnType<typeof of>)</code> è true; <code>routerMock.createUrlTree</code> non viene invocato.	S
TU-24	Contesto authGuard; caso 'redirect su /auth/login quando restore sessione fallisce' : verifica che await <code>firstValueFrom(result as ReturnType<typeof of>)</code> coincide con <code>{ redirected: true }</code> e <code>routerMock.createUrlTree</code> viene invocato con <code>['/auth/login']</code> .	await <code>firstValueFrom(result as ReturnType<typeof of>)</code> coincide con <code>{ redirected: true }</code> ; <code>routerMock.createUrlTree</code> viene invocato con <code>['/auth/login']</code> .	S
TU-25	Contesto roleGuard; caso 'restituisce true quando la rotta non richiede ruolo' : verifica che il valore di result è true e <code>authServiceMock.hasRole</code> non viene invocato.	il valore di result è true; <code>authServiceMock.hasRole</code> non viene invocato.	S
TU-26	Contesto roleGuard; caso 'restituisce true quando il ruolo richiesto è presente' : verifica che <code>authServiceMock.hasRole</code> viene invocato con <code>UserRole.AMMINISTRATORE</code> e il valore di result è true.	<code>authServiceMock.hasRole</code> viene invocato con <code>UserRole.AMMINISTRATORE</code> ; il valore di result è true.	S
TU-27	Contesto roleGuard; caso 'redirect su apartment-monitor quando il ruolo richiesto manca' : verifica che <code>routerMock.createUrlTree</code> viene invocato con <code>['/apartment-monitor']</code> e result coincide con <code>{ redirected: true }</code> .	<code>routerMock.createUrlTree</code> viene invocato con <code>['/apartment-monitor']</code> ; result coincide con <code>{ redirected: true }</code> .	S
TU-28	Contesto authInterceptor; caso 'inoltra la richiesta invariata quando il token non esiste' : verifica che il valore di <code>capturedRequest</code> è request e il valore di <code>capturedRequest?.headers.has('Authorization')</code> è false.	il valore di <code>capturedRequest</code> è request; il valore di <code>capturedRequest?.headers.has('Authorization')</code> è false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-29	Contesto <code>authInterceptor</code> ; caso 'aggiunge header Authorization Bearer quando il token esiste' : verifica che il valore di <code>capturedRequest</code> non è request e il valore di <code>capturedRequest?.headers.get('Authorization')</code> è <code>'Bearer jwt-123'</code> .	il valore di <code>capturedRequest</code> non è request; il valore di <code>capturedRequest?.headers.get('Authorization')</code> è <code>'Bearer jwt-123'</code> .	S
TU-30	Contesto <code>authInterceptor</code> ; caso 'aggiunge header Authorization anche su /auth/first-login' : verifica che il valore di <code>capturedRequest?.headers.get('Authorization')</code> è <code>'Bearer jwt-123'</code> .	il valore di <code>capturedRequest?.headers.get('Authorization')</code> è <code>'Bearer jwt-123'</code> .	S
TU-31	Contesto <code>authInterceptor</code> ; caso 'su 401 esegue refresh e ritenta con il nuovo access token' : verifica che <code>authServiceMock.refreshAccessToken</code> viene invocato 1 volta e <code>forwardedRequests</code> ha lunghezza 2.	<code>authServiceMock.refreshAccessToken</code> viene invocato 1 volta; <code>forwardedRequests</code> ha lunghezza 2; il valore di <code>forwardedRequests[0].headers.get('Authorization')</code> è <code>'Bearer jwt-old'</code> ; il valore di <code>forwardedRequests[1].headers.get('Authorization')</code> è <code>'Bearer jwt-new'</code> .	S
TU-32	Contesto <code>authInterceptor</code> ; caso 'non tenta refresh quando la 401 arriva da un endpoint auth' : verifica che <code>authServiceMock.refreshAccessToken</code> non viene invocato.	<code>authServiceMock.refreshAccessToken</code> non viene invocato.	S
TU-33	Contesto <code>authInterceptor</code> ; caso 'non tenta refresh quando la 401 arriva da /auth/first-login' : verifica che <code>authServiceMock.refreshAccessToken</code> non viene invocato.	<code>authServiceMock.refreshAccessToken</code> non viene invocato.	S
TU-34	Contesto <code>ApiErrorDisplayService</code> ; caso 'usa fallback per errori non HTTP quando nonHttpStrategy non è message' : verifica che il valore di <code>message</code> è <code>'Messaggio fallback'</code> .	il valore di <code>message</code> è <code>'Messaggio fallback'</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-35	Contesto <i>ApiErrorDisplayService</i> ; caso 'usa il messaggio degli errori non HTTP quando nonHttpStrategy e message': verifica che il valore di message è 'runtime failure'.	il valore di message è 'runtime failure'.	S
TU-36	Contesto <i>ApiErrorDisplayService</i> ; caso 'mappa status 0 con actionLabel': verifica che il valore di message è 'Impossibile contattare il server durante il caricamento degli allarmi.'.	il valore di message è 'Impossibile contattare il server durante il caricamento degli allarmi.'.	S
TU-37	Contesto <i>ApiErrorDisplayService</i> ; caso 'mappa status 401 con actionLabel': verifica che il valore di message è 'Sessione scaduta o non valida. Effettua nuovamente il login per aggiornare la vista.'.	il valore di message è 'Sessione scaduta o non valida. Effettua nuovamente il login per aggiornare la vista.'.	S
TU-38	Contesto <i>ApiErrorDisplayService</i> ; caso 'mappa status 403 con forbiddenMessage custom se presente': verifica che il valore di message è 'Accesso negato dalla policy applicativa.'.	il valore di message è 'Accesso negato dalla policy applicativa.'.	S
TU-39	Contesto <i>ApiErrorDisplayService</i> ; caso 'mappa status 403 usando actionLabel quando forbiddenMessage non è presente': verifica che il valore di message è 'Non hai i permessi necessari per eseguire l'operazione.'.	il valore di message è 'Non hai i permessi necessari per eseguire l'operazione.'.	S
TU-40	Contesto <i>ApiErrorDisplayService</i> ; caso 'mappa status 500 con actionLabel': verifica che il valore di message è 'Errore interno del server durante il completamento della richiesta. Riprova tra poco.'.	il valore di message è 'Errore interno del server durante il completamento della richiesta. Riprova tra poco.'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-41	Contesto <i>ApiErrorDisplayService</i> ; caso 'usa error.error string quando disponibile per status non speciali': verifica che il valore di message è 'Payload non valido'.	il valore di message è 'Payload non valido'.	S
TU-42	Contesto <i>ApiErrorDisplayService</i> ; caso 'usa error.error.message array quando disponibile per status non speciali': verifica che il valore di message è 'campo obbligatorio; formato non valido'.	il valore di message è 'campo obbligatorio; formato non valido'.	S
TU-43	Contesto <i>ApiErrorDisplayService</i> ; caso 'usa fallback quando payload HTTP non contiene messaggi utili': verifica che il valore di message è 'Messaggio fallback'.	il valore di message è 'Messaggio fallback'.	S
TU-44	Contesto <i>BreadcrumbService</i> ; caso 'should be created': verifica che service è valorizzato.	service è valorizzato.	S
TU-45	Contesto <i>BreadcrumbService</i> ; caso 'should expose a breadcrumbs\$ observable': verifica che service.breadcrumbs\$ è definito.	service.breadcrumbs\$ è definito.	S
TU-46	Contesto <i>BreadcrumbService</i> ; caso 'should emit an empty array immediately when there are no children': verifica che crumbs coincide con [].	crumbs coincide con [].	S
TU-47	Contesto <i>BreadcrumbService</i> ; caso 'should emit breadcrumbs immediately on subscription without waiting for navigation': verifica che il valore di crumbs.length è 1 e crumbs[0] coincide con { label: 'Home', url: '/home' }.	il valore di crumbs.length è 1; crumbs[0] coincide con { label: 'Home', url: '/home' }.	S
TU-48	Contesto <i>BreadcrumbService</i> ; caso 'should re-emit breadcrumbs on NavigationEnd': verifica che il valore di emissions.length è 2.	il valore di emissions.length è 2.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-49	Contesto <i>BreadcrumbService</i> ; caso 'should ignore router events that are not <i>NavigationEnd</i>' : verifica che crumbs coincide con [] e crumbsAfter coincide con [].	crumbs coincide con []; crumbsAfter coincide con [].	S
TU-50	Contesto <i>BreadcrumbService</i> ; caso 'should build a single breadcrumb for a one-level route' : verifica che crumbs coincide con [{ label: 'About', url: '/about' }].	crumbs coincide con [{ label: 'About', url: '/about' }].	S
TU-51	Contesto <i>BreadcrumbService</i> ; caso 'should skip routes without a breadcrumb data property' : verifica che crumbs coincide con [].	crumbs coincide con [].	S
TU-52	Contesto <i>BreadcrumbService</i> ; caso 'should skip routes with an empty url segment even if breadcrumb label is set' : verifica che crumbs coincide con [].	crumbs coincide con [].	S
TU-53	Contesto <i>BreadcrumbService</i> ; caso 'should handle multiple sibling top-level routes' : verifica che crumbs coincide con [{ label: 'Products', url: '/products' }, { label: 'Contact', url: '/contact' },].	crumbs coincide con [{ label: 'Products', url: '/products' }, { label: 'Contact', url: '/contact' },].	S
TU-54	Contesto <i>BreadcrumbService</i> ; caso 'should build breadcrumbs for two-level nested routes' : verifica che crumbs coincide con [{ label: 'Products', url: '/products' }, { label: 'Detail', url: '/products/detail' },].	crumbs coincide con [{ label: 'Products', url: '/products' }, { label: 'Detail', url: '/products/detail' },].	S
TU-55	Contesto <i>BreadcrumbService</i> ; caso 'should skip intermediate segments that have no breadcrumb label' : verifica che crumbs coincide con [{ label: 'Detail', url: '/products/detail' }].	crumbs coincide con [{ label: 'Detail', url: '/products/detail' }].	S
TU-56	Contesto <i>BreadcrumbService</i> ; caso 'should concatenate multi-segment url paths correctly' : verifica che crumbs coincide con [{ label: 'Locale', url: '/en/us' }].	crumbs coincide con [{ label: 'Locale', url: '/en/us' }].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-57	Contesto <i>BreadcrumbService</i> ; caso 'should not double-slash when a parent segment is empty' : verifica che il valore di <code>crumbs[0].url</code> è <code>'/settings'</code> e <code>crumbs[0].url</code> non contiene <code>'/'</code> .	il valore di <code>crumbs[0].url</code> è <code>'/settings'</code> ; <code>crumbs[0].url</code> non contiene <code>'/'</code> .	S
TU-58	Contesto <i>InternalAuthService</i> ; caso 'esegue login, salva sessione e rende autenticato l'utente' : verifica che il valore di <code>request.request.method</code> è <code>'POST'</code> e il valore di <code>request.request.withCredentials</code> è <code>true</code> .	il valore di <code>request.request.method</code> è <code>'POST'</code> ; il valore di <code>request.request.withCredentials</code> è <code>true</code> ; <code>request.request.body</code> coincide con <code>{ username: 'mrossi', password: loginCredential }</code> ; il valore di <code>session.username</code> è <code>'mrossi'</code> .	S
TU-59	Contesto <i>InternalAuthService</i> ; caso 'chiama endpoint first-login con payload corretto' : verifica che il valore di <code>request.request.method</code> è <code>'POST'</code> e il valore di <code>request.request.withCredentials</code> è <code>true</code> .	il valore di <code>request.request.method</code> è <code>'POST'</code> ; il valore di <code>request.request.withCredentials</code> è <code>true</code> ; <code>request.request.body</code> coincide con <code>{ username: 'mrossi', tempPassword: temporaryCredential, password: nextCredential, }</code> ; il valore di <code>session.username</code> è <code>'mrossi'</code> .	S
TU-60	Contesto <i>InternalAuthService</i> ; caso 'chiama refresh, aggiorna il token e restituisce il nuovo accessToken' : verifica che il valore di <code>request.request.method</code> è <code>'POST'</code> e il valore di <code>request.request.withCredentials</code> è <code>true</code> .	il valore di <code>request.request.method</code> è <code>'POST'</code> ; il valore di <code>request.request.withCredentials</code> è <code>true</code> ; <code>request.request.body</code> coincide con <code>{}</code> ; il valore di <code>refreshedToken</code> è <code>accessToken</code> .	S
TU-61	Contesto <i>InternalAuthService</i> ; caso 'ripristina sessione da refresh cookie quando token in memoria manca' : verifica che il valore di <code>request.request.method</code> è <code>'POST'</code> e il valore di <code>request.request.withCredentials</code> è <code>true</code> .	il valore di <code>request.request.method</code> è <code>'POST'</code> ; il valore di <code>request.request.withCredentials</code> è <code>true</code> ; il valore di <code>await restorePromise</code> è <code>true</code> ; il valore di <code>service.isAuthenticated()</code> è <code>true</code> .	S
TU-62	Contesto <i>InternalAuthService</i> ; caso 'restituisce false quando restore sessione fallisce' : verifica che il valore di <code>await restorePromise</code> è <code>false</code> e il valore di <code>service.isAuthenticated()</code> è <code>false</code> .	il valore di <code>await restorePromise</code> è <code>false</code> ; il valore di <code>service.isAuthenticated()</code> è <code>false</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-63	Contesto <i>InternalAuthService</i> ; caso 'esegue una sola richiesta refresh con restore concorrenti': verifica che il valore di await restorePromiseOne è true e il valore di await restorePromiseTwo è true.	il valore di await restorePromiseOne è true; il valore di await restorePromiseTwo è true.	S
TU-64	Contesto <i>InternalAuthService</i> ; caso 'resetta token e utente al logout': verifica che service.getToken() è null e service.getRole() è null.	service.getToken() è null; service.getRole() è null; il valore di service.isAuthenticated() è false.	S
TU-65	Contesto <i>InternalAuthService</i> ; caso 'chiama /auth/logout con credenziali e resetta sessione locale': verifica che il valore di request.request.method è 'POST' e il valore di request.request.withCredentials è true.	il valore di request.request.method è 'POST'; il valore di request.request.withCredentials è true; request.request.body coincide con {}; service.getToken() è null.	S
TU-66	Contesto <i>InternalAuthService</i> ; caso 'espone stream utente corrente con valore iniziale null': verifica che currentUser è null.	currentUser è null.	S
TU-67	Contesto <i>AnalyticsComponent</i> ; caso 'viene creato': verifica che component è valorizzato.	component è valorizzato.	S
TU-68	Contesto <i>AnalyticsComponent</i> ; caso 'carica il primo appartamento di default': verifica che mockAnalyticsService.getAllApartments viene invocato e mockAnalyticsService.getAnalytics viene invocato con '1'.	mockAnalyticsService.getAllApartments viene invocato; mockAnalyticsService.getAnalytics viene invocato con '1'.	F
TU-69	Contesto <i>AnalyticsComponent</i> ; caso 'aggiorna grafici al cambio appartamento': verifica che mockAnalyticsService.getAnalytics viene invocato con '2'.	mockAnalyticsService.getAnalytics viene invocato con '2'.	F
TU-70	Contesto <i>AnalyticsComponent</i> ; caso 'should return undefined if metric is not found': verifica che result è undefined.	result è undefined.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-71	Contesto <i>AnalyticsComponent</i> ; caso 'should find the correct metric in getChartByMetric' : verifica che result è definito e il valore di result?.metric è 'plant-anomalies'.	result è definito; il valore di result?.metric è 'plant-anomalies'.	S
TU-72	Contesto <i>MainLayoutComponent</i> ; caso 'carica i NavItem con il ruolo corretto' : verifica che mockNavService.getNavItems viene invocato con UserRole.OPERATORE_-SANITARIO e component.navItems.length è maggiore di 0.	mockNavService.getNavItems viene invocato con UserRole.OPERATORE_-SANITARIO; component.navItems.length è maggiore di 0.	S
TU-73	Contesto <i>MainLayoutComponent</i> ; caso 'reindirizza al login quando non esiste sessione utente' : verifica che component.navItems coincide con [] e navigateSpy viene invocato con ['/auth/login'].	component.navItems coincide con []; navigateSpy viene invocato con ['/auth/login'].	S
TU-74	Contesto <i>MainLayoutComponent</i> ; caso 'mostra avviso topbar per admin senza account MyVimar collegato' : verifica che mockMyVimarService.getLinkedAccount viene invocato 1 volta e il valore di component.showVimarAssociationWarning è true.	mockMyVimarService.getLinkedAccount viene invocato 1 volta; il valore di component.showVimarAssociationWarning è true.	S
TU-75	Contesto <i>MainLayoutComponent</i> ; caso 'non mostra avviso topbar quando account MyVimar risulta collegato' : verifica che il valore di component.showVimarAssociationWarning è false.	il valore di component.showVimarAssociationWarning è false.	S
TU-76	Contesto <i>MainLayoutComponent</i> ; caso 'inverte correttamente isCollapsed con il segnale ricevuto' : verifica che il valore di component.isCollapsed è true e il valore di component.isCollapsed è false.	il valore di component.isCollapsed è true; il valore di component.isCollapsed è false.	S
TU-77	Contesto <i>MainLayoutComponent</i> ; caso 'invoca correttamente logout' : verifica che mockAuthService.logoutFromBackend viene invocato e navigateSpy viene invocato con ['/auth/login'].	mockAuthService.logoutFromBackend viene invocato; navigateSpy viene invocato con ['/auth/login'].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-78	Contesto <i>MainLayoutComponent</i> ; caso 'apre il pannello profilo e carica stato MyVimar': verifica che il valore di <code>component.isProfilePanelOpen</code> è true e <code>mockMyVimarService.getLinkedAccount</code> viene invocato 2 volte.	il valore di <code>component.isProfilePanelOpen</code> è true; <code>mockMyVimarService.getLinkedAccount</code> viene invocato 2 volte; <code>component.vimarAccount</code> coincide con { email: ", isLinked: false }.	S
TU-79	Contesto <i>MainLayoutComponent</i> ; caso 'apre il pannello profilo anche per utenti non admin ma senza caricare stato MyVimar': verifica che il valore di <code>component.isProfilePanelOpen</code> è true e <code>mockMyVimarService.getLinkedAccount</code> non viene invocato.	il valore di <code>component.isProfilePanelOpen</code> è true; <code>mockMyVimarService.getLinkedAccount</code> non viene invocato.	S
TU-80	Contesto <i>MainLayoutComponent</i> ; caso 'naviga a vimar-link dalla sezione profilo': verifica che <code>navigateSpy</code> viene invocato con <code>['/vimar-link']</code> .	<code>navigateSpy</code> viene invocato con <code>['/vimar-link']</code> .	S
TU-81	Contesto <i>MainLayoutComponent</i> ; caso 'richiede refresh allarmi attivi quando si clicca la stessa voce menu della pagina corrente': verifica che <code>mockAlarmManagementRefreshService.requestRefresh</code> viene invocato 1 volta.	<code>mockAlarmManagementRefreshService.requestRefresh</code> viene invocato 1 volta.	S
TU-82	Contesto <i>MainLayoutComponent</i> ; caso 'dovrebbe collegarsi correttamente al conteggio delle notifiche non lette': verifica che il valore di <code>result</code> è <code>mockCount</code> .	il valore di <code>result</code> è <code>mockCount</code> .	S
TU-83	Contesto <i>MainLayoutComponent</i> ; caso 'apre automaticamente il pannello notifiche quando arriva una nuova notifica realtime': verifica che il valore di <code>component.isNotificationPanelOpen</code> è false e il valore di <code>component.isNotificationPanelOpen</code> è true.	il valore di <code>component.isNotificationPanelOpen</code> è false; il valore di <code>component.isNotificationPanelOpen</code> è true.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-84	Contesto Main Layout Routes; caso 'carica <i>MainLayoutComponent</i> in modalità lazy': verifica che <code>rootRoute?.loadComponent</code> è definito e il valore di <code>typeof rootRoute?.loadComponent</code> è 'function'.	<code>rootRoute?.loadComponent</code> è definito; il valore di <code>typeof rootRoute?.loadComponent</code> è 'function'.	S
TU-85	Contesto Main Layout Routes; caso 'dashboard è figlio': verifica che <code>dashboard</code> è definito e <code>dashboard?.canActivate</code> contiene <code>authGuard</code> .	<code>dashboard</code> è definito; <code>dashboard?.canActivate</code> contiene <code>authGuard</code> .	S
TU-86	Contesto Main Layout Routes - Gestione Allarmi (Nested Routes); caso 'alarms route esiste e ha dei figli': verifica che <code>alarmsRoute</code> è definito e <code>alarmsRoute?.children</code> è definito.	<code>alarmsRoute</code> è definito; <code>alarmsRoute?.children</code> è definito.	S
TU-87	Contesto Main Layout Routes - Gestione Allarmi (Nested Routes); caso 'configurazione allarmi riservata ad AMMINISTRATORE': verifica che <code>config?.canActivate</code> contiene <code>roleGuard</code> e il valore di <code>config?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	<code>config?.canActivate</code> contiene <code>roleGuard</code> ; il valore di <code>config?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	S
TU-88	Contesto Main Layout Routes - Gestione Allarmi (Nested Routes); caso 'gestione e storico allarmi richiedono authGuard': verifica che <code>mgmt?.canActivate</code> contiene <code>authGuard</code> e <code>history?.canActivate</code> contiene <code>authGuard</code> .	<code>mgmt?.canActivate</code> contiene <code>authGuard</code> ; <code>history?.canActivate</code> contiene <code>authGuard</code> .	S
TU-89	Contesto Main Layout Routes - Sicurezza e Ruoli; caso 'routes riservate con roleGuard e ruolo AMMINISTRATORE': verifica che <code>route?.canActivate</code> contiene <code>roleGuard</code> e il valore di <code>route?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	<code>route?.canActivate</code> contiene <code>roleGuard</code> ; il valore di <code>route?.data?.['requiredRole']</code> è <code>UserRole.AMMINISTRATORE</code> .	S
TU-90	Contesto Main Layout Routes; caso 'wildcard reindirizza a dashboard': verifica che il valore di <code>wildcard?.redirectTo</code> è 'dashboard'.	il valore di <code>wildcard?.redirectTo</code> è 'dashboard'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-91	Contesto user-auth.routes; caso 'espone i path login e first-access con lazy load dei componenti': verifica che il valore di typeof <code>loginRoute?.loadComponent</code> è 'function' e il valore di typeof <code>firstAccessRoute?.loadComponent</code> è 'function'.	il valore di typeof <code>loginRoute?.loadComponent</code> è 'function'; il valore di typeof <code>firstAccessRoute?.loadComponent</code> è 'function'; <code>loginComponent</code> è definito; <code>firstAccessComponent</code> è definito.	S
TU-92	Contesto user-auth.routes; caso 'contiene redirect di default verso login': verifica che il valore di <code>defaultRedirect?.pathMatch</code> è 'full' e il valore di <code>defaultRedirect?.redirectTo</code> è 'login'.	il valore di <code>defaultRedirect?.pathMatch</code> è 'full'; il valore di <code>defaultRedirect?.redirectTo</code> è 'login'.	S
TU-93	Contesto device-taxonomy; caso 'resolves thermostat when type token is explicit': verifica che il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	S
TU-94	Contesto device-taxonomy; caso 'infers thermostat from endpoint sfeType when raw type is ambiguous': verifica che il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	S
TU-95	Contesto device-taxonomy; caso 'infers thermostat from HVAC command endpoint when raw type is ambiguous': verifica che il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	il valore di resolved è <code>DeviceType.THERMOSTAT</code> .	S
TU-96	Contesto device-taxonomy; caso 'infers entrance door from italian device name when raw type is ambiguous': verifica che il valore di resolved è <code>DeviceType.ENTRANCE_DOOR</code> .	il valore di resolved è <code>DeviceType.ENTRANCE_DOOR</code> .	S
TU-97	Contesto device-taxonomy; caso 'infers scenario devices from name instead of falling back to light': verifica che il valore di resolved è <code>DeviceType.ALARM_BUTTON</code> .	il valore di resolved è <code>DeviceType.ALARM_BUTTON</code> .	S
TU-98	Contesto device-taxonomy; caso 'falls back to unknown for unknown device signatures': verifica che il valore di resolved è <code>DeviceType.UNKNOWN</code> .	il valore di resolved è <code>DeviceType.UNKNOWN</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-99	Contesto device-taxonomy; caso 'returns readable label for known endpoint code' : verifica che il valore di <i>getEndpointLabel('SFE_Cmd_-ChangeOverMode')</i> è 'Comando cambio modalità HVAC'.	il valore di <i>getEndpointLabel('SFE_Cmd_-ChangeOverMode')</i> è 'Comando cambio modalità HVAC'.	S
TU-100	Contesto device-taxonomy; caso 'returns readable label for fall state endpoint code' : verifica che il valore di <i>getEndpointLabel('SFE_State_-Fall')</i> è 'Stato caduta rilevata'.	il valore di <i>getEndpointLabel('SFE_State_-Fall')</i> è 'Stato caduta rilevata'.	S
TU-101	Contesto device-taxonomy; caso 'returns readable labels for newly observed runtime thermostat commands' : verifica che il valore di <i>getEndpointLabel('SFE_Cmd_-HVACMode')</i> è 'Comando modalità HVAC' e il valore di <i>getEndpointLabel('SFE_Cmd_-OffBehaviour')</i> è 'Comando comportamento in assenza'. il valore di <i>getEndpointLabel('SFE_Cmd_-OnBehaviour')</i> è 'Comando comportamento in presenza'.	il valore di <i>getEndpointLabel('SFE_Cmd_-HVACMode')</i> è 'Comando modalità HVAC'; il valore di <i>getEndpointLabel('SFE_Cmd_-OffBehaviour')</i> è 'Comando comportamento in assenza'; il valore di <i>getEndpointLabel('SFE_Cmd_-OnBehaviour')</i> è 'Comando comportamento in presenza'.	S
TU-102	Contesto device-taxonomy; caso 'returns readable label for active scene command' : verifica che il valore di <i>getEndpointLabel('SFE_Cmd_-DownKey_ActiveScene')</i> è 'Comando attivazione scenario'.	il valore di <i>getEndpointLabel('SFE_Cmd_-DownKey_ActiveScene')</i> è 'Comando attivazione scenario'.	S
TU-103	Contesto device-taxonomy; caso 'returns readable label for timed dynamic light command' : verifica che il valore di <i>getEndpointLabel('SFE_Cmd_-TimedDynamicMode')</i> è 'Comando modalità dinamica temporizzata'.	il valore di <i>getEndpointLabel('SFE_Cmd_-TimedDynamicMode')</i> è 'Comando modalità dinamica temporizzata'.	S
TU-104	Contesto device-taxonomy; caso 'returns raw endpoint code for unknown values' : verifica che il valore di <i>getEndpointLabel('SFE_Cmd_-FutureFeature')</i> è 'Comando future feature'.	il valore di <i>getEndpointLabel('SFE_Cmd_-FutureFeature')</i> è 'Comando future feature'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-105	Contesto device-taxonomy; caso 'tracks unknown endpoint codes discovered at runtime': verifica che <i>getDiscoveredUnmappedSfeCodes()</i> contiene 'SFE_Cmd_FutureFeature' e <i>getDiscoveredUnmappedSfeCodes()</i> contiene 'SFE_Cmd_CompletelyNewOne'.	<i>getDiscoveredUnmappedSfeCodes()</i> contiene 'SFE_Cmd_FutureFeature'; <i>getDiscoveredUnmappedSfeCodes()</i> contiene 'SFE_Cmd_CompletelyNewOne'.	S
TU-106	Contesto device-taxonomy; caso 'returns expected device type label': verifica che il valore di <i>getDeviceTypeLabel(DeviceType.PRESENCE_SENSOR)</i> è 'Sensore presenza'.	il valore di <i>getDeviceTypeLabel(DeviceType.PRESENCE_SENSOR)</i> è 'Sensore presenza'.	S
TU-107	Contesto device-taxonomy; caso 'returns expected fallback label for unknown device type': verifica che il valore di <i>getDeviceTypeLabel(DeviceType.UNKNOWN)</i> è 'Sconosciuto'.	il valore di <i>getDeviceTypeLabel(DeviceType.UNKNOWN)</i> è 'Sconosciuto'.	S
TU-108	Contesto <i>ElapsedTimePipe</i> ; caso 'ritorna il valore originale se la data non è valida': verifica che il valore di <i>pipe.transform('not-a-date')</i> è 'not-a-date'.	il valore di <i>pipe.transform('not-a-date')</i> è 'not-a-date'.	S
TU-109	Contesto <i>ElapsedTimePipe</i> ; caso 'mostra tra poco per timestamp futuri': verifica che il valore di <i>pipe.transform('2026-03-24T12:01:00.000Z')</i> è 'tra poco'.	il valore di <i>pipe.transform('2026-03-24T12:01:00.000Z')</i> è 'tra poco'.	S
TU-110	Contesto <i>ElapsedTimePipe</i> ; caso 'mostra secondi compatti per intervalli inferiori a 60 secondi': verifica che il valore di <i>pipe.transform('2026-03-24T11:59:30.000Z')</i> è '30s fa' e il valore di <i>pipe.transform('2026-03-24T11:59:01.000Z')</i> è '59s fa'.	il valore di <i>pipe.transform('2026-03-24T11:59:30.000Z')</i> è '30s fa'; il valore di <i>pipe.transform('2026-03-24T11:59:01.000Z')</i> è '59s fa'.	S
TU-111	Contesto <i>ElapsedTimePipe</i> ; caso 'mostra minuti compatti da 60 secondi a meno di 1 ora': verifica che il valore di <i>pipe.transform('2026-03-24T11:59:00.000Z')</i> è '1m fa' e il valore di <i>pipe.transform('2026-03-24T11:45:00.000Z')</i> è '15m fa'.	il valore di <i>pipe.transform('2026-03-24T11:59:00.000Z')</i> è '1m fa'; il valore di <i>pipe.transform('2026-03-24T11:45:00.000Z')</i> è '15m fa'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-112	Contesto <i>ElapsedTimePipe</i> ; caso 'mostra ore e minuti compatti per intervalli inferiori a 1 giorno': verifica che il valore di <code>pipe.transform('2026-03-24T11:00:00.000Z')</code> è '1h fa' e il valore di <code>pipe.transform('2026-03-24T10:48:00.000Z')</code> è '1h 12m fa'.	il valore di <code>pipe.transform('2026-03-24T11:00:00.000Z')</code> è '1h fa'; il valore di <code>pipe.transform('2026-03-24T10:48:00.000Z')</code> è '1h 12m fa'; il valore di <code>pipe.transform('2026-03-24T09:00:00.000Z')</code> è '3h fa'.	S
TU-113	Contesto <i>ElapsedTimePipe</i> ; caso 'mostra giorni compatti per intervalli pari o superiori a 1 giorno': verifica che il valore di <code>pipe.transform('2026-03-23T12:00:00.000Z')</code> è '1g fa' e il valore di <code>pipe.transform('2026-03-19T12:00:00.000Z')</code> è '5g fa'.	il valore di <code>pipe.transform('2026-03-23T12:00:00.000Z')</code> è '1g fa'; il valore di <code>pipe.transform('2026-03-19T12:00:00.000Z')</code> è '5g fa'.	S
TU-114	Contesto <i>AlarmApiService</i> ; caso 'getAlarmRules chiama GET /alarm-rules e restituisce la lista': verifica che <code>result</code> coincide con <code>[alarm]</code> e <code>result</code> ha lunghezza 1.	<code>result</code> coincide con <code>[alarm]</code> ; <code>result</code> ha lunghezza 1; il valore di <code>request.request.method</code> è 'GET'.	S
TU-115	Contesto <i>AlarmApiService</i> ; caso 'getAlarmRule chiama GET /alarm-rules/:id e restituisce il dettaglio': verifica che <code>result</code> coincide con <code>alarm</code> e il valore di <code>result.id</code> è 'alarm-1'.	<code>result</code> coincide con <code>alarm</code> ; il valore di <code>result.id</code> è 'alarm-1'; il valore di <code>request.request.method</code> è 'GET'.	S
TU-116	Contesto <i>AlarmApiService</i> ; caso 'getAlarmRule codifica id con caratteri riservati': verifica che <code>result</code> coincide con <code>alarm</code> e il valore di <code>request.request.method</code> è 'GET'.	<code>result</code> coincide con <code>alarm</code> ; il valore di <code>request.request.method</code> è 'GET'.	S
TU-117	Contesto <i>AlarmApiService</i> ; caso 'createAlarmRule chiama POST /alarm-rules con payload corretto': verifica che <code>result</code> coincide con <code>alarm</code> e il valore di <code>request.request.method</code> è 'POST'.	<code>result</code> coincide con <code>alarm</code> ; il valore di <code>request.request.method</code> è 'POST'; <code>request.request.body</code> coincide con <code>payload</code> .	S
TU-118	Contesto <i>AlarmApiService</i> ; caso 'updateAlarmRule chiama PUT /alarm-rules/:id con payload corretto': verifica che <code>result</code> coincide con <code>alarm</code> e il valore di <code>request.request.method</code> è 'PUT'.	<code>result</code> coincide con <code>alarm</code> ; il valore di <code>request.request.method</code> è 'PUT'; <code>request.request.body</code> coincide con <code>payload</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-119	Contesto <i>AlarmApiService</i> ; caso <code>'deleteAlarmRule</code> chiama <code>DELETE /alarm-rules/:id'</code> : verifica che result è null e il valore di <code>request.request.method</code> è <code>'DELETE'</code> .	result è null; il valore di <code>request.request.method</code> è <code>'DELETE'</code> .	S
TU-120	Contesto <i>AlarmApiService</i> ; caso <code>'getActiveAlarms</code> chiama <code>GET /alarm-events/unmanaged/:userId/:limit/:offset</code> e restituisce la lista': verifica che result coincide con <code>[activeAlarm]</code> e result ha lunghezza 1.	result coincide con <code>[activeAlarm]</code> ; result ha lunghezza 1; il valore di <code>request.request.method</code> è <code>'GET'</code> .	S
TU-121	Contesto <i>AlarmApiService</i> ; caso <code>'getResolvedAlarms</code> chiama <code>GET /alarm-events/managed/:userId/:limit/:offset</code> e restituisce la lista': verifica che result coincide con <code>[activeAlarm]</code> e il valore di <code>request.request.method</code> è <code>'GET'</code> .	result coincide con <code>[activeAlarm]</code> ; il valore di <code>request.request.method</code> è <code>'GET'</code> .	S
TU-122	Contesto <i>AlarmApiService</i> ; caso <code>'getAlarmEventById</code> chiama <code>GET /alarm-events/:id</code> e restituisce il dettaglio': verifica che result coincide con <code>activeAlarm</code> e il valore di <code>request.request.method</code> è <code>'GET'</code> .	result coincide con <code>activeAlarm</code> ; il valore di <code>request.request.method</code> è <code>'GET'</code> .	S
TU-123	Contesto <i>AlarmApiService</i> ; caso <code>'getActiveAlarms</code> valida <code>userId, limit e offset interi'</code> : verifica che <code>() => service.getActiveAlarms(1.5)</code> genera l'errore atteso e <code>() => service.getActiveAlarms(7, 2.5, 0)</code> genera l'errore atteso.	<code>() => service.getActiveAlarms(1.5)</code> genera l'errore atteso; <code>() => service.getActiveAlarms(7, 2.5, 0)</code> genera l'errore atteso; <code>() => service.getActiveAlarms(7, 2, 1.5)</code> genera l'errore atteso.	S
TU-124	Contesto <i>AlarmApiService</i> ; caso <code>'getResolvedAlarms</code> valida <code>userId, limit e offset interi'</code> : verifica che <code>() => service.getResolvedAlarms(1.5)</code> genera l'errore atteso e <code>() => service.getResolvedAlarms(7, 2.5, 0)</code> genera l'errore atteso.	<code>() => service.getResolvedAlarms(1.5)</code> genera l'errore atteso; <code>() => service.getResolvedAlarms(7, 2.5, 0)</code> genera l'errore atteso; <code>() => service.getResolvedAlarms(7, 2, 1.5)</code> genera l'errore atteso.	S
TU-125	Contesto <i>AlarmApiService</i> ; caso <code>'resolveAlarm</code> chiama <code>PATCH /alarm-events/resolve'</code> : verifica che result è null e il valore di <code>request.request.method</code> è <code>'PATCH'</code> .	result è null; il valore di <code>request.request.method</code> è <code>'PATCH'</code> ; <code>request.request.body</code> coincide con <code>{ alarmId: 'active-1', userId: 7 }</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-126	Contesto <i>AlarmApiService</i> ; caso <i>'resolveAlarm</i> invia <i>alarmId</i> nel body anche con caratteri riservati': verifica che result è null e il valore di <i>request.request.method</i> è 'PATCH'.	result è null; il valore di <i>request.request.method</i> è 'PATCH'; <i>request.request.body</i> coincide con { <i>alarmId</i> : 'active/1', <i>userId</i> : 9 }.	S
TU-127	Contesto <i>AlarmApiService</i> ; caso <i>'getAlarmRules</i> restituisce lista vuota senza errori di parsing': verifica che result coincide con [] e il valore di <i>request.request.method</i> è 'GET'.	result coincide con []; il valore di <i>request.request.method</i> è 'GET'.	S
TU-128	Contesto <i>AlarmApiService</i> ; caso <i>'propaga</i> l'errore HTTP al chiamante': verifica che il valore di <i>receivedStatus</i> è 404.	il valore di <i>receivedStatus</i> è 404.	S
TU-129	Contesto <i>AlarmManagementRefreshService</i> ; caso <i>'espone</i> un <i>observable hot</i> che emette alla richiesta di refresh': verifica che dopo risoluzione, <i>refreshPromise</i> è undefined.	dopo risoluzione, <i>refreshPromise</i> è undefined.	S
TU-130	Contesto <i>AlarmManagementRefreshService</i> ; caso <i>'non</i> bufferizza refresh precedenti per nuovi subscriber': verifica che <i>nextEmission</i> non viene invocato e <i>nextEmission</i> viene invocato 1 volta.	<i>nextEmission</i> non viene invocato; <i>nextEmission</i> viene invocato 1 volta.	S
TU-131	Contesto <i>AlarmStateService</i> ; caso <i>'inizializza</i> lo stato con allarmi e notifiche vuoti': verifica che il valore di <i>alarmsHistory.at(-1)</i> è 0 e il valore di <i>notificationsHistory.at(-1)</i> è 0.	il valore di <i>alarmsHistory.at(-1)</i> è 0; il valore di <i>notificationsHistory.at(-1)</i> è 0.	S
TU-132	Contesto <i>AlarmStateService</i> ; caso <i>'aggiunge</i> un nuovo allarme attivo quando viene chiamato <i>onAlarmTriggered</i> ': verifica che il valore di <i>latestAlarms.length</i> è 1 e il valore di <i>latestAlarms[0].id</i> è <i>alarmEventA.id</i> .	il valore di <i>latestAlarms.length</i> è 1; il valore di <i>latestAlarms[0].id</i> è <i>alarmEventA.id</i> ; il valore di <i>latestAlarms[0].alarmRuleId</i> è <i>alarmEventA.alarmRuleId</i> ; il valore di <i>latestAlarms[0].alarmName</i> è <i>alarmEventA.alarmName</i> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-133	Contesto <i>AlarmStateService</i> ; caso 'aggiorna un allarme esistente senza duplicarlo per id istanza': verifica che il valore di <code>latestAlarms.length</code> è 1 e il valore di <code>latestAlarms[0].id</code> è <code>alarmEventA.id</code> .	il valore di <code>latestAlarms.length</code> è 1; il valore di <code>latestAlarms[0].id</code> è <code>alarmEventA.id</code> ; il valore di <code>latestAlarms[0].alarmName</code> è 'Nome allarme aggiornato'; il valore di <code>latestAlarms[0].priority</code> è <code>AlarmPriority.WHITE</code> .	S
TU-134	Contesto <i>AlarmStateService</i> ; caso 'rimuove solo 1'allarme target quando viene chiamato <i>onAlarmResolved</i> ': verifica che <code>latestActiveAlarmIds</code> coincide con <code>[alarmEventB.id]</code> .	<code>latestActiveAlarmIds</code> coincide con <code>[alarmEventB.id]</code> .	S
TU-135	Contesto <i>AlarmStateService</i> ; caso 'antepone le notifiche in modo che la più recente sia la prima': verifica che <code>latestNotificationIds</code> coincide con <code>[notificationB.notificationId, notificationA.notificationId]</code> .	<code>latestNotificationIds</code> coincide con <code>[notificationB.notificationId, notificationA.notificationId]</code> .	S
TU-136	Contesto <i>AlarmStateService</i> ; caso ' <i>removeNotification</i> elimina una sola notifica e aggiorna il contatore': verifica che <code>latestNotificationIds</code> coincide con <code>[notificationB.notificationId]</code> e il valore di <code>unreadNotificationsCount</code> è 1.	<code>latestNotificationIds</code> coincide con <code>[notificationB.notificationId]</code> ; il valore di <code>unreadNotificationsCount</code> è 1.	S
TU-137	Contesto <i>AlarmStateService</i> ; caso ' <i>clearNotifications</i> svuota la lista notifiche e porta il contatore a zero': verifica che <code>latestNotificationIds</code> coincide con <code>[]</code> e il valore di <code>unreadNotificationsCount</code> è 0.	<code>latestNotificationIds</code> coincide con <code>[]</code> ; il valore di <code>unreadNotificationsCount</code> è 0.	S
TU-138	Contesto <i>AlarmStateService</i> ; caso 'una notifica dismissata non viene reintrodotta quando arriva di nuovo': verifica che <code>latestNotificationIds</code> coincide con <code>[]</code> .	<code>latestNotificationIds</code> coincide con <code>[]</code> .	S
TU-139	Contesto <i>AlarmStateService</i> ; caso ' <i>setActiveAlarms</i> imposta snapshot iniziale degli allarmi attivi': verifica che <code>latestActiveAlarmIds</code> coincide con <code>['snapshot-1']</code> .	<code>latestActiveAlarmIds</code> coincide con <code>['snapshot-1']</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-140	Contesto <i>AlarmStateService</i> ; caso <i>'setActiveAlarms preserva eventuali allarmi locali non inclusi nello snapshot'</i> : verifica che <code>latestActiveAlarmIds</code> coincide con <code>['snapshot-1', alarmEventA.id]</code> .	<code>latestActiveAlarmIds</code> coincide con <code>['snapshot-1', alarmEventA.id]</code> .	S
TU-141	Contesto <i>AlarmStateService</i> ; caso <i>'setActiveAlarms non reintroduce allarmi già risolti localmente'</i> : verifica che <code>latestActiveAlarmIds</code> coincide con <code>[]</code> .	<code>latestActiveAlarmIds</code> coincide con <code>[]</code> .	S
TU-142	Contesto <i>AlarmStateService</i> ; caso <i>'setActiveAlarms in replace sostituisce la pagina corrente senza mantenere elementi precedenti'</i> : verifica che <code>latestActiveAlarmIds</code> coincide con <code>['page-2-a']</code> .	<code>latestActiveAlarmIds</code> coincide con <code>['page-2-a']</code> .	S
TU-143	Contesto <i>AlarmStateService</i> ; caso <i>'emette conteggi derivati corretti per allarmi e notifiche non lette'</i> : verifica che il valore di <code>activeAlarmsCount</code> è 2 e il valore di <code>unreadNotificationsCount</code> è 1.	il valore di <code>activeAlarmsCount</code> è 2; il valore di <code>unreadNotificationsCount</code> è 1; il valore di <code>activeAlarmsCount</code> è 1; il valore di <code>unreadNotificationsCount</code> è 2.	S
TU-144	Contesto <i>EventSubscriptionService</i> ; caso <i>'crea la connessione socket in initialize usando API_BASE_URL normalizzato'</i> : verifica che <code>socketIoFactoryMock</code> viene invocato con <code>'http://api.example.local/ws'</code> .	<code>socketIoFactoryMock</code> viene invocato con <code>'http://api.example.local/ws'</code> .	S
TU-145	Contesto <i>EventSubscriptionService</i> ; caso <i>'risolve il namespace websocket ws quando API_BASE_URL termina con api'</i> : verifica che <code>socketIoFactoryMock</code> viene invocato con <code>'http://localhost/ws'</code> .	<code>socketIoFactoryMock</code> viene invocato con <code>'http://localhost/ws'</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-146	Contesto <i>EventSubscriptionService</i> ; caso 'aggiorna lo stato connessione su connect, reconnect attempt e disconnect': verifica che statuses coincide con [ConnectionStatus.DISCONNECTED, ConnectionStatus.CONNECTED, ConnectionStatus.RECONNECTING, Co...	statuses coincide con [ConnectionStatus.DISCONNECTED, ConnectionStatus.CONNECTED, ConnectionStatus.RECONNECTING, Co...	S
TU-147	Contesto <i>EventSubscriptionService</i> ; caso 'esegue join e leave delle room con deduplicazione': verifica che fakeSocket.emit viene invocato con 'join-ward' e fakeSocket.emit viene invocato con 'leave-ward'.	fakeSocket.emit viene invocato con 'join-ward'; fakeSocket.emit viene invocato con 'leave-ward'; il valore di joinWardCalls.length è 2; il valore di leaveWardCalls.length è 1.	S
TU-148	Contesto <i>EventSubscriptionService</i> ; caso 'riesegue il join delle room già sottoscritte al connect': verifica che il valore di joinCallsBeforeReconnect è 1 e il valore di joinCallsAfterReconnect.length è 2.	il valore di joinCallsBeforeReconnect è 1; il valore di joinCallsAfterReconnect.length è 2; joinCallsAfterReconnect.at(-1) coincide con ['join-ward', 'ward-a'].	S
TU-149	Contesto <i>EventSubscriptionService</i> ; caso 'bootstrap delle room da plant all al login quando cache ward è vuota': verifica che httpClientSpy.get viene invocato con 'http://api.example.local/plant/all' e joinWardCalls coincide con [['join-ward', '21'], ['join-ward', '22'],].	httpClientSpy.get viene invocato con 'http://api.example.local/plant/all'; joinWardCalls coincide con [['join-ward', '21'], ['join-ward', '22'],].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-150	Contesto <i>EventSubscriptionService</i> ; caso 'non interrompe il realtime quando bootstrap plant all fallisce': verifica che <code>httpClientSpy.get</code> viene invocato 1 volta e <code>alarmStateSpy.onAlarmTriggered</code> viene invocato con <code>expect.objectContaining({ id: 'active-alarm-bootstrap-fallback', })</code> .	<code>httpClientSpy.get</code> viene invocato 1 volta; <code>alarmStateSpy.onAlarmTriggered</code> viene invocato con <code>expect.objectContaining({ id: 'active-alarm-bootstrap-fallback', })</code> ; <code>alarmStateSpy.onNotificationReceived</code> viene invocato con <code>expect.objectContaining({ notificationId: 'alarm-triggered-active-alarm-bootstrap-fallback', })</code> .	S
TU-151	Contesto <i>EventSubscriptionService</i> ; caso 'evita bootstrap ripetuto per lo stesso utente nella stessa sessione': verifica che <code>httpClientSpy.get</code> viene invocato 1 volta.	<code>httpClientSpy.get</code> viene invocato 1 volta.	S
TU-152	Contesto <i>EventSubscriptionService</i> ; caso 'refreshWardRoomSubscription riallinea le room senza reload pagina': verifica che <code>httpClientSpy.get</code> viene invocato 2 volte e <code>fakeSocket.emit</code> viene invocato con 'join-ward'.	<code>httpClientSpy.get</code> viene invocato 2 volte; <code>fakeSocket.emit</code> viene invocato con 'join-ward'; <code>fakeSocket.emit</code> viene invocato con 'leave-ward'.	S
TU-153	Contesto <i>EventSubscriptionService</i> ; caso 'ignora eventi raw malformati e payload backend non validi': verifica che <code>alarmStateSpy.onAlarmTriggered</code> non viene invocato e <code>alarmStateSpy.onAlarmResolved</code> non viene invocato.	<code>alarmStateSpy.onAlarmTriggered</code> non viene invocato; <code>alarmStateSpy.onAlarmResolved</code> non viene invocato; <code>alarmStateSpy.onNotificationReceived</code> non viene invocato.	S
TU-154	Contesto <i>EventSubscriptionService</i> ; caso 'disconnette il socket e rimuove i listener alla distruzione': verifica che <code>fakeSocket.removeAllListeners</code> viene invocato e <code>fakeSocket.disconnect</code> viene invocato.	<code>fakeSocket.removeAllListeners</code> viene invocato; <code>fakeSocket.disconnect</code> viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-155	Contesto <i>RealtimeAlarmEventNormalizerService</i> ; caso 'parseBackendTriggeredPayload accetta il contratto backend nativo' : verifica che parsed coincide con { alarmEventId: 'alarm-2', wardId: 3, alarmRuleId: 'rule-2', }.	parsed coincide con { alarmEventId: 'alarm-2', wardId: 3, alarmRuleId: 'rule-2', }.	S
TU-156	Contesto <i>RealtimeAlarmEventNormalizerService</i> ; caso 'parseBackendTriggeredPayload accetta wardId numerico espresso come stringa' : verifica che parsed coincide con { alarmEventId: 'alarm-3', wardId: 12, alarmRuleId: 'rule-3', }.	parsed coincide con { alarmEventId: 'alarm-3', wardId: 12, alarmRuleId: 'rule-3', }.	S
TU-157	Contesto <i>RealtimeAlarmEventNormalizerService</i> ; caso 'parseBackendResolvedPayload estrae alarmEventId e wardId opzionale' : verifica che withWardId coincide con { alarmEventId: 'alarm-4', wardId: 8 } e withoutWardId coincide con { alarmEventId: 'alarm-5', wardId: undefined }.	withWardId coincide con { alarmEventId: 'alarm-4', wardId: 8 }; withoutWardId coincide con { alarmEventId: 'alarm-5', wardId: undefined }.	S
TU-158	Contesto <i>WardRealtimeCacheService</i> ; caso 'mergeWardIds deduplica e normalizza valori stringa/numero' : verifica che service.getWardIds('user-1') coincide con ['1', '2', '3'].	service.getWardIds('user-1') coincide con ['1', '2', '3'].	S
TU-159	Contesto <i>WardRealtimeCacheService</i> ; caso 'mantiene cache separata per utente' : verifica che service.getWardIds('user-a') coincide con ['10'] e service.getWardIds('user-b') coincide con ['11', '12'].	service.getWardIds('user-a') coincide con ['10']; service.getWardIds('user-b') coincide con ['11', '12'].	S
TU-160	Contesto <i>WardRealtimeCacheService</i> ; caso 'clearWardIds rimuove solo 1'utente target' : verifica che service.getWardIds('user-a') coincide con [] e service.getWardIds('user-b') coincide con ['11'].	service.getWardIds('user-a') coincide con []; service.getWardIds('user-b') coincide con ['11'].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-161	Contesto <i>WardRealtimeCacheService</i> ; caso 'recupera cache vuota in caso di JSON non valido': verifica che <code>service.getWardIds('user-z')</code> coincide con [].	<code>service.getWardIds('user-z')</code> coincide con [].	S
TU-162	Contesto <i>WardSocketRoomCoordinatorService</i> ; caso 'activateUser carica wardIds da cache e propone join': verifica che <code>actions.roomsToLeave</code> coincide con [] e <code>actions.roomsToJoin</code> coincide con ['10', '11'].	<code>actions.roomsToLeave</code> coincide con []; <code>actions.roomsToJoin</code> coincide con ['10', '11']; <code>service.getJoinedRooms()</code> coincide con ['10', '11'].	S
TU-163	Contesto <i>WardSocketRoomCoordinatorService</i> ; caso 'requestJoinRoom deduplica e persiste su cache': verifica che il valore di <code>service.requestJoinRoom('20')</code> è '20' e <code>service.requestJoinRoom('20')</code> è null.	il valore di <code>service.requestJoinRoom('20')</code> è '20'; <code>service.requestJoinRoom('20')</code> è null; <code>cacheSpy.setWardIds</code> viene invocato.	S
TU-164	Contesto <i>WardSocketRoomCoordinatorService</i> ; caso 'deactivateUser rimuove room runtime e pulisce cache utente corrente': verifica che <code>roomsToLeave</code> coincide con ['30'] e <code>service.getJoinedRooms()</code> coincide con [].	<code>roomsToLeave</code> coincide con ['30']; <code>service.getJoinedRooms()</code> coincide con []; <code>cacheSpy.clearWardIds</code> viene invocato con 'user-1'.	S
TU-165	Contesto <i>WardSocketRoomCoordinatorService</i> ; caso 'switch utente restituisce room da lasciare e room cache del nuovo utente': verifica che <code>actions.roomsToLeave</code> coincide con ['10', '30'] e <code>actions.roomsToJoin</code> coincide con ['40', '50'].	<code>actions.roomsToLeave</code> coincide con ['10', '30']; <code>actions.roomsToJoin</code> coincide con ['40', '50']; <code>service.getJoinedRooms()</code> coincide con ['40', '50'].	S
TU-166	Contesto <i>NotificationApiService</i> ; caso 'getNotificationsHistory chiama managed/unmanaged e mappa verso NotificationEvent': verifica che <code>result</code> coincide con <code>expected</code> e il valore di <code>managedRequest.request.method</code> è 'GET'.	<code>result</code> coincide con <code>expected</code> ; il valore di <code>managedRequest.request.method</code> è 'GET'; il valore di <code>unmanagedRequest.request.method</code> è 'GET'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-167	Contesto <i>NotificationApiService</i> ; caso <i>'getNotificationsHistory'</i> emette una volta e completa': verifica che il valore di emissions è 1 e il valore di completed è true.	il valore di emissions è 1; il valore di completed è true.	S
TU-168	Contesto <i>NotificationApiService</i> ; caso <i>'getNotificationsHistory'</i> : due sottoscrizioni producono due coppie di request HTTP': verifica che requests ha lunghezza 4 e il valore di request.request.method è 'GET'.	requests ha lunghezza 4; il valore di request.request.method è 'GET'.	S
TU-169	Contesto <i>NotificationApiService</i> ; caso <i>'se non c'è sessione attiva'</i> restituisce [] senza chiamate HTTP': verifica che result coincide con [] e requests ha lunghezza 0.	result coincide con []; requests ha lunghezza 0.	S
TU-170	Contesto <i>AlarmRuleFormMapper</i> ; caso <i>'mappa operatore >= come GREATER_THAN_OR_EQUAL'</i> : verifica che il valore di mapper.toFormValue(rule).thresholdOperator è ThresholdOperator.GREATER_THAN_OR_EQUAL.	il valore di mapper.toFormValue(rule).thresholdOperator è ThresholdOperator.GREATER_THAN_OR_EQUAL.	S
TU-171	Contesto <i>AlarmRuleFormMapper</i> ; caso <i>'mappa operatore <= come LESS_THAN_OR_EQUAL'</i> : verifica che il valore di mapper.toFormValue(rule).thresholdOperator è ThresholdOperator.LESS_THAN_OR_EQUAL.	il valore di mapper.toFormValue(rule).thresholdOperator è ThresholdOperator.LESS_THAN_OR_EQUAL.	S
TU-172	Contesto <i>AlarmRuleFormMapper</i> ; caso <i>'fallisce su operatori sconosciuti'</i> : verifica che () => mapper.toFormValue(rule) genera l'errore atteso.	() => mapper.toFormValue(rule) genera l'errore atteso.	S
TU-173	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toUpdateRequest serializza operatore >= senza degradarlo'</i> : verifica che il valore di result.thresholdOperator è '>='.	il valore di result.thresholdOperator è '>='.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-174	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toCreateRequest</i> serializza operatore <i><=</i> senza degradarlo': verifica che il valore di <i>result.thresholdOperator</i> è <i>'<='</i> .	il valore di <i>result.thresholdOperator</i> è <i>'<='</i> .	S
TU-175	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toCreateRequest</i> normalizza soglia booleana in <i>on/off</i> ': verifica che l'asserzione <i>mapper.toCreateRequest(onFormValue) toMatchObject { plantId: 'plant-1', datapointId: 'dp-4', thresholdOperator: '=', thresholdValue: 'on', }</i> risulta soddisfatta e l'asserzione <i>mapper.toCreateRequest(offFormValue) toMatchObject { plantId: 'plant-1', datapointId: 'dp-4', thresholdOperator: '=', thresholdValue: 'off', }</i> risulta soddisfatta.	l'asserzione <i>mapper.toCreateRequest(onFormValue) toMatchObject { plantId: 'plant-1', datapointId: 'dp-4', thresholdOperator: '=', thresholdValue: 'on', }</i> risulta soddisfatta; l'asserzione <i>mapper.toCreateRequest(offFormValue) toMatchObject { plantId: 'plant-1', datapointId: 'dp-4', thresholdOperator: '=', thresholdValue: 'off', }</i> risulta soddisfatta.	S
TU-176	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toCreateRequest</i> fallisce se <i>thresholdValue</i> è vuoto dopo <i>trim</i> ': verifica che <i>() => mapper.toCreateRequest(invalidFormValue)</i> genera l'errore atteso.	<i>() => mapper.toCreateRequest(invalidFormValue)</i> genera l'errore atteso.	S
TU-177	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toCreateRequest</i> fallisce se <i>thresholdOperator</i> è <i>null</i> ': verifica che <i>() => mapper.toCreateRequest(invalidFormValue)</i> genera l'errore atteso.	<i>() => mapper.toCreateRequest(invalidFormValue)</i> genera l'errore atteso.	S
TU-178	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toToggleRequest</i> fallisce se <i>name</i> è vuoto': verifica che <i>() => mapper.toToggleRequest(invalidRule, true)</i> genera l'errore atteso.	<i>() => mapper.toToggleRequest(invalidRule, true)</i> genera l'errore atteso.	S
TU-179	Contesto <i>AlarmRuleRequestMapper</i> ; caso <i>'toToggleRequest</i> normalizza operatori con padding dal backend': verifica che il valore di <i>result.thresholdOperator</i> è <i>'>'</i> .	il valore di <i>result.thresholdOperator</i> è <i>'>'</i> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-180	Contesto <i>AlarmConfigFormFieldOrchestratorHelper</i> ; caso 'applyModeState in edit mode disabilita campi non modificabili': verifica che il valore di formLike.controls.name.disabled è true e il valore di formLike.controls.plantId.disabled è true.	il valore di formLike.controls.name.disabled è true; il valore di formLike.controls.plantId.disabled è true; il valore di formLike.controls.deviceId.disabled è true; il valore di formLike.controls.datapointId.disabled è true.	S
TU-181	Contesto <i>AlarmConfigFormFieldOrchestratorHelper</i> ; caso 'resolveDeviceSelection ritorna solo datapoint leggibili e auto-selezione se singolo': verifica che il valore di selection.normalizedDeviceId è 'device-1' e selection.readableDatapoints coincide con [readableDatapoint].	il valore di selection.normalizedDeviceId è 'device-1'; selection.readableDatapoints coincide con [readableDatapoint]; il valore di selection.autoSelectedDatapointId è 'dp-readable'.	S
TU-182	Contesto <i>AlarmConfigFormFieldOrchestratorHelper</i> ; caso 'resolveDatapointSelection normalizza id e restituisce il datapoint atteso': verifica che il valore di selection.normalizedDatapointId è 'dp-1' e selection.selectedDatapoint coincide con datapoint.	il valore di selection.normalizedDatapointId è 'dp-1'; selection.selectedDatapoint coincide con datapoint.	S
TU-183	Contesto <i>AlarmConfigFormValidationHelper</i> ; caso 'in create mode con enum forza operatore uguale e valida threshold su enum': verifica che operatorControl.value è null e il valore di valueControl.errors?.['invalidEnumThreshold'] è true.	operatorControl.value è null; il valore di valueControl.errors?.['invalidEnumThreshold'] è true.	S
TU-184	Contesto <i>AlarmConfigFormValidationHelper</i> ; caso 'in create mode con datapoint numerico applica validazione numerica': verifica che il valore di operatorControl.value è ThresholdOperator.GREATER_THAN e valueControl.errors?.['pattern'] è valorizzato.	il valore di operatorControl.value è ThresholdOperator.GREATER_THAN; valueControl.errors?.['pattern'] è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-185	Contesto <i>AlarmConfigFormValidationHelper</i> ; caso 'in edit mode con datapoint enum applica lo stesso vincolo operatore della create': verifica che operatorControl.value è null e il valore di valueControl.errors?.['invalidEnumThreshold'] è true.	operatorControl.value è null; il valore di valueControl.errors?.['invalidEnumThreshold'] è true.	S
TU-186	Contesto <i>AlarmConfigFormValidationHelper</i> ; caso 'in edit mode con datapoint numerico applica validazione numerica': verifica che valueControl.errors?.['pattern'] è valorizzato.	valueControl.errors?.['pattern'] è valorizzato.	S
TU-187	Contesto <i>AlarmConfigFormValidationHelper</i> ; caso 'in edit mode senza metadati datapoint mantiene come ammesso solo l'operatore corrente': verifica che il valore di operatorControl.errors?.['unsupportedOperator'] è true.	il valore di operatorControl.errors?.['unsupportedOperator'] è true.	S
TU-188	Contesto <i>AlarmConfigFormStateService</i> ; caso 'ensurePlantsLoaded in create mode carica e deduplica gli impianti': verifica che service.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3'] e service.plantsLoadError() è null.	service.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3']; service.plantsLoadError() è null.	S
TU-189	Contesto <i>AlarmConfigFormStateService</i> ; caso 'ensurePlantsLoaded in edit mode non esegue chiamate': verifica che wardApiStub.getAvailablePlants non viene invocato e wardApiStub.getWards non viene invocato.	wardApiStub.getAvailablePlants non viene invocato; wardApiStub.getWards non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-190	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>loadDeviceOptionsByPlant</i> ritorna opzioni dispositivi e pulisce loading state': verifica che <i>apartmentApiStub.loadApartmentViewForPlantId</i> viene invocato con 'plant-1' e result coincide con options.	<i>apartmentApiStub.loadApartmentViewForPlantId</i> viene invocato con 'plant-1'; result coincide con options; il valore di <i>service.isDevicesLoading()</i> è false; <i>service.devicesLoadError()</i> è null.	
TU-191	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>loadDeviceOptionsByPlant</i> in errore restituisce lista vuota e imposta messaggio': verifica che result coincide con [] e il valore di <i>service.isDevicesLoading()</i> è false.	result coincide con []; il valore di <i>service.isDevicesLoading()</i> è false; il valore di <i>service.devicesLoadError()</i> è 'Errore durante il caricamento dei dispositivi.'	S
TU-192	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>resolveDatapointForEdit</i> usa il catalogo impianti e restituisce il datapoint trovato': verifica che <i>apartmentApiStub.getAllPlants</i> viene invocato 1 volta e <i>datapointExtractionStub.findDatapointByDeviceAndDatapointId</i> viene invocato con [{ id: 'plant-1', name: 'Apt', rooms: [] }].	<i>apartmentApiStub.getAllPlants</i> viene invocato 1 volta; <i>datapointExtractionStub.findDatapointByDeviceAndDatapointId</i> viene invocato con [{ id: 'plant-1', name: 'Apt', rooms: [] }]; result coincide con <i>expectedDatapoint</i> .	S
TU-193	Contesto <i>AlarmConfigFormStateService</i> ; caso ' <i>resolveDatapointForEdit</i> ritorna null se il catalogo impianti fallisce': verifica che result è null.	result è null.	S
TU-194	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>inizializza stato con alarms vuoti e nessun errore</i> ': verifica che <i>await firstValueFrom(service.alarms\$)</i> coincide con [] e <i>await firstValueFrom(service.error\$)</i> è null.	<i>await firstValueFrom(service.alarms\$)</i> coincide con []; <i>await firstValueFrom(service.error\$)</i> è null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-195	Contesto <i>AlarmConfigStateService</i> ; caso 'loadAlarmRules aggiorna stato locale quando API risponde con successo': verifica che <i>apiStub.getAlarmRules</i> viene invocato 1 volta e alarms coincide con [alarmA, alarmB].	<i>apiStub.getAlarmRules</i> viene invocato 1 volta; alarms coincide con [alarmA, alarmB]; alarms ha lunghezza 2; await <i>firstValueFrom(service.error\$)</i> è null.	S
TU-196	Contesto <i>AlarmConfigStateService</i> ; caso 'loadAlarmRules espone messaggio errore e pulisce error precedente al retry': verifica che il valore di await <i>firstValueFrom(service.error\$)</i> è 'Errore durante il caricamento degli allarmi.' e await <i>firstValueFrom(service.error\$)</i> è null.	il valore di await <i>firstValueFrom(service.error\$)</i> è 'Errore durante il caricamento degli allarmi.'; await <i>firstValueFrom(service.error\$)</i> è null; await <i>firstValueFrom(service.alarms\$)</i> coincide con [alarmA].	S
TU-197	Contesto <i>AlarmConfigStateService</i> ; caso 'getAlarmRuleById inoltra il valore di API in caso di successo': verifica che <i>apiStub.getAlarmRule</i> viene invocato con 'alarm-1' e result coincide con alarmA.	<i>apiStub.getAlarmRule</i> viene invocato con 'alarm-1'; result coincide con alarmA.	S
TU-198	Contesto <i>AlarmConfigStateService</i> ; caso 'getAlarmRuleById in errore imposta messaggio e completa senza emissioni': verifica che il valore di emitted è false e il valore di await <i>firstValueFrom(service.error\$)</i> è 'Errore durante il recupero dell\{\}'allarme.?	il valore di emitted è false; il valore di await <i>firstValueFrom(service.error\$)</i> è 'Errore durante il recupero dell\{\}'allarme.?	S
TU-199	Contesto <i>AlarmConfigStateService</i> ; caso 'createAlarmRule con form invalido non chiama API e imposta errore di validazione': verifica che il valore di emitted è false e <i>apiStub.createAlarmRule</i> non viene invocato.	il valore di emitted è false; <i>apiStub.createAlarmRule</i> non viene invocato; il valore di await <i>firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme.?	S

Codice	Descrizione	Esito atteso	Stato del test
TU-200	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>createAlarmRule</i> in errore API imposta messaggio e non altera lo stato locale': verifica che il valore di <i>emitted</i> è false e il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante la creazione dell\{\}'allarme.'	il valore di <i>emitted</i> è false; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Errore durante la creazione dell\{\}'allarme.'; <i>await firstValueFrom(service.alarms\$)</i> coincide con [alarmA]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TU-201	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>createAlarmRule</i> pulisce l'errore precedente quando una creazione successiva va a buon fine': verifica che il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme.' e <i>await firstValueFrom(service.error\$)</i> è null.	il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per la creazione dell\{\}'allarme.'; <i>await firstValueFrom(service.error\$)</i> è null; <i>await firstValueFrom(service.alarms\$)</i> coincide con [alarmB].	S
TU-202	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>updateAlarmRule</i> mantiene il nome originale nel payload update e sostituisce l'allarme nello stato': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1' e <i>await firstValueFrom(service.alarms\$)</i> coincide con [updatedAlarm, alarmB].	<i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1'; <i>await firstValueFrom(service.alarms\$)</i> coincide con [updatedAlarm, alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TU-203	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>updateAlarmRule</i> con form invalido non chiama API e imposta errore': verifica che il valore di <i>emitted</i> è false e <i>apiStub.updateAlarmRule</i> non viene invocato.	il valore di <i>emitted</i> è false; <i>apiStub.updateAlarmRule</i> non viene invocato; il valore di <i>await firstValueFrom(service.error\$)</i> è 'Dati del form non validi per l\{\}'aggiornamento dell\{\}'allarme.'	S

Codice	Descrizione	Esito atteso	Stato del test
TU-204	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l'aggiornamento dell'allarme.'.	il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l'aggiornamento dell'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA, alarmB]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TU-205	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> senza allarme locale usa il nome del form nel payload': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'missing-id'.	<i>apiStub.updateAlarmRule</i> viene invocato con 'missing-id'.	S
TU-206	Contesto <i>AlarmConfigStateService</i> ; caso <i>'updateAlarmRule</i> sostituisce la riga locale anche se il backend ritorna un nuovo id': verifica che alarms coincide con [updatedWithNewId, alarmB] e il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false.	alarms coincide con [updatedWithNewId, alarmB]; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1-v2')</code> è true; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TU-207	Contesto <i>AlarmConfigStateService</i> ; caso <i>'toggleEnabled</i> aggiorna enabled costruendo payload dai dati locali': verifica che <i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1' e await <code>firstValueFrom(service.alarms\$)</code> coincide con [toggledAlarm].	<i>apiStub.updateAlarmRule</i> viene invocato con 'alarm-1'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [toggledAlarm]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TU-208	Contesto <i>AlarmConfigStateService</i> ; caso <i>'toggleEnabled</i> su allarme assente non chiama API e imposta errore': verifica che il valore di emitted è false e <i>apiStub.updateAlarmRule</i> non viene invocato.	il valore di emitted è false; <i>apiStub.updateAlarmRule</i> non viene invocato; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Allarme non trovato nello stato locale.'; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-209	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>toggleEnabled</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante la modifica dello stato dell\{\}'allarme.'.	il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante la modifica dello stato dell\{\}'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TU-210	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>toggleEnabled</i> aggiorna subito lo stato locale anche se il backend ritorna un nuovo id': verifica che alarms coincide con [toggledWithNewId, alarmB] e il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false.	alarms coincide con [toggledWithNewId, alarmB]; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1')</code> è false; il valore di <code>alarms.some((alarm) => alarm.id === 'alarm-1-v2')</code> è true.	S
TU-211	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> rimuove l'allarme locale quando API conferma': verifica che il valore di emitted è true e <i>apiStub.deleteAlarmRule</i> viene invocato con 'alarm-1'.	il valore di emitted è true; <i>apiStub.deleteAlarmRule</i> viene invocato con 'alarm-1'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S
TU-212	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> in errore API imposta messaggio e mantiene lo stato invariato': verifica che il valore di emitted è false e il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'.	il valore di emitted è false; il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmA, alarmB]; <i>refreshServiceStub.requestRefresh</i> non viene invocato.	S
TU-213	Contesto <i>AlarmConfigStateService</i> ; caso ' <i>deleteAlarmRule</i> pulisce errore precedente quando un delete successivo riesce': verifica che il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.' e await <code>firstValueFrom(service.error\$)</code> è null.	il valore di await <code>firstValueFrom(service.error\$)</code> è 'Errore durante l\{\}'eliminazione dell\{\}'allarme.'; await <code>firstValueFrom(service.error\$)</code> è null; await <code>firstValueFrom(service.alarms\$)</code> coincide con [alarmB]; <i>refreshServiceStub.requestRefresh</i> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-214	Contesto <i>AlarmConfigTablePre-senterService</i> ; caso 'normalizza gli spazi intorno ai separatori della posizione' : verifica che il valore di <code>row.position</code> è 'Appartamento 2 - Cucina - Sensore fumo' .	il valore di <code>row.position</code> è 'Appartamento 2 - Cucina - Sensore fumo' .	S
TU-215	Contesto <i>AlarmConfigTablePre-senterService</i> ; caso 'applica fallback su posizione quando è vuota' : verifica che il valore di <code>row.position</code> è '-' .	il valore di <code>row.position</code> è '-' .	S
TU-216	Contesto <i>AlarmConfigTablePre-senterService</i> ; caso 'mantiene fallback orario quando il valore non è ISO/HH:mm:ss' : verifica che il valore di <code>row.armedTime</code> è 'xx:yy' e il valore di <code>row.dearmedTime</code> è 'foo' .	il valore di <code>row.armedTime</code> è 'xx:yy' ; il valore di <code>row.dearmedTime</code> è 'foo' .	S
TU-217	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'extractDeviceOptions crea label stanza-dispositivo e normalizza enum duplicati' : verifica che <code>options</code> ha lunghezza 1 e il valore di <code>options[0].id</code> è 'dev-1' .	<code>options</code> ha lunghezza 1; il valore di <code>options[0].id</code> è 'dev-1' ; il valore di <code>options[0].label</code> è 'Soggiorno - Sensore UWB' ; <code>options[0].datapoints[0].enum</code> coincide con ['Off', 'On'] .	S
TU-218	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findReadableDatapoints ritorna solo datapoint leggibili' : verifica che <code>datapoints.map((datapoint) => datapoint.id)</code> coincide con ['dp-1', 'dp-2'] e il valore di <code>datapoints.every((datapoint) => datapoint.readable)</code> è true .	<code>datapoints.map((datapoint) => datapoint.id)</code> coincide con ['dp-1', 'dp-2'] ; il valore di <code>datapoints.every((datapoint) => datapoint.readable)</code> è true .	S
TU-219	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findReadableDatapoints ritorna lista vuota se il device non esiste' : verifica che <code>datapoints</code> coincide con [] .	<code>datapoints</code> coincide con [] .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-220	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'getAllowedOperators limita a uguale quando il datapoint ha enum': verifica che <i>service.getAllowedOperators(enumerationDatapoint)</i> coincide con [ThresholdOperator.EQUAL_TO] e <i>service.getAllowedOperators(numericDatapoint)</i> coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	<i>service.getAllowedOperators(enumerationDatapoint)</i> coincide con [ThresholdOperator.EQUAL_TO]; <i>service.getAllowedOperators(numericDatapoint)</i> coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	S
TU-221	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'isThresholdValueValid valida enum case-insensitive e soglie numeriche : verifica che il valore di <i>service.isThresholdValueValid(enumerationDatapoint, 'on')</i> è true; il valore di <i>service.isThresholdValueValid(enumerationDatapoint, 'Unknown')</i> è false; il valore di <i>service.isThresholdValueValid(numericDatapoint, 100)</i> è true; il valore di <i>service.isThresholdValueValid(numericDatapoint, 'Unknown')</i> è false.	il valore di <i>service.isThresholdValueValid(enumerationDatapoint, 'on')</i> è true; il valore di <i>service.isThresholdValueValid(enumerationDatapoint, 'Unknown')</i> è false; il valore di <i>service.isThresholdValueValid(numericDatapoint, 100)</i> è true; il valore di <i>service.isThresholdValueValid(numericDatapoint, 'Unknown')</i> è false.	S
TU-222	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findDatapointByDeviceAndDatapointId trova e normalizza il datapoint dal catalogo plant : verifica che datapoint non è null e il valore di <i>datapoint?.id</i> è 'dp-lookup'.	datapoint non è null; il valore di <i>datapoint?.id</i> è 'dp-lookup'; <i>datapoint?.enum</i> coincide con ['Off', 'On'].	S
TU-223	Contesto <i>DeviceDatapointExtractionService</i> ; caso 'findDatapointByDeviceAndDatapointId ritorna null se non trova corrispondenze': verifica che datapoint è null.	datapoint è null.	S
TU-224	Contesto <i>AlarmHistoryPageComponent</i> ; caso 'crea il componente e inzializza il service : verifica che component è valorizzato e <i>alarmHistoryServiceStub.initialize</i> viene invocato 1 volta.	component è valorizzato; <i>alarmHistoryServiceStub.initialize</i> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-225	Contesto <i>AlarmHistoryPageComponent</i> ; caso 'mappa e ordina le righe per closedAt desc': verifica che rows ha lunghezza 2 e il valore di rows[0].id è 'a-2'.	rows ha lunghezza 2; il valore di rows[0].id è 'a-2'; il valore di rows[1].id è 'a-1'; il valore di rows[0].status è 'Gestito'.	S
TU-226	Contesto <i>AlarmHistoryPageComponent</i> ; caso 'toRow applica fallback e stato per allarme non risolto': verifica che il valore di row.status è 'Da gestire' e il valore di row.closedAt è 'sconosciuto'.	il valore di row.status è 'Da gestire'; il valore di row.closedAt è 'sconosciuto'; il valore di row.handlerUsername è 'sconosciuto'.	S
TU-227	Contesto <i>AlarmHistoryPageComponent</i> ; caso 'delega next e previous page al service': verifica che alarmHistoryServiceStub.nextPage viene invocato 1 volta e alarmHistoryServiceStub.previousPage viene invocato 1 volta.	alarmHistoryServiceStub.nextPage viene invocato 1 volta; alarmHistoryServiceStub.previousPage viene invocato 1 volta.	S
TU-228	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'mappa un allarme aperto con azione gestibile': verifica che il valore di row.id è 'active-1' e il valore di row.status è 'Da gestire'.	il valore di row.id è 'active-1'; il valore di row.status è 'Da gestire'; il valore di row.isManaged è false; il valore di row.isResolving è false.	S
TU-229	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'mappa un allarme già gestito come non azionabile': verifica che il valore di row.status è 'Non da gestire' e il valore di row.isManaged è true.	il valore di row.status è 'Non da gestire'; il valore di row.isManaged è true; il valore di row.isActionDisabled è true; il valore di row.actionLabel è 'GESTITO'.	S
TU-230	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'quando l'allarme e in risoluzione mostra stato e label specifici': verifica che il valore di row.isResolving è true e il valore di row.isActionDisabled è true.	il valore di row.isResolving è true; il valore di row.isActionDisabled è true; il valore di row.actionLabel è 'GESTIONE...'; il valore di row.status è 'Da gestire'.	S
TU-231	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'applica fallback su posizione, manager e orari invalidi': verifica che il valore di row.location è '-' e il valore di row.device è '-'.	il valore di row.location è '-'; il valore di row.device è '-'; il valore di row.openedAt è 'abcde12345'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-232	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'normalizza nome allarme vuoto per testo riga e aria label': verifica che il valore di row.name è 'senza nome' e il valore di row.actionAriaLabel è 'Gestisci allarme senza nome'.	il valore di row.name è 'senza nome'; il valore di row.actionAriaLabel è 'Gestisci allarme senza nome'.	S
TU-233	Contesto <i>AlarmManagementTablePresenterService</i> ; caso 'gestisce posizione null senza eccezioni runtime': verifica che il valore di row.location è '-'.	il valore di row.location è '-'.	S
TU-234	Contesto <i>AlarmManagementService</i> ; caso 'non carica lo snapshot iniziale automaticamente alla creazione': verifica che <i>alarmApiStub.getActiveAlarms</i> non viene invocato e <i>alarmStateStub.setActiveAlarms</i> non viene invocato.	<i>alarmApiStub.getActiveAlarms</i> non viene invocato; <i>alarmStateStub.setActiveAlarms</i> non viene invocato.	S
TU-235	Contesto <i>AlarmManagementService</i> ; caso 'initialize per amministratore mostra tutti gli allarmi senza filtri frontend': verifica che <i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte e <i>alarmApiStub.getActiveAlarms</i> viene invocato con 99.	<i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte; <i>alarmApiStub.getActiveAlarms</i> viene invocato con 99; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; <i>alarmStateStub.setActiveAlarms</i> viene invocato con <i>initialSnapshot</i> .	S
TU-236	Contesto <i>AlarmManagementService</i> ; caso 'initialize per OSS usa endpoint unmanaged con userId numerico': verifica che <i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte e <i>alarmApiStub.getActiveAlarms</i> viene invocato con 7.	<i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte; <i>alarmApiStub.getActiveAlarms</i> viene invocato con 7; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; <i>alarmStateStub.setActiveAlarms</i> viene invocato con <i>initialSnapshot</i> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-237	Contesto <i>AlarmManagementService</i> ; caso 'se initialize fallisce non propaga eccezioni e valorizza l'errore nel vm': verifica che () => service.initialize() non genera l'errore atteso e alarmStateStub.setActiveAlarms non viene invocato.	() => service.initialize() non genera l'errore atteso; alarmStateStub.setActiveAlarms non viene invocato; il valore di vm.resolveError è 'bootstrap error'.	S
TU-238	Contesto <i>AlarmManagementService</i> ; caso 'resolveAlarm segue il flusso successo: resolving true, API call, update state, resolving false': verifica che il valore di vm.isResolving è true e il valore di vm.resolvingId è alarmA.id.	il valore di vm.isResolving è true; il valore di vm.resolvingId è alarmA.id; vm.resolveError è null; il valore di vm.isResolving è false.	S
TU-239	Contesto <i>AlarmManagementService</i> ; caso 'prima della resolve il vm riflette solo AlarmStateService': verifica che latestVm.alarms coincide con [alarmB] e vmHistory.at(-1) coincide con [alarmB].	latestVm.alarms coincide con [alarmB]; vmHistory.at(-1) coincide con [alarmB].	S
TU-240	Contesto <i>AlarmManagementService</i> ; caso 'dopo resolve non mantiene righe locali e riallinea lo stato allo snapshot backend': verifica che vm.alarms coincide con [] e alarmStateStub.setActiveAlarms viene invocato con [].	vm.alarms coincide con []; alarmStateStub.setActiveAlarms viene invocato con [].	S
TU-241	Contesto <i>AlarmManagementService</i> ; caso 'dopo resolve ricarica la pagina corrente dal backend': verifica che il valore di vm.alarms.length è 1 e il valore di vm.alarms[0].id è alarmA.id.	il valore di vm.alarms.length è 1; il valore di vm.alarms[0].id è alarmA.id; l'asserzione alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1 risulta soddisfatta; l'asserzione alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2 risulta soddisfatta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-242	Contesto <i>AlarmManagementService</i> ; caso 'dopo resolve su pagina non iniziale vuota torna automaticamente alla pagina precedente': verifica che l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta e l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta.	l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 3</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 4</i> risulta soddisfatta.	S
TU-243	Contesto <i>AlarmManagementService</i> ; caso 'nextPage usa limit 6 e incrementa offset quando la pagina corrente e piena': verifica che l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta e l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta.	l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 3</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 4</i> risulta soddisfatta.	S
TU-244	Contesto <i>AlarmManagementService</i> ; caso 'nextPage non fa nulla quando canGoNext è false': verifica che <i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte e l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta.	<i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; il valore di <i>vm.currentPage</i> è 1.	S
TU-245	Contesto <i>AlarmManagementService</i> ; caso 'la pagina successiva con meno di 6 elementi imposta canGoNext a false': verifica che il valore di <i>vm.currentPage</i> è 2 e il valore di <i>vm.canGoNext</i> è false.	il valore di <i>vm.currentPage</i> è 2; il valore di <i>vm.canGoNext</i> è false; <i>vm.alarms.map((alarm) => alarm.id)</i> coincide con ['active-7', 'active-8'].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-246	Contesto <i>AlarmManagementService</i> ; caso 'previousPage decrementa l'offset fino a 0': verifica che l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta e l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta.	l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 3</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 4</i> risulta soddisfatta.	S
TU-247	Contesto <i>AlarmManagementService</i> ; caso 'previousPage non fa nulla se l'offset è già a zero': verifica che <i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte e l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta.	<i>alarmApiStub.getActiveAlarms</i> viene invocato 2 volte; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 1</i> risulta soddisfatta; l'asserzione <i>alarmApiStub.getActiveAlarms toHaveBeenNthCalledWith 2</i> risulta soddisfatta.	S
TU-248	Contesto <i>AlarmManagementService</i> ; caso 'in errore resetta resolving e valorizza resolveError nel vm': verifica che il valore di <i>vm.isResolving</i> è false e <i>vm.resolvingId</i> è null.	il valore di <i>vm.isResolving</i> è false; <i>vm.resolvingId</i> è null; il valore di <i>vm.resolveError</i> è 'timeout rete'; <i>alarmStateStub.onAlarmResolved</i> non viene invocato.	S
TU-249	Contesto <i>AlarmManagementService</i> ; caso 'resolveAlarm con userId non numerico non chiama API e valorizza errore esplicito': verifica che il valore di <i>vm.isResolving</i> è false e <i>vm.resolvingId</i> è null.	il valore di <i>vm.isResolving</i> è false; <i>vm.resolvingId</i> è null; il valore di <i>vm.resolveError</i> è "Utente corrente non valido per risolvere l'allarme."; <i>alarmApiStub.resolveAlarm</i> non viene invocato.	S
TU-250	Contesto <i>AlarmManagementService</i> ; caso 'initialize con errore non strutturato usa fallback di caricamento': verifica che il valore di <i>vm.resolveError</i> è 'Errore durante il caricamento degli allarmi attivi?.'	il valore di <i>vm.resolveError</i> è 'Errore durante il caricamento degli allarmi attivi?.'	S

Codice	Descrizione	Esito atteso	Stato del test
TU-251	Contesto <i>AlarmManagementService</i> ; caso 'initialize con userId non numerico non chiama API e valorizza errore esplicito': verifica che <i>alarmApiStub.getActiveAlarms</i> non viene invocato e <i>alarmStateStub.setActiveAlarms</i> non viene invocato.	<i>alarmApiStub.getActiveAlarms</i> non viene invocato; <i>alarmStateStub.setActiveAlarms</i> non viene invocato; il valore di <i>vm.resolveError</i> è 'Utente corrente non valido per caricare gli allarmi attivi.'	S
TU-252	Contesto <i>AlarmManagementService</i> ; caso 'consente richieste concorrenti e termina resolving solo a completamento di tutte': verifica che il valore di <i>vm.isResolving</i> è true e il valore di <i>vm.resolvingId</i> è 'active-2'.	il valore di <i>vm.isResolving</i> è true; il valore di <i>vm.resolvingId</i> è 'active-2'; il valore di <i>vm.isResolving</i> è false; <i>vm.resolvingId</i> è null.	S
TU-253	Contesto <i>AlarmManagementService</i> ; caso 'resetta eventuale errore precedente quando parte una nuova <i>resolveAlarm</i> ': verifica che il valore di <i>vm.resolveError</i> è 'errore precedente' e <i>vm.resolveError</i> è null.	il valore di <i>vm.resolveError</i> è 'errore precedente'; <i>vm.resolveError</i> è null.	S
TU-254	Contesto <i>AlarmHistoryService</i> ; caso 'initialize carica la prima pagina con offset 0': verifica che <i>alarmApiStub.getResolvedAlarms</i> viene invocato con 99.	<i>alarmApiStub.getResolvedAlarms</i> viene invocato con 99.	S
TU-255	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con lista vuota non imposta <i>canGoNext</i> ': verifica che il valore di <i>vm.alarms.length</i> è 0 e <i>vm.canGoNext</i> è falsy.	il valore di <i>vm.alarms.length</i> è 0; <i>vm.canGoNext</i> è falsy.	S
TU-256	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con <i>pageLimit+1</i> elementi imposta <i>canGoNext</i> e fa lo slice': verifica che <i>vm.canGoNext</i> è valorizzato e il valore di <i>vm.alarms.length</i> è 6.	<i>vm.canGoNext</i> è valorizzato; il valore di <i>vm.alarms.length</i> è 6.	S
TU-257	Contesto <i>AlarmHistoryService</i> ; caso 'initialize con userId non numerico non chiama API': verifica che <i>alarmApiStub.getResolvedAlarms</i> non viene invocato.	<i>alarmApiStub.getResolvedAlarms</i> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-258	Contesto <i>AlarmHistoryService</i> ; caso 'initialize non propaga eccezioni al chiamante': verifica che () => service.initialize() non genera l'errore atteso.	() => service.initialize() non genera l'errore atteso.	S
TU-259	Contesto <i>AlarmHistoryService</i> ; caso 'nextPage avanza alla pagina successiva': verifica che il valore di vm.currentPage è 2 e il valore di vm.pageOffset è 6.	il valore di vm.currentPage è 2; il valore di vm.pageOffset è 6; vm.canGoPrevious è valorizzato; vm.alarms.map((a) => a.id) coincide con ['resolved-10'].	S
TU-260	Contesto <i>AlarmHistoryService</i> ; caso 'nextPage non fa nulla se canGoNext è false': verifica che alarmApiStub.getResolvedAlarms viene invocato 1 volta.	alarmApiStub.getResolvedAlarms viene invocato 1 volta.	S
TU-261	Contesto <i>AlarmHistoryService</i> ; caso 'pagina successiva con meno di pageLimit elementi imposta canGoNext a false': verifica che vm.canGoNext è falsy e vm.alarms.map((a) => a.id) coincide con ['resolved-10', 'resolved-11'].	vm.canGoNext è falsy; vm.alarms.map((a) => a.id) coincide con ['resolved-10', 'resolved-11'].	S
TU-262	Contesto <i>AlarmHistoryService</i> ; caso 'previousPage torna alla pagina precedente': verifica che il valore di vm.currentPage è 1 e il valore di vm.pageOffset è 0.	il valore di vm.currentPage è 1; il valore di vm.pageOffset è 0; vm.canGoPrevious è falsy; l'asserzione alarmApiStub.getResolvedAlarms toHaveBeenNthCalledWith 1 risulta soddisfatta.	S
TU-263	Contesto <i>AlarmHistoryService</i> ; caso 'previousPage non fa nulla se l'offset è già a zero': verifica che alarmApiStub.getResolvedAlarms viene invocato 1 volta.	alarmApiStub.getResolvedAlarms viene invocato 1 volta.	S
TU-264	Contesto <i>AnalyticsApiService</i> ; caso 'should be created': verifica che service è valorizzato.	service è valorizzato.	S
TU-265	Contesto <i>AnalyticsApiService - getAllApartments</i> ; caso 'should fetch all apartments and map them correctly': verifica che il valore di apartments.length è 2 e apartments[0] coincide con { id: '1', name: 'Apartment 1' }.	il valore di apartments.length è 2; apartments[0] coincide con { id: '1', name: 'Apartment 1' }; l'asserzione apartments[0] toHaveProperty 'extraField' risulta non soddisfatta; il valore di req.request.method è 'GET'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-266	Contesto <i>AnalyticsApiService - getAnalytics</i> ; caso 'should fetch analytics and map the complex DTO structure' : verifica che il valore di <code>result.apartmentId</code> è <code>apartmentId</code> e il valore di <code>result.analyticsInfo.length</code> è 1.	il valore di <code>result.apartmentId</code> è <code>apartmentId</code> ; il valore di <code>result.analyticsInfo.length</code> è 1; il valore di <code>info.title</code> è <code>'Test Chart'</code> ; <code>info.suggestions.messages</code> coincide con <code>['Suggestion 1']</code> .	S
TU-267	Contesto <i>ApartmentApiService</i> ; caso 'RF83-OBL abilita un appartamento nel Sistema' : verifica che il valore di <code>request.request.method</code> è <code>'PATCH'</code> e <code>request.request.body</code> coincide con <code>{}</code> .	il valore di <code>request.request.method</code> è <code>'PATCH'</code> ; <code>request.request.body</code> coincide con <code>{}</code> .	S
TU-268	Contesto <i>ApartmentApiService</i> ; caso 'RF84-OBL disabilita un appartamento nel Sistema' : verifica che il valore di <code>request.request.method</code> è <code>'PATCH'</code> e <code>request.request.body</code> coincide con <code>{}</code> .	il valore di <code>request.request.method</code> è <code>'PATCH'</code> ; <code>request.request.body</code> coincide con <code>{}</code> .	S
TU-269	Contesto Action Card Components; caso 'AlarmButtonCardComponent emette actionRequested con payload corretto' : verifica che <code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> e <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> ; <code>emitSpy</code> viene invocato 1 volta.	S
TU-270	Contesto Action Card Components; caso 'BlindCardComponent emette actionRequested con payload corretto' : verifica che <code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> e <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> ; <code>emitSpy</code> viene invocato 1 volta.	S
TU-271	Contesto Action Card Components; caso 'EntranceDoorCardComponent emette actionRequested con payload corretto' : verifica che <code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> e <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato con <code>{ roomId: 'room-1', deviceId: 'device-1', action: 'toggle', }</code> ; <code>emitSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-272	Contesto Action Card Components; caso <i>'FallSensorCardComponent'</i> emette <i>actionRequested</i> con <i>payload corretto'</i> : verifica che <i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', } e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', }; <i>emitSpy</i> viene invocato 1 volta.	S
TU-273	Contesto Action Card Components; caso <i>'LightCardComponent'</i> emette <i>actionRequested</i> con <i>payload corretto'</i> : verifica che <i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', } e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', }; <i>emitSpy</i> viene invocato 1 volta.	S
TU-274	Contesto Action Card Components; caso <i>'PresenceSensorCardComponent'</i> emette <i>actionRequested</i> con <i>payload corretto'</i> : verifica che <i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', } e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', }; <i>emitSpy</i> viene invocato 1 volta.	S
TU-275	Contesto Action Card Components; caso <i>'ThermostatCardComponent'</i> emette <i>actionRequested</i> con <i>payload corretto'</i> : verifica che <i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', } e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con { <i>roomId</i> : 'room-1', <i>deviceId</i> : 'device-1', <i>action</i> : 'toggle', }; <i>emitSpy</i> viene invocato 1 volta.	S
TU-276	Contesto Action Card Components; caso <i>'ThermostatCardComponent'</i> espone una <i>label valida per il tipo'</i> : verifica che l'asserzione <i>component.getTypeLabel(DeviceType.THERMOSTAT) toBeTypeOf 'string'</i> risulta soddisfatta e <i>component.getTypeLabel(DeviceType.THERMOSTAT).length</i> è maggiore di 0.	l'asserzione <i>component.getTypeLabel(DeviceType.THERMOSTAT) toBeTypeOf 'string'</i> risulta soddisfatta; <i>component.getTypeLabel(DeviceType.THERMOSTAT).length</i> è maggiore di 0.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-277	Contesto <i>DeviceApiService</i> ; caso 'mappa i datapoint dal payload plant nella stanza richiesta': verifica che il valore di request.request.method è 'GET' e il valore di room.id è 'room-1'.	il valore di request.request.method è 'GET'; il valore di room.id è 'room-1'; room.devices ha lunghezza 1; room.devices[0].datapoints ha lunghezza 1.	S
TU-278	Contesto <i>DeviceApiService</i> ; caso 'invia la scrittura datapoint su POST /api/device con payload corretto': verifica che il valore di request.request.method è 'POST' e request.request.body coincide con { datapointId: 'dp-1', value: 'On' }.	il valore di request.request.method è 'POST'; request.request.body coincide con { datapointId: 'dp-1', value: 'On' }.	S
TU-279	Contesto <i>DeviceApiService</i> ; caso 'propaga errore quando la scrittura datapoint fallisce': verifica che dopo rigetto, writePromise è valorizzato.	dopo rigetto, writePromise è valorizzato.	S
TU-280	Contesto <i>DeviceApiService</i> ; caso 'quando activePlantId manca, risolve il plant dall endpoint /plant/all': verifica che il valore di allPlantsRequest.request.method è 'GET' e il valore di plantByIdRequest.request.method è 'GET'.	il valore di allPlantsRequest.request.method è 'GET'; il valore di plantByIdRequest.request.method è 'GET'; rows coincide con []; il valore di storage.getItem('activePlantId') è 'plant-2'.	S
TU-281	Contesto <i>DeviceApiService</i> ; caso 'esclude dispositivi energy measure dalla tabella endpoint': verifica che il valore di request.request.method è 'GET' e rows ha lunghezza 1.	il valore di request.request.method è 'GET'; rows ha lunghezza 1; il valore di rows[0].deviceId è 'allowed-device'.	S
TU-282	Contesto <i>DeviceApiService</i> ; caso 'deduplica dispositivi con stesso nome mantenendo quello a priorità maggiore': verifica che il valore di request.request.method è 'GET' e rows ha lunghezza 1.	il valore di request.request.method è 'GET'; rows ha lunghezza 1; il valore di rows[0].deviceId è 'priority-device'.	S
TU-283	Contesto <i>DeviceApiService</i> ; caso 'recupera i valori correnti dai device unici e li mappa per datapoint': verifica che il valore di req1.request.method è 'GET' e il valore di req2.request.method è 'GET'.	il valore di req1.request.method è 'GET'; il valore di req2.request.method è 'GET'; il valore di values.get('dp-1') è 'On'; il valore di values.get('dp-2') è 21.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-284	Contesto <i>DeviceApiService</i> ; caso 'continua il merge valori anche se un device value endpoint fallisce': verifica che il valore di <code>values.get('dp-ok')</code> è 'Off'.	il valore di <code>values.get('dp-ok')</code> è 'Off'.	S
TU-285	Contesto <i>NavService</i> ; caso 'viene creato': verifica che service è valorizzato.	service è valorizzato.	S
TU-286	Contesto <i>NavService</i> ; caso 'restituisce tutti i NavItems per AMMINISTRATORE': verifica che <code>items.length</code> è maggiore di 0 e il valore di <code>items.some(i => i.label === "Dashboard")</code> è true.	<code>items.length</code> è maggiore di 0; il valore di <code>items.some(i => i.label === "Dashboard")</code> è true; il valore di <code>items.some(i => i.route.includes('dashboard'))</code> è true; il valore di <code>items.some(i => i.label === "Configurazione Allarmi")</code> è true.	S
TU-287	Contesto <i>NavService</i> ; caso 'restituisce solo i NavItems filtrati per OPERATORE SANITARIO': verifica che <code>OSSItems.length</code> è maggiore di 0 e il valore di <code>roleReqItems.length</code> è 0.	<code>OSSItems.length</code> è maggiore di 0; il valore di <code>roleReqItems.length</code> è 0; il valore di <code>items.some(i => i.label === "Dashboard")</code> è true; il valore di <code>items.some(i => i.route.includes('dashboard'))</code> è true.	S
TU-288	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'chiama GET /my-vimar/account in <code>getLinkedAccount</code> ': verifica che account coincide con <code>{ email: 'admin@example.com', isLinked: true }</code> e il valore di <code>request.request.method</code> è 'GET'.	account coincide con <code>{ email: 'admin@example.com', isLinked: true }</code> ; il valore di <code>request.request.method</code> è 'GET'.	S
TU-289	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'fallback su <code>/api/vimar-account</code> quando <code>/my-vimar/account</code> risponde 404': verifica che account coincide con <code>{ email: 'admin@example.com', isLinked: true }</code> e il valore di <code>firstRequest.request.method</code> è 'GET'.	account coincide con <code>{ email: 'admin@example.com', isLinked: true }</code> ; il valore di <code>firstRequest.request.method</code> è 'GET'; il valore di <code>fallbackRequest.request.method</code> è 'GET'.	S
TU-290	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'ritorna account scollegato quando entrambi gli endpoint non esistono': verifica che account coincide con <code>{ email: "", isLinked: false }</code> .	account coincide con <code>{ email: "", isLinked: false }</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-291	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'chiama DELETE /my-vimar/account in unlinkAccount': verifica che result è null e il valore di request.request.method è 'DELETE'.	result è null; il valore di request.request.method è 'DELETE'.	S
TU-292	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'fallback su DELETE /api/vimar-account quando /my-vimar/account non esiste': verifica che result è null e il valore di firstRequest.request.method è 'DELETE'.	result è null; il valore di firstRequest.request.method è 'DELETE'; il valore di fallbackRequest.request.method è 'DELETE'.	S
TU-293	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'richiede ticket OAuth e poi naviga su /api/auth/authorize': verifica che il valore di request.request.method è 'POST' e il valore di mockLocation.href è '\${baseUrl}/api/auth/authorize?ticket=ticket-123&redirect_url=http%3A%2F%2Flocalhost%3A4200%2F...'.	il valore di request.request.method è 'POST'; il valore di mockLocation.href è '\${baseUrl}/api/auth/authorize?ticket=ticket-123&redirect_url=http%3A%2F%2Flocalhost%3A4200%2F...'.	S
TU-294	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'mostra alert quando il backend restituisce ticket vuoto': verifica che mockAlert viene invocato 1 volta e mockAlert viene invocato con 'Ticket OAuth non valido. Riprova tra qualche istante.'.	mockAlert viene invocato 1 volta; mockAlert viene invocato con 'Ticket OAuth non valido. Riprova tra qualche istante.'; il valore di mockLocation.href è 'http://localhost:4200/vimar-link'.	S
TU-295	Contesto <i>MyVimarCloudApiFeatureService</i> ; caso 'mostra alert con messaggio mappato quando prepare-oauth fallisce': verifica che apiErrorDisplayServiceStub.toMessage viene invocato 1 volta e mockAlert viene invocato 1 volta.	apiErrorDisplayServiceStub.toMessage viene invocato 1 volta; mockAlert viene invocato 1 volta; mockAlert viene invocato con 'Errore avvio OAuth'; il valore di mockLocation.href è 'http://localhost:4200/vimar-link'.	S
TU-296	Contesto <i>NotificationService</i> ; caso 'espone vm iniziale vuoto quando storico e stream in-session sono vuoti': verifica che vm coincide con { notifications: [], unreadCount: 0, }.	vm coincide con { notifications: [], unreadCount: 0, }.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-297	Contesto <i>NotificationService</i> ; caso 'combina storico e in-session, deduplica per notificationId e ordina per sentAt desc': verifica che il valore di vm.unreadCount è 2 e vm.notifications coincide con [sharedInSession, inSessionOnly, historyOlder].	il valore di vm.unreadCount è 2; vm.notifications coincide con [sharedInSession, inSessionOnly, historyOlder]; vm.notifications ha lunghezza 3.	S
TU-298	Contesto <i>NotificationService</i> ; caso 'in errore sullo storico applica fallback a [] e mantiene i dati in-session': verifica che vm.notifications coincide con [inSessionOnly] e il valore di vm.unreadCount è 1.	vm.notifications coincide con [inSessionOnly]; il valore di vm.unreadCount è 1.	S
TU-299	Contesto <i>NotificationService</i> ; caso 'propaga nuovi unreadCount senza alterare la lista notifiche': verifica che vm.notifications coincide con [historyOlder] e il valore di vm.unreadCount è 5.	vm.notifications coincide con [historyOlder]; il valore di vm.unreadCount è 5.	S
TU-300	Contesto <i>NotificationService</i> ; caso 'aggiorna il vm quando arrivano nuove notifiche in-session dopo la prima emissione': verifica che vm.notifications coincide con [inSessionOnly, historyOlder] e il valore di vm.unreadCount è 0.	vm.notifications coincide con [inSessionOnly, historyOlder]; il valore di vm.unreadCount è 0.	S
TU-301	Contesto <i>NotificationService</i> ; caso 'deduplica tenendo la versione in-session quando notificationId è presente anche nello storico': verifica che vm.notifications coincide con [sharedInSession] e vm.notifications ha lunghezza 1.	vm.notifications coincide con [sharedInSession]; vm.notifications ha lunghezza 1.	S
TU-302	Contesto <i>NotificationService</i> ; caso 'gestisce timestamp non validi senza lanciare errori e li ordina in fondo': verifica che vm.notifications coincide con [historyOlder, invalidTimestampNotification].	vm.notifications coincide con [historyOlder, invalidTimestampNotification].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-303	Contesto <i>NotificationService</i> ; caso 'riusa la stessa pipeline vm con più subscriber senza richiamare di nuovo lo storico': verifica che vmA coincide con vmB e notificationApi-Stub. <i>getNotificationsHistory</i> viene invocato 1 volta.	vmA coincide con vmB; notificationApi-Stub. <i>getNotificationsHistory</i> viene invocato 1 volta.	S
TU-304	Contesto <i>NotificationService</i> ; caso 'mantiene uno snapshot coerente per subscriber tardivi senza ricaricare lo storico': verifica che vmLateSubscriber.notifications coincide con [inSessionOnly, historyOlder] e notificationApi-Stub. <i>getNotificationsHistory</i> viene invocato 1 volta.	vmLateSubscriber.notifications coincide con [inSessionOnly, historyOlder]; notificationApi-Stub. <i>getNotificationsHistory</i> viene invocato 1 volta.	S
TU-305	Contesto <i>NotificationService</i> ; caso 'non muta gli array sorgente ricevuti da API e <i>AlarmStateService</i> ': verifica che historicList coincide con [sharedHistoric, historyOlder] e inSessionList coincide con [sharedInSession, inSessionOnly].	historicList coincide con [sharedHistoric, historyOlder]; inSessionList coincide con [sharedInSession, inSessionOnly].	S
TU-306	Contesto <i>NotificationService</i> ; caso 'deduplica anche duplicati interni alla stessa sorgente mantenendo l'ultimo elemento': verifica che vm.notifications coincide con [duplicatedSessionB, duplicatedHistoryB] e vm.notifications ha lunghezza 2.	vm.notifications coincide con [duplicatedSessionB, duplicatedHistoryB]; vm.notifications ha lunghezza 2.	S
TU-307	Contesto <i>NotificationService</i> ; caso 'mantiene coerenza: unreadCount segue lo stream dedicato anche quando la deduplica riduce la lista': verifica che vm.notifications ha lunghezza 1 e il valore di vm.unreadCount è 5.	vm.notifications ha lunghezza 1; il valore di vm.unreadCount è 5.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-308	Contesto <i>NotificationService</i> ; caso 'filtra le notifiche cancellate localmente dall'archivio unificato': verifica che <code>vm.notifications.map((notification) => notification.notificationId)</code> coincide con ['session-1', 'shared-id',].	<code>vm.notifications.map((notification) => notification.notificationId)</code> coincide con ['session-1', 'shared-id',].	S
TU-309	Contesto <i>NotificationService</i> ; caso ' <i>removeNotification</i> delega <i>dismiss</i> + <i>remove</i> allo stato <i>realtime</i> ': verifica che <code>alarmStateStub.dismissNotification</code> viene invocato con 'n-100' e <code>alarmStateStub.removeNotification</code> viene invocato con 'n-100'.	<code>alarmStateStub.dismissNotification</code> viene invocato con 'n-100'; <code>alarmStateStub.removeNotification</code> viene invocato con 'n-100'.	S
TU-310	Contesto <i>NotificationService</i> ; caso ' <i>clearAllNotifications</i> delega <i>dismiss</i> multiplo + <i>clear</i> <i>realtime</i> ': verifica che <code>alarmStateStub.dismissNotifications</code> viene invocato con [<code>historyOlder.notificationId</code> , <code>inSessionOnly.notificationId</code> ,] e <code>alarmStateStub.clearNotifications</code> viene invocato 1 volta.	<code>alarmStateStub.dismissNotifications</code> viene invocato con [<code>historyOlder.notificationId</code> , <code>inSessionOnly.notificationId</code> ,]; <code>alarmStateStub.clearNotifications</code> viene invocato 1 volta.	S
TU-311	Contesto <i>UserApiService</i> ; caso ' <i>deleteUser</i> chiama DELETE /users/:id con id numerico': verifica che <code>result</code> è null e il valore di <code>request.request.method</code> è 'DELETE'.	<code>result</code> è null; il valore di <code>request.request.method</code> è 'DELETE'.	S
TU-312	Contesto <i>UserApiService</i> ; caso ' <i>createUser</i> decodifica <i>tempPassword</i> base64 prima di esporla al chiamante': verifica che il valore di <code>result.tempPassword</code> è 'TempPass123' e il valore di <code>request.request.method</code> è 'POST'.	il valore di <code>result.tempPassword</code> è 'TempPass123'; il valore di <code>request.request.method</code> è 'POST'; <code>request.request.body</code> coincide con <code>dto</code> .	S
TU-313	Contesto <i>UserApiService</i> ; caso ' <i>createUser</i> mantiene <i>tempPassword</i> invariata se non è base64 valida': verifica che il valore di <code>result.tempPassword</code> è 'TempPass123'.	il valore di <code>result.tempPassword</code> è 'TempPass123'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-314	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>assignOperator</i> esegue operazione e poi ricarica i wards': verifica che result è undefined e <i>apiStub.assignOperatorToWard</i> viene invocato con 1.	result è undefined; <i>apiStub.assignOperatorToWard</i> viene invocato con 1; l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> viene invocato con 1.	S
TU-315	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>removeOperator</i> in errore espone errore testo da <i>HttpErrorResponse.error</i> ': verifica che il valore di <i>emitted</i> è false e <i>storeStub.setError</i> viene invocato con 'Operatore non assegnato'.	il valore di <i>emitted</i> è false; <i>storeStub.setError</i> viene invocato con 'Operatore non assegnato'; <i>storeStub.setError</i> viene invocato 1 volta; <i>apiStub.getWards</i> non viene invocato.	S
TU-316	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>getAvailablePlantsForWard</i> filtra i plant già assegnati in qualunque ward': verifica che result coincide con [{ id: '203', name: 'App. 203' }] e l'asserzione <i>apiStub.getAvailablePlants toHaveBeenCalledOnce</i> risulta soddisfatta.	result coincide con [{ id: '203', name: 'App. 203' }]; l'asserzione <i>apiStub.getAvailablePlants toHaveBeenCalledOnce</i> risulta soddisfatta.	S
TU-317	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>dopo rimozione un appartamento torna disponibile</i> ': verifica che result coincide con [{ id: '202', name: 'App. 202' }] e result coincide con [{ id: '201', name: 'App. 201' }, { id: '202', name: 'App. 202' },].	result coincide con [{ id: '202', name: 'App. 202' }]; result coincide con [{ id: '201', name: 'App. 201' }, { id: '202', name: 'App. 202' },].	S
TU-318	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>dopo eliminazione reparto, gli appartamenti del reparto eliminato tornano disponibili</i> ': verifica che result coincide con [{ id: '203', name: 'App. 203' }] e result coincide con [{ id: '201', name: 'App. 201' }, { id: '203', name: 'App. 203' },].	result coincide con [{ id: '203', name: 'App. 203' }]; result coincide con [{ id: '201', name: 'App. 201' }, { id: '203', name: 'App. 203' },].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-319	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>assignPlant</i> in errore usa <i>HttpErrorResponse.message</i> quando non c'è error string': verifica che <i>messageArg.length</i> è maggiore di 0 e <i>apiStub.getWards</i> non viene invocato.	<i>messageArg.length</i> è maggiore di 0; <i>apiStub.getWards</i> non viene invocato; <i>storeStub.setWards</i> non viene invocato.	S
TU-320	Contesto <i>AssignmentOperationsService</i> ; caso ' <i>removePlant</i> in errore non-http usa fallback message': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione di assegnazione non riuscita.' e <i>storeStub.setError</i> viene invocato 1 volta.	<i>storeStub.setError</i> viene invocato con 'Operazione di assegnazione non riuscita.'; <i>storeStub.setError</i> viene invocato 1 volta; <i>apiStub.getWards</i> non viene invocato.	S
TU-321	Contesto <i>WardApiService</i> ; caso 'chiama GET /api/wards in <i>getWards</i> ': verifica che result coincide con [{ id: 1, name: 'Cardiologia', },] e result ha lunghezza 1.	result coincide con [{ id: 1, name: 'Cardiologia', },]; result ha lunghezza 1; il valore di <i>request.request.method</i> è 'GET'.	S
TU-322	Contesto <i>WardApiService</i> ; caso 'chiama GET /api/wards-users-relationships/:wardId in <i>getOperatorsByWardId</i> ': verifica che result coincide con [{ id: 1, username: 'mrossi' }] e il valore di <i>request.request.method</i> è 'GET'.	result coincide con [{ id: 1, username: 'mrossi' }]; il valore di <i>request.request.method</i> è 'GET'.	S
TU-323	Contesto <i>WardApiService</i> ; caso 'chiama GET /users in <i>getAvailableOperators</i> ': verifica che result coincide con [{ id: 2, username: 'lverde' }] e il valore di <i>request.request.method</i> è 'GET'.	result coincide con [{ id: 2, username: 'lverde' }]; il valore di <i>request.request.method</i> è 'GET'.	S
TU-324	Contesto <i>WardApiService</i> ; caso 'chiama GET /api/wards-plants-relationships/:wardId in <i>getPlantsByWardId</i> ': verifica che result coincide con [{ id: '101', name: 'App. 101' }] e il valore di <i>request.request.method</i> è 'GET'.	result coincide con [{ id: '101', name: 'App. 101' }]; il valore di <i>request.request.method</i> è 'GET'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-325	Contesto <i>WardApiService</i> ; caso 'chiama GET /api/plant/all in <i>getAvailablePlants</i> ': verifica che result coincide con [{ id: '103', name: 'App. 103' }] e il valore di request.request.method è 'GET'.	result coincide con [{ id: '103', name: 'App. 103' }]; il valore di request.request.method è 'GET'.	S
TU-326	Contesto <i>WardApiService</i> ; caso 'chiama POST /api/wards in <i>createWard</i> ': verifica che result coincide con ward e il valore di result.id è 1.	result coincide con ward; il valore di result.id è 1; il valore di request.request.method è 'POST'; request.request.body coincide con dto.	S
TU-327	Contesto <i>WardApiService</i> ; caso 'chiama PUT /api/wards/:id in <i>updateWard</i> ': verifica che result coincide con ward e il valore di request.request.method è 'PUT'.	result coincide con ward; il valore di request.request.method è 'PUT'; request.request.body coincide con { name: 'Cardio 2' }.	S
TU-328	Contesto <i>WardApiService</i> ; caso 'chiama DELETE /api/wards-users-relationships/:wardId/:userId': verifica che result è null e il valore di request.request.method è 'DELETE'.	result è null; il valore di request.request.method è 'DELETE'.	S
TU-329	Contesto <i>WardApiService</i> ; caso 'chiama POST /api/wards-plants-relationships con wardId e plantId': verifica che result è null e il valore di request.request.method è 'POST'.	result è null; il valore di request.request.method è 'POST'; request.request.body coincide con { wardId: 1, plantId: '102' }.	S
TU-330	Contesto <i>WardApiService</i> ; caso 'chiama DELETE /api/wards/:id in <i>deleteWard</i> ': verifica che result è null e il valore di request.request.method è 'DELETE'.	result è null; il valore di request.request.method è 'DELETE'.	S
TU-331	Contesto <i>WardApiService</i> ; caso 'chiama POST /api/wards-users-relationships con wardId e userId': verifica che result è null e il valore di request.request.method è 'POST'.	result è null; il valore di request.request.method è 'POST'; request.request.body coincide con { wardId: 1, userId: 9 }.	S
TU-332	Contesto <i>WardApiService</i> ; caso 'chiama DELETE /api/wards-plants-relationships/:plantId': verifica che result è null e il valore di request.request.method è 'DELETE'.	result è null; il valore di request.request.method è 'DELETE'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-333	Contesto <i>WardApiService</i> ; caso 'mantiene compatibilita con <i>assignPlantToWard</i> mappando <i>plantId -> plantId</i> ': verifica che result è null e il valore di <i>request.request.method</i> è 'POST'.	result è null; il valore di <i>request.request.method</i> è 'POST'; <i>request.request.body</i> coincide con { <i>wardId</i> : 1, <i>plantId</i> : '103' }.	S
TU-334	Contesto <i>WardManagementStore</i> ; caso ' <i>loadWards</i> imposta <i>loading</i> e delega a <i>WardOperationsService</i> ': verifica che <i>wardStoreStub.setLoading</i> viene invocato con true e <i>wardStoreStub.setLoading</i> viene invocato 1 volta.	<i>wardStoreStub.setLoading</i> viene invocato con true; <i>wardStoreStub.setLoading</i> viene invocato 1 volta; l'asserzione <i>wardOperationsStub.loadWards toHaveBeenCalledOnce</i> risulta soddisfatta.	S
TU-335	Contesto <i>WardManagementStore</i> ; caso ' <i>create/update/delete</i> delegano al service corretto': verifica che <i>wardOperationsStub.createWard</i> viene invocato con { <i>name</i> : 'Cardio' } e <i>wardOperationsStub.updateWard</i> viene invocato con 1.	<i>wardOperationsStub.createWard</i> viene invocato con { <i>name</i> : 'Cardio' }; <i>wardOperationsStub.updateWard</i> viene invocato con 1; <i>wardOperationsStub.deleteWard</i> viene invocato con 1; <i>wardStoreStub.setLoading</i> viene invocato 3 volte.	S
TU-336	Contesto <i>WardManagementStore</i> ; caso ' <i>assign/remove operator</i> e <i>plant</i> delegano ad <i>AssignmentOperationsService</i> ': verifica che <i>assignmentOperationsStub.assignOperator</i> viene invocato con 1 e <i>assignmentOperationsStub.removeOperator</i> viene invocato con 1.	<i>assignmentOperationsStub.assignOperator</i> viene invocato con 1; <i>assignmentOperationsStub.removeOperator</i> viene invocato con 1; <i>assignmentOperationsStub.assignPlant</i> viene invocato con 1; <i>assignmentOperationsStub.removePlant</i> viene invocato con 1.	S
TU-337	Contesto <i>WardManagementStore</i> ; caso ' <i>getAvailablePlantsForWard</i> delega ad <i>AssignmentOperationsService</i> ': verifica che <i>assignmentOperationsStub.getAvailablePlantsForWard</i> viene invocato con 10 e result coincide con [{ <i>id</i> : '200', <i>name</i> : 'App. 200' }].	<i>assignmentOperationsStub.getAvailablePlantsForWard</i> viene invocato con 10; result coincide con [{ <i>id</i> : '200', <i>name</i> : 'App. 200' }]; <i>wardStoreStub.setError</i> non viene invocato.	S
TU-338	Contesto <i>WardManagementStore</i> ; caso ' <i>getAvailablePlantsForWard</i> in errore ritorna null e setta errore': verifica che result è null e <i>wardStoreStub.setError</i> viene invocato con 'fetch failed'.	result è null; <i>wardStoreStub.setError</i> viene invocato con 'fetch failed'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-339	Contesto <i>WardManagementStore</i> ; caso <i>'getAvailablePlantsForWard con errore non strutturato usa il messaggio di fallback'</i> : verifica che result è null e <i>wardStoreStub.setError</i> viene invocato con <i>'Operazione plant non riuscita.'</i>	result è null; <i>wardStoreStub.setError</i> viene invocato con <i>'Operazione plant non riuscita.'</i>	S
TU-340	Contesto <i>WardManagementStore</i> ; caso <i>'getAvailableUsersForWard in errore ritorna null e setta errore'</i> : verifica che result è null e <i>wardStoreStub.setError</i> viene invocato con <i>'users fetch failed'</i> .	result è null; <i>wardStoreStub.setError</i> viene invocato con <i>'users fetch failed'</i> .	S
TU-341	Contesto <i>WardOperationsService</i> ; caso <i>'loadWards aggiorna store e termina senza valore'</i> : verifica che result è undefined e l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta.	result è undefined; l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> viene invocato con 1; <i>apiStub.getOperatorsByWardId</i> viene invocato con 1.	S
TU-342	Contesto <i>WardOperationsService</i> ; caso <i>'loadWards con lista vuota non richiama le relationship API'</i> : verifica che l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta e <i>apiStub.getPlantsByWardId</i> non viene invocato.	l'asserzione <i>apiStub.getWards toHaveBeenCalledOnce</i> risulta soddisfatta; <i>apiStub.getPlantsByWardId</i> non viene invocato; <i>apiStub.getOperatorsByWardId</i> non viene invocato; <i>storeStub.setWards</i> viene invocato con [].	S
TU-343	Contesto <i>WardOperationsService</i> ; caso <i>'createWard con 409 imposta messaggio funzionale'</i> : verifica che il valore di <i>emitted</i> è false e <i>storeStub.addWard</i> non viene invocato.	il valore di <i>emitted</i> è false; <i>storeStub.addWard</i> non viene invocato; <i>storeStub.setError</i> viene invocato con <i>'Esiste già un reparto con questo nome.'</i> ; <i>storeStub.setError</i> viene invocato 1 volta.	S
TU-344	Contesto <i>WardOperationsService</i> ; caso <i>'createWard normalizza payload minimale mantenendo il modello Ward coerente'</i> : verifica che <i>storeStub.addWard</i> viene invocato con { id: 11, name: 'Oncologia', apartments: [], operators: [], } e <i>storeStub.setLoading</i> viene invocato con false.	<i>storeStub.addWard</i> viene invocato con { id: 11, name: 'Oncologia', apartments: [], operators: [], }; <i>storeStub.setLoading</i> viene invocato con false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-345	Contesto <i>WardOperationsService</i> ; caso ' <i>createWard</i> preserva <i>apartments/operators</i> dallo <i>snapshot</i> se il backend non li restituisce': verifica che <i>storeStub.addWard</i> viene invocato con { id: 1, name: 'Cardiologia nuova', apartments: ward.apartments, operators: ward.operators, }.	<i>storeStub.addWard</i> viene invocato con { id: 1, name: 'Cardiologia nuova', apartments: ward.apartments, operators: ward.operators, }.	S
TU-346	Contesto <i>WardOperationsService</i> ; caso ' <i>updateWard</i> sostituisce il ward nello store': verifica che <i>apiStub.updateWard</i> viene invocato con 1 e <i>storeStub.replaceWard</i> viene invocato con updatedWard.	<i>apiStub.updateWard</i> viene invocato con 1; <i>storeStub.replaceWard</i> viene invocato con updatedWard; <i>storeStub.replaceWard</i> viene invocato 1 volta; <i>storeStub.setLoading</i> viene invocato con false.	S
TU-347	Contesto <i>WardOperationsService</i> ; caso ' <i>updateWard</i> preserva <i>apartments/operators</i> quando backend restituisce solo id e name': verifica che <i>storeStub.replaceWard</i> viene invocato con { id: 1, name: 'Cardiologia X', apartments: ward.apartments, operators: ward.operators, } e <i>storeStub.setLoading</i> viene invocato con false.	<i>storeStub.replaceWard</i> viene invocato con { id: 1, name: 'Cardiologia X', apartments: ward.apartments, operators: ward.operators, }; <i>storeStub.setLoading</i> viene invocato con false.	S
TU-348	Contesto <i>WardOperationsService</i> ; caso ' <i>updateWard</i> gestisce errore HTTP con messaggio testuale del server': verifica che <i>storeStub.setError</i> viene invocato con 'update failed' e <i>storeStub.replaceWard</i> non viene invocato.	<i>storeStub.setError</i> viene invocato con 'update failed'; <i>storeStub.replaceWard</i> non viene invocato.	S
TU-349	Contesto <i>WardOperationsService</i> ; caso ' <i>deleteWard</i> rimuove il ward dallo store': verifica che <i>apiStub.deleteWard</i> viene invocato con 1 e <i>storeStub.removeWard</i> viene invocato con 1.	<i>apiStub.deleteWard</i> viene invocato con 1; <i>storeStub.removeWard</i> viene invocato con 1; <i>storeStub.removeWard</i> viene invocato 1 volta; <i>storeStub.setLoading</i> viene invocato con false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-350	Contesto <i>WardOperationsService</i> ; caso ' <i>deleteWard</i> usa il fallback generico quando l'errore non è strutturato': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.' e <i>storeStub.removeWard</i> non viene invocato.	<i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'; <i>storeStub.removeWard</i> non viene invocato.	S
TU-351	Contesto <i>WardOperationsService</i> ; caso ' <i>createWard</i> usa il fallback generico quando l'errore non è strutturato': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'	<i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'	S
TU-352	Contesto <i>WardOperationsService</i> ; caso ' <i>in caso di errore generico usa il fallback message</i> ': verifica che <i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.' e <i>storeStub.setError</i> viene invocato 1 volta.	<i>storeStub.setError</i> viene invocato con 'Operazione sui reparti non riuscita.'; <i>storeStub.setError</i> viene invocato 1 volta; <i>storeStub.setWards</i> non viene invocato.	S
TU-353	Contesto <i>WardOperationsService</i> ; caso ' <i>loadWards normalizza apartments dal dto</i> ': verifica che <i>storeStub.setWards</i> viene invocato con [{ ...hydratedWard, apartments: [{ id: '101', name: 'App. 101' }], },].	<i>storeStub.setWards</i> viene invocato con [{ ...hydratedWard, apartments: [{ id: '101', name: 'App. 101' }], },].	S
TU-354	Contesto <i>WardStore</i> ; caso ' <i>inizializza stato vuoto</i> ': verifica che <i>await firstValueFrom(store.wards\$)</i> coincide con [] e il valore di <i>await firstValueFrom(store.isLoading\$)</i> è false.	<i>await firstValueFrom(store.wards\$)</i> coincide con []; il valore di <i>await firstValueFrom(store.isLoading\$)</i> è false; <i>await firstValueFrom(store.error\$)</i> è null.	S
TU-355	Contesto <i>WardStore</i> ; caso ' <i>setWards sovrascrive i wards</i> ': verifica che <i>wards</i> coincide con [wardA, wardB] e <i>wards</i> ha lunghezza 2.	<i>wards</i> coincide con [wardA, wardB]; <i>wards</i> ha lunghezza 2.	S
TU-356	Contesto <i>WardStore</i> ; caso ' <i>addWard aggiunge il reparto e resetta error/loading</i> ': verifica che <i>wards</i> coincide con [wardA] e <i>wards</i> ha lunghezza 1.	<i>wards</i> coincide con [wardA]; <i>wards</i> ha lunghezza 1; il valore di <i>await firstValueFrom(store.isLoading\$)</i> è false; <i>await firstValueFrom(store.error\$)</i> è null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-357	Contesto <i>WardStore</i> ; caso 'replaceWard aggiorna il reparto target' : verifica che wards coincide con [wardA, renamed] e wards ha lunghezza 2.	wards coincide con [wardA, renamed]; wards ha lunghezza 2.	S
TU-358	Contesto <i>WardStore</i> ; caso 'removeWard elimina il reparto target' : verifica che wards coincide con [wardB] e wards ha lunghezza 1.	wards coincide con [wardB]; wards ha lunghezza 1.	S
TU-359	Contesto <i>WardStore</i> ; caso 'patchPlant aggiorna solo l'appartamento richiesto' : verifica che wards ha lunghezza 2 e wards[0].apartments[0] coincide con wardA.apartments[0].	wards ha lunghezza 2; wards[0].apartments[0] coincide con wardA.apartments[0]; wards[1].apartments[0] coincide con { id: '102', name: 'App. 102A' }.	S
TU-360	Contesto <i>AlarmActionButtonComponent</i> ; caso 'renderizza label e aria-label di fallback' : verifica che button.textContent contiene 'GESTISCI' e il valore di button.getAttribute('aria-label') è 'GESTISCI'.	button.textContent contiene 'GESTISCI'; il valore di button.getAttribute('aria-label') è 'GESTISCI'; il valore di button.disabled è false.	S
TU-361	Contesto <i>AlarmActionButtonComponent</i> ; caso 'usa aria-label esplicita quando presente' : verifica che il valore di button.getAttribute('aria-label') è 'Gestisci allarme Antipanico'.	il valore di button.getAttribute('aria-label') è 'Gestisci allarme Antipanico'.	S
TU-362	Contesto <i>AlarmActionButtonComponent</i> ; caso 'applica la classe corretta per il tone primary' : verifica che component.buttonClass() contiene 'care-btn' e component.buttonClass() contiene 'care-btn-primary'.	component.buttonClass() contiene 'care-btn'; component.buttonClass() contiene 'care-btn-primary'.	S
TU-363	Contesto <i>AlarmActionButtonComponent</i> ; caso 'applica la classe corretta per il tone danger' : verifica che component.buttonClass() contiene 'care-btn' e component.buttonClass() contiene 'care-btn-danger'.	component.buttonClass() contiene 'care-btn'; component.buttonClass() contiene 'care-btn-danger'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-364	Contesto <i>AlarmActionButtonComponent</i> ; caso 'non emette click quando disabilitato': verifica che emitSpy non viene invocato e il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled</code> è true.	emitSpy non viene invocato; il valore di <code>(fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled</code> è true.	S
TU-365	Contesto <i>AlarmActionButtonComponent</i> ; caso 'emette pressed al click quando abilitato': verifica che emitSpy viene invocato 1 volta.	emitSpy viene invocato 1 volta.	S
TU-366	Contesto <i>AlarmPriorityIndicatorComponent</i> ; caso 'mappa la priorità rossa': verifica che il valore di <code>component.ui().label</code> è 'Alta' e il valore di <code>nativeElement.querySelector('[aria-label']?.getAttribute('aria-label')</code> è 'priorità Alta'.	il valore di <code>component.ui().label</code> è 'Alta'; il valore di <code>nativeElement.querySelector('[aria-label']?.getAttribute('aria-label')</code> è 'priorità Alta'; <code>nativeElement.textContent</code> contiene 'triangolo'.	S
TU-367	Contesto <i>AlarmPriorityIndicatorComponent</i> ; caso 'mappa la priorità arancione': verifica che il valore di <code>component.ui().label</code> è 'Media' e <code>fixture.nativeElement.textContent</code> contiene '!'.	il valore di <code>component.ui().label</code> è 'Media'; <code>fixture.nativeElement.textContent</code> contiene '!'.	S
TU-368	Contesto <i>AlarmPriorityIndicatorComponent</i> ; caso 'mappa la priorità verde': verifica che il valore di <code>component.ui().label</code> è 'Bassa' e <code>fixture.nativeElement.textContent</code> contiene 'pallino'.	il valore di <code>component.ui().label</code> è 'Bassa'; <code>fixture.nativeElement.textContent</code> contiene 'pallino'.	S
TU-369	Contesto <i>AlarmPriorityIndicatorComponent</i> ; caso 'mappa la priorità bianca': verifica che il valore di <code>component.ui().label</code> è 'Informativa' e <code>fixture.nativeElement.textContent</code> contiene 'i'.	il valore di <code>component.ui().label</code> è 'Informativa'; <code>fixture.nativeElement.textContent</code> contiene 'i'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-370	Contesto <i>AlarmTableShellComponent</i> ; caso 'renderizza intestazione, caption e contenuto proiettato': verifica che <code>nativeElement.textContent</code> contiene 'Allarmi attivi' e il valore di <code>nativeElement.querySelector('table')?.getAttribute('aria-label')</code> è 'Tabella test'.	<code>nativeElement.textContent</code> contiene 'Allarmi attivi'; il valore di <code>nativeElement.querySelector('table')?.getAttribute('aria-label')</code> è 'Tabella test'; <code>nativeElement.textContent</code> contiene 'Contenuto proiettato'.	S
TU-371	Contesto <i>AlarmTableShellComponent</i> ; caso 'mostra il pulsante create e emette <i>createRequested</i> al click': verifica che <i>createButton</i> non è null e <code>fixture.componentInstance.onCreateRequested</code> viene invocato 1 volta.	<i>createButton</i> non è null; <code>fixture.componentInstance.onCreateRequested</code> viene invocato 1 volta.	S
TU-372	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'mostra lo stato iniziale non attivo': verifica che il valore di <code>button.getAttribute('aria-pressed')</code> è 'false' e <code>component.trackClass()</code> contiene 'border-slate-500'.	il valore di <code>button.getAttribute('aria-pressed')</code> è 'false'; <code>component.trackClass()</code> contiene 'border-slate-500'; <code>component.thumbClass()</code> contiene 'translate-x-0.5'.	S
TU-373	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'mostra lo stato attivo e la classe corretta': verifica che il valore di <code>button.getAttribute('aria-pressed')</code> è 'true' e <code>component.trackClass()</code> contiene 'border-yellow-700'.	il valore di <code>button.getAttribute('aria-pressed')</code> è 'true'; <code>component.trackClass()</code> contiene 'border-yellow-700'; <code>component.thumbClass()</code> contiene 'translate-x-4'.	S
TU-374	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'emette il nuovo valore quando viene attivato': verifica che <code>emitSpy</code> viene invocato con true e <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato con true; <code>emitSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-375	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'non emette quando è disabilitato': verifica che emitSpy non viene invocato e il valore di (fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled è true.	emitSpy non viene invocato; il valore di (fixture.nativeElement.querySelector('button') as HTMLButtonElement).disabled è true.	S
TU-376	Contesto <i>AlarmToggleSwitchComponent</i> ; caso 'usa aria-label custom quando fornita': verifica che il valore di (fixture.nativeElement.querySelector('button') as HTMLButtonElement).getAttribute('aria-label') è 'Attiva allarme'.	il valore di (fixture.nativeElement.querySelector('button') as HTMLButtonElement).getAttribute('aria-label') è 'Attiva allarme'.	S
TU-377	Contesto <i>ConfirmDialogComponent</i> ; caso 'renderizza i valori di default': verifica che nativeElement.textContent contiene 'Confermi questa operazione?' e nativeElement.textContent contiene 'Conferma'.	nativeElement.textContent contiene 'Confermi questa operazione?'; nativeElement.textContent contiene 'Conferma'; nativeElement.textContent contiene 'Annulla'.	S
TU-378	Contesto <i>ConfirmDialogComponent</i> ; caso 'emette confirmed e cancelled ai click corretti': verifica che cancelledSpy viene invocato 1 volta e confirmedSpy viene invocato 1 volta.	cancelledSpy viene invocato 1 volta; confirmedSpy viene invocato 1 volta.	S
TU-379	Contesto <i>ConfirmDialogComponent</i> ; caso 'supporta label personalizzate': verifica che nativeElement.textContent contiene 'Procedere con la cancellazione?' e nativeElement.textContent contiene 'Sì, elimina'.	nativeElement.textContent contiene 'Procedere con la cancellazione?'; nativeElement.textContent contiene 'Sì, elimina'; nativeElement.textContent contiene 'No, torna indietro'.	S
TU-380	Contesto <i>ModalShellComponent</i> ; caso 'non renderizza il dialog quando open è false': verifica che fixture.nativeElement.querySelector('dialog') è null.	fixture.nativeElement.querySelector('dialog') è null.	

Codice	Descrizione	Esito atteso	Stato del test
TU-381	Contesto <i>ModalShellComponent</i> ; caso 'renderizza il dialog e la descrizione quando open è true': verifica che dialog non è null e il valore di <code>dialog?.getAttribute('aria-describedby')</code> è 'app-modal-shell-description'.	dialog non è null; il valore di <code>dialog?.getAttribute('aria-describedby')</code> è 'app-modal-shell-description'; <code>fixture.nativeElement.textContent</code> contiene 'Finestra test'; <code>fixture.nativeElement.textContent</code> contiene 'Descrizione di test'.	S
TU-382	Contesto <i>ModalShellComponent</i> ; caso 'non espone aria-describedby quando la descrizione è nulla': verifica che <code>(fixture.nativeElement.querySelector('dialog') as HTMLDialogElement null)?.getAttribute('ari... è null.</code>	<code>(fixture.nativeElement.querySelector('dialog') as HTMLDialogElement null)?.getAttribute('ari... è null.</code>	S
TU-383	Contesto <i>ModalShellComponent</i> ; caso 'emette closed al click sul pulsante di chiusura': verifica che <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato 1 volta.	S
TU-384	Contesto <i>AlarmConfigFormComponent</i> ; caso 'crea il componente': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-385	Contesto <i>AlarmConfigFormComponent</i> ; caso 'buildForm applica i validatori richiesti': verifica che il valore di <code>component.form.controls.plantId.invalid</code> è true e il valore di <code>component.form.controls.deviceId.invalid</code> è true.	il valore di <code>component.form.controls.plantId.invalid</code> è true; il valore di <code>component.form.controls.deviceId.invalid</code> è true; il valore di <code>component.form.controls.datapointId.invalid</code> è true; il valore di <code>component.form.controls.priority.invalid</code> è true.	S
TU-386	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in create mode mantiene il dispositivo bloccato finché non viene selezionato un plant': verifica che il valore di <code>component.form.controls.deviceId.disabled</code> è true e <code>apartmentApiStub.loadApartmentViewForPlantId</code> viene invocato con 'plant-1'.	il valore di <code>component.form.controls.deviceId.disabled</code> è true; <code>apartmentApiStub.loadApartmentViewForPlantId</code> viene invocato con 'plant-1'; il valore di <code>component.form.controls.deviceId.enabled</code> è true; il valore di <code>component.deviceOptions()[0]?.id</code> è 'sensor-1'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-387	Contesto <i>AlarmConfigFormComponent</i> ; caso 'mostra solo datapoint leggibili dopo la selezione dispositivo': verifica che component.datapointOptions().map((datapoint => datapoint.id) coincide con ['dp-readable-1', 'dp-readable-2',] e il valore di component.datapointOptions().every((datapoint => datapoint.readable) è true.	component.datapointOptions().map((datapoint => datapoint.id) coincide con ['dp-readable-1', 'dp-readable-2',]; il valore di component.datapointOptions().every((datapoint => datapoint.readable) è true.	S
TU-388	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in create mode non mostra il campo posizione read-only': verifica che positionInput è null.	positionInput è null.	S
TU-389	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode mostra il campo posizione read-only valorizzato': verifica che positionInput non è null e il valore di positionInput?.value è 'Appartamento 2 - Ingresso - Sensore porta'.	positionInput non è null; il valore di positionInput?.value è 'Appartamento 2 - Ingresso - Sensore porta'.	S
TU-390	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode non mostra il campo datapoint nel form': verifica che datapointInput è null.	datapointInput è null.	S
TU-391	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode mostra thresholdOperator come campo readonly': verifica che thresholdOperatorReadOnly non è null e il valore di thresholdOperatorReadOnly?.value è ThresholdOperator.EQUAL_TO.	thresholdOperatorReadOnly non è null; il valore di thresholdOperatorReadOnly?.value è ThresholdOperator.EQUAL_TO; thresholdOperatorSelect è null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-392	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode limita gli operatori al datapoint risolto e resetta operatori non supportati': verifica che <code>component.thresholdOperatorOptions()</code> coincide con <code>[ThresholdOperator.EQUAL_TO]</code> e <code>component.form.controls.thresholdOperator.value</code> è null.	<code>component.thresholdOperatorOptions()</code> S coincide con <code>[ThresholdOperator.EQUAL_TO]</code> ; <code>component.form.controls.thresholdOperator.value</code> è null.	S
TU-393	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode valida <code>thresholdValue</code> in base al datapoint risolto': verifica che il valore di <code>component.form.controls.thresholdValue.errors?.['invalidEnumThreshold']</code> è true.	il valore di <code>component.form.controls.thresholdValue.errors?.['invalidEnumThreshold']</code> è true.	S
TU-394	Contesto <i>AlarmConfigFormComponent</i> ; caso ' <i>onSubmit</i> emette <code>submittedForm</code> in create mode con form valido': verifica che <code>emitSpy</code> viene invocato con <code>validFormValue</code> e <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato con <code>validFormValue</code> ; <code>emitSpy</code> viene invocato 1 volta.	S
TU-395	Contesto <i>AlarmConfigFormComponent</i> ; caso ' <i>onSubmit</i> non invia se il form è invalido': verifica che <code>emitSpy</code> non viene invocato.	<code>emitSpy</code> non viene invocato.	S
TU-396	Contesto <i>AlarmConfigFormComponent</i> ; caso ' <i>onCancel</i> emette evento <code>cancelled</code> ': verifica che <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-397	Contesto <i>AlarmConfigFormComponent</i> ; caso 'espone opzioni enum per priorità e operatore': verifica che component.priorityOptions coincide con [AlarmPriority.WHITE, AlarmPriority.GREEN, AlarmPriority.ORANGE, AlarmPriority.RED,] e component.thresholdOperatorOptions() coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	component.priorityOptions coincide con [AlarmPriority.WHITE, AlarmPriority.GREEN, AlarmPriority.ORANGE, AlarmPriority.RED,]; component.thresholdOperatorOptions() coincide con [ThresholdOperator.GREATER_THAN, ThresholdOperator.GREATER_THAN_OR_EQUAL, ThresholdOperator.L...	S
TU-398	Contesto <i>AlarmConfigFormComponent</i> ; caso 'limita operatori e valore soglia quando il datapoint ha enum': verifica che component.thresholdOperatorOptions() coincide con [ThresholdOperator.EQUAL_TO] e il valore di component.form.controls.thresholdOperator.invalid è true.	component.thresholdOperatorOptions() coincide con [ThresholdOperator.EQUAL_TO]; il valore di component.form.controls.thresholdOperator.invalid è true; il valore di component.form.controls.thresholdValue.invalid è true; il valore di component.form.controls.thresholdValue.errors?.['invalidEnumThresholdOperator'] è true.	S
TU-399	Contesto <i>AlarmConfigFormComponent</i> ; caso 'accetta valore soglia booleano quando operatore è uguale': verifica che il valore di component.form.controls.thresholdOperator.valid è true e il valore di component.form.controls.thresholdValue.valid è true.	il valore di component.form.controls.thresholdOperator.valid è true; il valore di component.form.controls.thresholdValue.valid è true.	S
TU-400	Contesto <i>AlarmConfigFormComponent</i> ; caso 'carica impianti da /plant/all deduplicando e ordinando per nome': verifica che component.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3'] e component.plantsLoadError() è null.	component.plants().map((plant) => plant.id) coincide con ['plant-1', 'plant-2', 'plant-3']; component.plantsLoadError() è null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-401	Contesto <i>AlarmConfigFormComponent</i> ; caso 'se /plant/all fallisce usa fallback available + plants assegnati ai reparti': verifica che component.plants().map((plant) => plant.id) coincide con ['plant-7', 'plant-8', 'plant-9'] e component.plantsLoadError() è null.	component.plants().map((plant) => plant.id) coincide con ['plant-7', 'plant-8', 'plant-9']; component.plantsLoadError() è null; wardApi- Stub.getPlantsByWardId viene invocato 1 volta.	S
TU-402	Contesto <i>AlarmConfigFormComponent</i> ; caso 'se il plant viene deselezionato resetta opzioni dispositivo e blocca il campo dispositivo': verifica che il valore di component.deviceOptions()[0]?.id è 'sensor-1' e il valore di compo- nent.form.controls.deviceId.enabled è true.	il valore di component.deviceOptions()[0]?.id è 'sensor-1'; il valore di compo- nent.form.controls.deviceId.enabled è true; component.deviceOptions() coincide con []; il valore di compo- nent.form.controls.deviceId.value è ''.	S
TU-403	Contesto <i>AlarmConfigFormComponent</i> ; caso 'in edit mode non consente di modificare il dispositivo associato': verifica che il valore di compo- nent.form.controls.deviceId.disabled è true e component.deviceOptions() coincide con [{ id: 'sensor-9', label: 'sensor-9', datapoints: [] }].	il valore di compo- nent.form.controls.deviceId.disabled è true; component.deviceOptions() coincide con [{ id: 'sensor-9', label: 'sensor-9', datapoints: [] }].	S
TU-404	Contesto <i>AlarmConfigFormComponent</i> ; caso 'imposta errore quando il caricamento dispositivi fallisce': verifica che il valore di component.devicesLoadError() è 'Errore durante il caricamento dei dispositivi.' e il valore di compo- nent.form.controls.deviceId.disabled è true.	il valore di component.devicesLoadError() è 'Errore durante il caricamento dei dispositivi.'; il valore di compo- nent.form.controls.deviceId.disabled è true.	S
TU-405	Contesto <i>AlarmConfigPageComponent</i> ; caso 'crea il componente': verifica che component è valorizzato.	component è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-406	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>ngOnInit</i> carica gli allarmi e aggiorna le righe tabellari': verifica che <i>stateServiceStub.loadAlarmRules</i> viene invocato 1 volta e il valore di <i>component.rows().length</i> è 1.	<i>stateServiceStub.loadAlarmRules</i> viene invocato 1 volta; il valore di <i>component.rows().length</i> è 1; il valore di <i>component.error()</i> è 'Errore test'.	S
TU-407	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onCreateNew</i> apre la modale in create mode': verifica che il valore di <i>component.isModalOpen()</i> è true e <i>component.editingRule()</i> è null.	il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null; il valore di <i>component.modalTitle()</i> è 'Creazione allarme'.	S
TU-408	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onEdit</i> apre la modale in edit mode con regola selezionata': verifica che il valore di <i>component.isModalOpen()</i> è true e il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	il valore di <i>component.isModalOpen()</i> è true; il valore di <i>component.editingRule()?.id</i> è 'alarm-1'; il valore di <i>component.modalTitle()</i> è 'Modifica allarme'.	S
TU-409	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onEdit</i> ignora id non trovato': verifica che il valore di <i>component.isModalOpen()</i> è false e <i>component.editingRule()</i> è null.	il valore di <i>component.isModalOpen()</i> è false; <i>component.editingRule()</i> è null.	S
TU-410	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in create mode invoca create e chiude modale': verifica che <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> e il valore di <i>component.isModalOpen()</i> è false.	<i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> ; il valore di <i>component.isModalOpen()</i> è false.	S
TU-411	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in create mode mantiene aperta la modale se non arriva alcun esito': verifica che <i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> e il valore di <i>component.isModalOpen()</i> è true.	<i>stateServiceStub.createAlarmRule</i> viene invocato con <i>formValue</i> ; il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-412	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in edit mode invoca update e chiude modale': verifica che <i>stateServiceStub.updateAlarmRule</i> viene invocato con 'alarm-1' e il valore di <i>component.isModalOpen()</i> è false.	<i>stateServiceStub.updateAlarmRule</i> viene invocato con 'alarm-1'; il valore di <i>component.isModalOpen()</i> è false.	S
TU-413	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onFormSubmitted</i> in edit mode mantiene aperta la modale se non arriva alcun esito': verifica che il valore di <i>component.isModalOpen()</i> è true e il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	il valore di <i>component.isModalOpen()</i> è true; il valore di <i>component.editingRule()?.id</i> è 'alarm-1'.	S
TU-414	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onToggleEnabled</i> invoca <i>toggleEnabled</i> con lo stato richiesto': verifica che <i>stateServiceStub.toggleEnabled</i> viene invocato con 'alarm-1'.	<i>stateServiceStub.toggleEnabled</i> viene invocato con 'alarm-1'.	S
TU-415	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onToggleEnabled</i> blocca i click ripetuti finché la richiesta è in corso': verifica che il valore di <i>component.pendingToggleRuleId()</i> è 'alarm-1' e <i>stateServiceStub.toggleEnabled</i> viene invocato 1 volta.	il valore di <i>component.pendingToggleRuleId()</i> è 'alarm-1'; <i>stateServiceStub.toggleEnabled</i> viene invocato 1 volta; <i>component.pendingToggleRuleId()</i> è null; <i>stateServiceStub.toggleEnabled</i> viene invocato 2 volte.	S
TU-416	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDelete</i> apre la modale di conferma senza chiamare subito la delete': verifica che <i>component.pendingDelete()</i> coincide con { id: 'alarm-1', name: 'Temperatura alta' } e <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	<i>component.pendingDelete()</i> coincide con { id: 'alarm-1', name: 'Temperatura alta' }; <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-417	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDeleteConfirmed</i> invoca <i>deleteAlarmRule</i> con <i>id</i> e chiude la modale di conferma': verifica che <i>stateServiceStub.deleteAlarmRule</i> viene invocato con 'alarm-1' e <i>component.pendingDelete()</i> è null.	<i>stateServiceStub.deleteAlarmRule</i> viene invocato con 'alarm-1'; <i>component.pendingDelete()</i> è null.	S
TU-418	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onDeleteCancelled</i> chiude la modale di conferma senza eliminare': verifica che <i>component.pendingDelete()</i> è null e <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	<i>component.pendingDelete()</i> è null; <i>stateServiceStub.deleteAlarmRule</i> non viene invocato.	S
TU-419	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>renderizza la modale di conferma cancellazione quando richiesta</i> ': verifica che <i>confirmDialog</i> non è null e <i>content</i> contiene 'Confermi \{\}eliminazione della soglia "Temperatura alta"?'.	<i>confirmDialog</i> non è null; <i>content</i> contiene 'Confermi \{\}eliminazione della soglia "Temperatura alta"?'.	S
TU-420	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onModalClosed</i> chiude modale e resetta <i>editingRule</i> ': verifica che il valore di <i>component.isModalOpen()</i> è false e <i>component.editingRule()</i> è null.	il valore di <i>component.isModalOpen()</i> è false; <i>component.editingRule()</i> è null.	S
TU-421	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>onCreateNew</i> resetta l'eventuale regola in edit e apre la modale in create': verifica che il valore di <i>component.isModalOpen()</i> è true e <i>component.editingRule()</i> è null.	il valore di <i>component.isModalOpen()</i> è true; <i>component.editingRule()</i> è null.	S
TU-422	Contesto <i>AlarmConfigPageComponent</i> ; caso ' <i>renderizza la tabella con le righe e i pulsanti azione</i> ': verifica che il valore di <i>tableRows.length</i> è 1 e <i>content</i> contiene 'Temperatura alta'.	il valore di <i>tableRows.length</i> è 1; <i>content</i> contiene 'Temperatura alta'; <i>content</i> contiene 'Posizione'; <i>content</i> contiene 'Appartamento 1 - Soggiorno - Sensore porta'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-423	Contesto <i>AlarmConfigPageComponent</i> ; caso 'renderizza messaggio vuoto quando non ci sono regole': verifica che content contiene 'Nessuna regola configurata.'	content contiene 'Nessuna regola configurata.'	S
TU-424	Contesto <i>AlarmConfigPageComponent</i> ; caso 'renderizza il form in modale quando la modale è aperta': verifica che modalForm non è null.	modalForm non è null.	S
TU-425	Contesto <i>AlarmItemComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-426	Contesto <i>AlarmItemComponent</i> ; caso 'priorityUi mappa correttamente label per tutti i livelli': verifica che il valore di component.vm().priorityLabel è 'Alta' e il valore di component.vm().priorityLabel è 'Media'.	il valore di component.vm().priorityLabel è 'Alta'; il valore di component.vm().priorityLabel è 'Media'; il valore di component.vm().priorityLabel è 'Bassa'; il valore di component.vm().priorityLabel è 'Informativa'.	S
TU-427	Contesto <i>AlarmItemComponent</i> ; caso 'onResolveClick emette alarm.id': verifica che emitSpy viene invocato con 'active-1' e emitSpy viene invocato 1 volta.	emitSpy viene invocato con 'active-1'; emitSpy viene invocato 1 volta.	S
TU-428	Contesto <i>AlarmItemComponent</i> ; caso 'renderizza nome allarme, priorità label e tempo trasformato dalla pipe mockata': verifica che nativeElement.querySelector('.alarm-item__title)?.textContent contiene 'Allarme antipanico in "Camera 101"' e nativeElement.querySelector('.alarm-item__priority)?.textContent contiene 'Alta'.	nativeElement.querySelector('.alarm-item__title)?.textContent contiene 'Allarme antipanico in "Camera 101"'; nativeElement.querySelector('.alarm-item__priority)?.textContent contiene 'Alta'; metaValues.item(1)?.textContent contiene 'mock-elapsed:2026-03-24T10:00:00.000Z'; resolveButton?.getAttribute('aria-label') contiene 'Risolvi allarme Allarme antipanico'.	S
TU-429	Contesto <i>AlarmItemComponent</i> ; caso 'quando isResolving è true disabilita il bottone e mostra lo stato di avanzamento': verifica che button non è null e il valore di button?.hasAttribute('disabled') è true.	button non è null; il valore di button?.hasAttribute('disabled') è true; button?.textContent contiene 'Risoluzione...'; nativeElement.querySelector('.alarm-item__status)?.textContent contiene 'Operazione in corso'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-430	Contesto <i>AlarmItemComponent</i> ; caso 'quando <i>isResolving</i> è <i>false</i> abilita il bottone e non mostra lo stato di avanzamento': verifica che <i>button</i> non è null e il valore di <i>button?.hasAttribute('disabled')</i> è <i>false</i> .	<i>button</i> non è null; il valore di <i>button?.hasAttribute('disabled')</i> è <i>false</i> ; <i>button?.textContent</i> contiene 'Risolvi'; <i>nativeElement.querySelector('.alarm-item__status')</i> è null.	S
TU-431	Contesto <i>AlarmItemComponent</i> ; caso 'click sul bottone invia evento <i>resolve</i> con <i>alarm.id</i> ': verifica che <i>emitSpy</i> viene invocato con 'active-1' e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con 'active-1'; <i>emitSpy</i> viene invocato 1 volta.	S
TU-432	Contesto <i>AlarmItemComponent</i> ; caso 'usa <i>fallback</i> per nome e posizione quando i campi sono vuoti o solo spazi': verifica che il valore di <i>component.vm().alarmTitle</i> è 'senza nome in "posizione sconosciuta"' e il valore di <i>component.vm().articleAriaLabel</i> è 'Allarme senza nome'.	il valore di <i>component.vm().alarmTitle</i> è 'senza nome in "posizione sconosciuta"'; il valore di <i>component.vm().articleAriaLabel</i> è 'Allarme senza nome'; il valore di <i>component.vm().resolveButtonAriaLabel</i> è 'Risolvi allarme senza nome'.	S
TU-433	Contesto <i>AlarmItemComponent</i> ; caso 'quando è in <i>resolving</i> aggiorna <i>aria-label</i> del bottone in modo specifico': verifica che il valore di <i>component.vm().resolveButtonText</i> è 'Risoluzione...?' e <i>button?.getAttribute('aria-label')</i> contiene 'Risoluzione in corso per Allarme antipanico'.	il valore di <i>component.vm().resolveButtonText</i> è 'Risoluzione...?'; <i>button?.getAttribute('aria-label')</i> contiene 'Risoluzione in corso per Allarme antipanico'.	S
TU-434	Contesto <i>AlarmItemComponent</i> ; caso 'renderizza nomi con caratteri speciali come testo senza errori runtime': verifica che <i>title</i> contiene '<script>alert("xss")</script>' in "Sala monitoraggio".	<i>title</i> contiene '<script>alert("xss")</script>' in "Sala monitoraggio".	S
TU-435	Contesto <i>AlarmPageManagementComponent</i> ; caso 'should <i>create</i> ': verifica che <i>component</i> è valorizzato.	<i>component</i> è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-436	Contesto <i>AlarmPageManagementComponent</i> ; caso 'onResolve delega a facade.resolveAlarm' : verifica che <code>alarmManagementStub.resolveAlarm</code> viene invocato con <code>'active-2'</code> e <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	<code>alarmManagementStub.resolveAlarm</code> viene invocato con <code>'active-2'</code> ; <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	S
TU-437	Contesto <i>AlarmPageManagementComponent</i> ; caso 'renderizza stato lista vuota quando non ci sono allarmi' : verifica che <code>nativeElement.querySelector('.alarm-management__empty')</code> contiene <code>'Nessun allarme attivo al momento.'</code> e il valore di <code>nativeElement.querySelectorAll('tbody tr').length</code> è 0.	<code>nativeElement.querySelector('.alarm-management__empty')</code> contiene <code>'Nessun allarme attivo al momento.'</code> ; il valore di <code>nativeElement.querySelectorAll('tbody tr').length</code> è 0.	S
TU-438	Contesto <i>AlarmPageManagementComponent</i> ; caso 'renderizza errore e stato resolving quando presenti nel vm' : verifica che <code>nativeElement.querySelector('.alarm-management__error')</code> contiene <code>'Errore durante la risoluzione'</code> e <code>nativeElement.querySelector('.alarm-management__status')</code> contiene <code>'Risoluzione allarme in corso...'</code> .	<code>nativeElement.querySelector('.alarm-management__error')</code> contiene <code>'Errore durante la risoluzione'</code> ; <code>nativeElement.querySelector('.alarm-management__status')</code> contiene <code>'Risoluzione allarme in corso...'</code> .	S
TU-439	Contesto <i>AlarmPageManagementComponent</i> ; caso 'renderizza tabella con una riga per ogni allarme' : verifica che il valore di <code>rows.length</code> è 2 e <code>nativeElement.textContent</code> contiene <code>'Priorità'</code> .	il valore di <code>rows.length</code> è 2; <code>nativeElement.textContent</code> contiene <code>'Priorità'</code> ; <code>nativeElement.textContent</code> contiene <code>'Corridoio Nord'</code> ; <code>nativeElement.textContent</code> contiene <code>'mock-elapsed:2026-03-24T10:00:00.000Z'</code> .	S
TU-440	Contesto <i>AlarmPageManagementComponent</i> ; caso 'nasconde la colonna gestore per operatore sanitario' : verifica che <code>nativeElement.textContent</code> non contiene <code>'Gestore'</code> e <code>nativeElement.textContent</code> non contiene <code>'oss_1'</code> .	<code>nativeElement.textContent</code> non contiene <code>'Gestore'</code> ; <code>nativeElement.textContent</code> non contiene <code>'oss_1'</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-441	Contesto <i>AlarmPageManagementComponent</i> ; caso 'disabilita solo il bottone associato al resolvingId corrente': verifica che il valore di <code>firstButton?.disabled</code> è true e il valore di <code>secondButton?.disabled</code> è false.	il valore di <code>firstButton?.disabled</code> è true; il valore di <code>secondButton?.disabled</code> è false.	S
TU-442	Contesto <i>AlarmPageManagementComponent</i> ; caso 'renderizza paginazione coerente con vm e delega i click ai metodi del service': verifica che <code>nativeElement.querySelector('.care-pagination__status')?.textContent</code> contiene 'Pagina 3' e il valore di <code>previousButton?.disabled</code> è false.	<code>nativeElement.querySelector('.care-pagination__status')?.textContent</code> contiene 'Pagina 3'; il valore di <code>previousButton?.disabled</code> è false; il valore di <code>nextButton?.disabled</code> è true; <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	S
TU-443	Contesto <i>AlarmPageManagementComponent</i> ; caso 'click su GESTISCI propaga resolve verso facade': verifica che <code>alarmManagementStub.resolveAlarm</code> viene invocato con 'active-1' e <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	<code>alarmManagementStub.resolveAlarm</code> viene invocato con 'active-1'; <code>alarmManagementStub.resolveAlarm</code> viene invocato 1 volta.	S
TU-444	Contesto <i>AlarmPageManagementComponent</i> ; caso 'mantiene la riga visibile se l'allarme è già gestito e disabilita l'azione': verifica che il valore di <code>rows.length</code> è 1 e <code>nativeElement.textContent</code> contiene 'Non da gestire'.	il valore di <code>rows.length</code> è 1; <code>nativeElement.textContent</code> contiene 'Non da gestire'; <code>managedButton</code> non è null; il valore di <code>managedButton?.disabled</code> è true.	S
TU-445	Contesto <i>AlarmPageManagementComponent</i> ; caso 'onNextPage e onPreviousPage delegano al service': verifica che <code>alarmManagementStub.nextPage</code> viene invocato 1 volta e <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	<code>alarmManagementStub.nextPage</code> viene invocato 1 volta; <code>alarmManagementStub.previousPage</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-446	Contesto <i>AlarmPageManagementComponent</i> ; caso 'quando arriva <i>refreshRequested</i> reinizializza la pagina': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta e <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	<i>alarmManagementStub.initialize</i> viene invocato 1 volta; <i>alarmManagementStub.initialize</i> viene invocato 2 volte.	S
TU-447	Contesto <i>AlarmPageManagementComponent</i> ; caso 'dopo <i>destroy</i> non risponde più agli eventi di <i>refresh</i> ': verifica che <i>alarmManagementStub.initialize</i> viene invocato 1 volta.	<i>alarmManagementStub.initialize</i> viene invocato 1 volta.	S
TU-448	Contesto <i>AlarmsSentResolved-ChartComponent</i> ; caso 'espone metadati <i>chart</i> consistenti': verifica che il valore di <i>component.chartType</i> è 'bar' e <i>component.description.length</i> è maggiore di 0.	il valore di <i>component.chartType</i> è 'bar'; <i>component.description.length</i> è maggiore di 0.	S
TU-449	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe <i>responsive lg:w-[70%]</i> quando i suggerimenti sono attivi.	l'elemento [<i>data-test="chart-container"</i>] include la classe <i>lg:w-[70%]</i> .	S
TU-450	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [<i>data-test="suggestions-sidebar"</i>] è presente e non risulta null.	S
TU-451	Contesto <i>AlarmsSentResolved-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [<i>data-test="suggestion-item"</i>] sono esattamente 2, in linea con i messaggi forniti.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-452	Contesto <i>AlarmsSentResolved-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S
TU-453	Contesto <i>AlarmsSentResolved-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (<code>.lg:w-[30%]</code>) è assente e la query restituisce null.	S
TU-454	Contesto <i>AlarmFrequencyChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di <code>component.chartType</code> è 'bar' e <code>component.description.length</code> è maggiore di 0.	il valore di <code>component.chartType</code> è 'bar'; <code>component.description.length</code> è maggiore di 0.	S
TU-455	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive <code>lg:w-[70%]</code> quando i suggerimenti sono attivi.	l'elemento <code>[data-test="chart-container"]</code> include la classe <code>lg:w-[70%]</code> .	S
TU-456	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento <code>[data-test="suggestions-sidebar"]</code> è presente e non risulta null.	S
TU-457	Contesto <i>AlarmFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi <code>[data-test="suggestion-item"]</code> sono esattamente 2, in linea con i messaggi forniti.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-458	Contesto <i>AlarmFrequencyChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TU-459	Contesto <i>AlarmFrequencyChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TU-460	Contesto <i>EnergyConsumption-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'line' e component.description.length è maggiore di 0.	il valore di component.chartType è 'line'; component.description.length è maggiore di 0.	S
TU-461	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TU-462	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-463	Contesto <i>EnergyConsumption-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TU-464	Contesto <i>EnergyConsumption-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TU-465	Contesto <i>EnergyConsumption-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TU-466	Contesto <i>FallFrequencyChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S
TU-467	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S

Codice	Descrizione	Esito atteso	Stato del test
TU-468	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S
TU-469	Contesto <i>FallFrequencyChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TU-470	Contesto <i>FallFrequencyChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TU-471	Contesto <i>FallFrequencyChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S
TU-472	Contesto <i>PlantAnomaliesChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di component.chartType è 'bar' e component.description.length è maggiore di 0.	il valore di component.chartType è 'bar'; component.description.length è maggiore di 0.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-473	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive lg:w-[70%] quando i suggerimenti sono attivi.	l'elemento [data-test="chart-container"] include la classe lg:w-[70%].	S
TU-474	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [data-test="suggestions-sidebar"] è presente e non risulta null.	S
TU-475	Contesto <i>PlantAnomaliesChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [data-test="suggestion-item"] sono esattamente 2, in linea con i messaggi forniti.	S
TU-476	Contesto <i>PlantAnomaliesChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe .w-full, quindi la vista occupa il 100% della larghezza.	S
TU-477	Contesto <i>PlantAnomaliesChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (.lg:w-[30%]) è assente e la query restituisce null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-478	Contesto <i>PresenceDetection-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di <code>component.chartType</code> è 'bar' e <code>component.description.length</code> è maggiore di 0.	il valore di <code>component.chartType</code> è 'bar'; <code>component.description.length</code> è maggiore di 0.	S
TU-479	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe <code>responsive lg:w-[70%]</code> quando i suggerimenti sono attivi.	l'elemento [<code>data-test="chart-container"</code>] include la classe <code>lg:w-[70%]</code> .	S
TU-480	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [<code>data-test="suggestions-sidebar"</code>] è presente e non risulta null.	S
TU-481	Contesto <i>PresenceDetection-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [<code>data-test="suggestion-item"</code>] sono esattamente 2, in linea con i messaggi forniti.	S
TU-482	Contesto <i>PresenceDetection-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S
TU-483	Contesto <i>PresenceDetection-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (<code>.lg:w-[30%]</code>) è assente e la query restituisce null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-484	Contesto <i>ProlongedPresence-ChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di <code>component.chartType</code> è 'bar' e <code>component.description.length</code> è maggiore di 0.	il valore di <code>component.chartType</code> è 'bar'; <code>component.description.length</code> è maggiore di 0.	S
TU-485	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive <code>lg:w-[70%]</code> quando i suggerimenti sono attivi.	l'elemento [<code>data-test="chart-container"</code>] include la classe <code>lg:w-[70%]</code> .	S
TU-486	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [<code>data-test="suggestions-sidebar"</code>] è presente e non risulta null.	S
TU-487	Contesto <i>ProlongedPresence-ChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [<code>data-test="suggestion-item"</code>] sono esattamente 2, in linea con i messaggi forniti.	S
TU-488	Contesto <i>ProlongedPresence-ChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S
TU-489	Contesto <i>ProlongedPresence-ChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (<code>.lg:w-[30%]</code>) è assente e la query restituisce null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-490	Contesto <i>TemperatureVariation-sChartComponent</i> ; caso 'espone metadati chart consistenti': verifica che il valore di <code>component.chartType</code> è 'line' e <code>component.description.length</code> è maggiore di 0.	il valore di <code>component.chartType</code> è 'line'; <code>component.description.length</code> è maggiore di 0.	S
TU-491	Contesto <i>TemperatureVariation-sChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe restringere il grafico al 70% su schermi grandi': verifica che il contenitore del grafico applica la classe responsive <code>lg:w-[70%]</code> quando i suggerimenti sono attivi.	l'elemento [<code>data-test="chart-container"</code>] include la classe <code>lg:w-[70%]</code> .	S
TU-492	Contesto <i>TemperatureVariation-sChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe mostrare la colonna dei suggerimenti': verifica che la sidebar dei suggerimenti viene renderizzata nel DOM quando la modalità suggerimenti è attiva.	l'elemento [<code>data-test="suggestions-sidebar"</code>] è presente e non risulta null.	S
TU-493	Contesto <i>TemperatureVariation-sChartComponent</i> - con suggerimenti attivi; caso 'dovrebbe renderizzare il numero corretto di messaggi': verifica che la lista dei suggerimenti mostra esattamente il numero di messaggi previsto dai dati di input.	gli elementi [<code>data-test="suggestion-item"</code>] sono esattamente 2, in linea con i messaggi forniti.	S
TU-494	Contesto <i>TemperatureVariation-sChartComponent</i> - senza suggerimenti; caso 'dovrebbe espandere il grafico al 100% della larghezza': verifica che senza suggerimenti il contenitore del grafico occupa tutta la larghezza disponibile.	è presente un contenitore grafico con classe <code>.w-full</code> , quindi la vista occupa il 100% della larghezza.	S
TU-495	Contesto <i>TemperatureVariation-sChartComponent</i> - senza suggerimenti; caso 'NON dovrebbe esistere la colonna dei suggerimenti nel DOM': verifica che senza suggerimenti la colonna laterale non viene montata nel DOM.	la colonna suggerimenti (<code>.lg:w-[30%]</code>) è assente e la query restituisce null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-496	Contesto <i>ApartmentMonitorComponent</i> ; caso ' crea il componente e sincronizza <i>activeApartmentId</i> dal feed <i>apartment\$</i> ': verifica che <i>component</i> è valorizzato e il valore di <i>component.activeApartmentId</i> è 'ap-1'.	<i>component</i> è valorizzato; il valore di <i>component.activeApartmentId</i> è 'ap-1'; <i>apartmentApi-</i> <i>Stub.getCurrentApartment</i> viene invocato.	S
TU-497	Contesto <i>ApartmentMonitorComponent</i> ; caso ' onApartmentSelected <i>ignora id vuoto o uguale</i> ': verifica che <i>apartmentApi-</i> <i>Stub.setActivePlantId</i> non viene invocato.	<i>apartmentApiStub.setActivePlantId</i> non viene invocato.	S
TU-498	Contesto <i>ApartmentMonitorComponent</i> ; caso ' onApartmentSelected <i>cambia appartamento,</i> <i>resetta errore e attiva</i> <i>refresh</i> ': verifica che <i>apartmen-</i> <i>tApiStub.setActivePlantId</i> viene invocato con 'ap-2' e il valore di <i>component.error</i> è ''.	<i>apartmentApiStub.setActivePlantId</i> viene invocato con 'ap-2'; il valore di <i>component.error</i> è ''.	S
TU-499	Contesto <i>ApartmentMonitorComponent</i> ; caso ' refresh realtime <i>azzerà errore e ricarica</i> <i>apartment\$</i> ': verifica che il valore di <i>component.error</i> è '' e <i>apartmentApi-</i> <i>Stub.getCurrentApartment.mock.calls.length</i> è maggiore di <i>callsBefore</i> .	il valore di <i>component.error</i> è ''; <i>apartmentApi-</i> <i>Stub.getCurrentApartment.mock.calls.length</i> è maggiore di <i>callsBefore</i> .	S
TU-500	Contesto <i>RoomListComponent</i> ; caso ' crea il componente ': verifica che <i>component</i> è valorizzato.	<i>component</i> è valorizzato.	S
TU-501	Contesto <i>RoomListComponent</i> ; caso ' onRoomSelect <i>emette</i> <i>l'id selezionato</i> ': verifica che <i>emitSpy</i> viene invocato con 'room-77' e <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato con 'room-77'; <i>emitSpy</i> viene invocato 1 volta.	S
TU-502	Contesto <i>AlarmWidgetComponent</i> ; caso ' dovrebbe essere creato ': verifica che <i>component</i> è valorizzato.	<i>component</i> è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-503	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe inizializzare il servizio al caricamento (<i>ngOnInit</i>)': verifica che <i>mockAlarmService.initialize</i> viene invocato.	<i>mockAlarmService.initialize</i> viene invocato.	S
TU-504	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe mostrare lo stato di caricamento se <i>vm</i> è <i>null</i> ': verifica che <i>compiled.textContent</i> contiene 'Caricamento allarmi in corso'.	<i>compiled.textContent</i> contiene 'Caricamento allarmi in corso'.	S
TU-505	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe mostrare il messaggio "Nessun allarme" se la lista è vuota': verifica che <i>compiled.textContent</i> contiene 'Nessun allarme attivo al momento'.	<i>compiled.textContent</i> contiene 'Nessun allarme attivo al momento'.	S
TU-506	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe chiamare <i>resolveAlarm</i> quando <i>onResolve</i> viene invocato': verifica che <i>mockAlarmService.resolveAlarm</i> viene invocato con <i>alarmId</i> .	<i>mockAlarmService.resolveAlarm</i> viene invocato con <i>alarmId</i> .	S
TU-507	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe ricaricare i dati quando il <i>refresh service</i> emette': verifica che <i>mockAlarmService.initialize</i> viene invocato.	<i>mockAlarmService.initialize</i> viene invocato.	S
TU-508	Contesto <i>AlarmWidgetComponent</i> ; caso 'dovrebbe calcolare le righe correttamente tramite il <i>presenter</i> ': verifica che <i>mockPresenter.toRows</i> viene invocato con <i>alarms</i> .	<i>mockPresenter.toRows</i> viene invocato con <i>alarms</i> .	S
TU-509	Contesto <i>AnalyticsComponent</i> ; caso 'viene creato': verifica che <i>component</i> è valorizzato.	<i>component</i> è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-510	Contesto <i>AnalyticsComponent</i> ; caso 'carica il primo appartamento di default': verifica che <code>mockAnalyticsService.getAllApartments</code> viene invocato e <code>mockAnalyticsService.getAnalytics</code> viene invocato con '1'.	<code>mockAnalyticsService.getAllApartments</code> viene invocato; <code>mockAnalyticsService.getAnalytics</code> viene invocato con '1'.	S
TU-511	Contesto <i>AnalyticsComponent</i> ; caso 'aggiorna grafici al cambio appartamento': verifica che <code>mockAnalyticsService.getAnalytics</code> viene invocato con '2'.	<code>mockAnalyticsService.getAnalytics</code> viene invocato con '2'.	S
TU-512	Contesto <i>AnalyticsComponent</i> ; caso 'should return undefined if metric is not found': verifica che <code>result</code> è undefined.	<code>result</code> è undefined.	S
TU-513	Contesto <i>AnalyticsComponent</i> ; caso 'should find the correct metric in <i>getChartByMetric</i> ': verifica che <code>result</code> è definito e il valore di <code>result?.metric</code> è 'plant-anomalies'.	<code>result</code> è definito; il valore di <code>result?.metric</code> è 'plant-anomalies'.	S
TU-514	Contesto <i>DashboardComponent</i> ; caso 'dovrebbe creare il componente dashboard': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-515	Contesto <i>DashboardComponent</i> ; caso 'dovrebbe renderizzare il widget degli allarmi tramite il selettore': verifica che <code>alarmWidget</code> non è null.	<code>alarmWidget</code> non è null.	S
TU-516	Contesto <i>DashboardComponent</i> ; caso 'dovrebbe renderizzare il widget degli analytics tramite il selettore': verifica che <code>analyticsWidget</code> non è null.	<code>analyticsWidget</code> non è null.	S
TU-517	Contesto <i>PlantOverviewComponent</i> ; caso '1. Dovrebbe mostrare il nome dell'impianto': verifica che <code>html.textContent</code> contiene 'Casa Test'.	<code>html.textContent</code> contiene 'Casa Test'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-518	Contesto <i>PlantOverviewComponent</i> ; caso '2. Dovrebbe mostrare i nomi delle stanze': verifica che <code>html.textContent</code> contiene 'Soggiorno' e <code>html.textContent</code> contiene 'Camera'.	<code>html.textContent</code> contiene 'Soggiorno'; <code>html.textContent</code> contiene 'Camera'.	S
TU-519	Contesto <i>PlantOverviewComponent</i> ; caso '3. Dovrebbe gestire il caso senza dispositivi': verifica che <code>html.textContent</code> contiene 'Nessun dispositivo configurato'.	<code>html.textContent</code> contiene 'Nessun dispositivo configurato'.	S
TU-520	Contesto <i>PlantOverviewComponent</i> ; caso '4. Non dovrebbe mostrare nulla se <code>apartment</code> è null': verifica che <code>html.querySelector('header')</code> è null.	<code>html.querySelector('header')</code> è null.	S
TU-521	Contesto <i>DeviceCardComponent</i> ; caso 'filtra solo datapoint scrivibili con valori enum': verifica che <code>writable</code> ha lunghezza 1 e il valore di <code>writable[0].id</code> è 'dp-writable'.	<code>writable</code> ha lunghezza 1; il valore di <code>writable[0].id</code> è 'dp-writable'; il valore di <code>component.hasWritableDatapoints()</code> è true.	S
TU-522	Contesto <i>DeviceCardComponent</i> ; caso ' <i>getSelectedValue</i> usa fallback iniziale e poi mantiene il valore selezionato': verifica che il valore di <code>firstRead</code> è 'OFF' e il valore di <code>secondRead</code> è 'ON'.	il valore di <code>firstRead</code> è 'OFF'; il valore di <code>secondRead</code> è 'ON'.	S
TU-523	Contesto <i>DeviceCardComponent</i> ; caso ' <i>onWriteDatapoint</i> non emette se <code>isExecuting</code> è true': verifica che <code>emitSpy</code> non viene invocato.	<code>emitSpy</code> non viene invocato.	S
TU-524	Contesto <i>DeviceCardComponent</i> ; caso ' <i>onWriteDatapoint</i> non emette se non c'è valore selezionabile': verifica che <code>emitSpy</code> non viene invocato.	<code>emitSpy</code> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-525	Contesto <i>DeviceCardComponent</i> ; caso 'onWriteDatapoint emette payload completo quando c'è un valore' : verifica che emitSpy viene invocato con { roomId: 'room-1', deviceId: 'device-1', datapointId: 'dp-writable', value: 'ON', } e emitSpy viene invocato 1 volta.	emitSpy viene invocato con { roomId: 'room-1', deviceId: 'device-1', datapointId: 'dp-writable', value: 'ON', }; emitSpy viene invocato 1 volta.	S
TU-526	Contesto <i>DeviceCardComponent</i> ; caso 'espone label non vuote per tipo e datapoint' : verifica che component. <i>getTypeLabel</i> (DeviceType.LIGHT).length è maggiore di 0 e component. <i>getDatapointLabel</i> (writableDatapoint).length è maggiore di 0.	component. <i>getTypeLabel</i> (DeviceType.LIGHT).length è maggiore di 0; component. <i>getDatapointLabel</i> (writableDatapoint).length è maggiore di 0.	S
TU-527	Contesto <i>EndpointTableComponent</i> ; caso 'creates component and loads writable endpoints' : verifica che component è valorizzato e deviceApiMock. <i>getWritableEndpointRows</i> viene invocato 1 volta.	component è valorizzato; deviceApiMock. <i>getWritableEndpointRows</i> viene invocato 1 volta; deviceApiMock. <i>getCurrentValuePointsByDeviceIds</i> viene invocato 1 volta.	S
TU-528	Contesto <i>EndpointTableComponent</i> ; caso 'renders mapped endpoint label for known sfeType' : verifica che il valore di component. <i>getEndpointLabel</i> (mappedRow) è 'Comando cambio modalità HVAC'.	il valore di component. <i>getEndpointLabel</i> (mappedRow) è 'Comando cambio modalità HVAC'.	S
TU-529	Contesto <i>EndpointTableComponent</i> ; caso 'falls back to a runtime humanized label when sfeType is unknown' : verifica che il valore di component. <i>getEndpointLabel</i> (unknownRow) è 'Comando future feature'.	il valore di component. <i>getEndpointLabel</i> (unknownRow) è 'Comando future feature'.	S
TU-530	Contesto <i>EndpointTableComponent</i> ; caso 'returns semantic current value by matching Cmd endpoint to State datapoint' : verifica che il valore di component. <i>getCurrentValue</i> (mappedRow) è 'Heat'.	il valore di component. <i>getCurrentValue</i> (mappedRow) è 'Heat'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-531	Contesto <i>EndpointTableComponent</i> ; caso 'returns fallback when current datapoint value is unavailable': verifica che il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'.	il valore di <code>component.getCurrentValue(unknownRow)</code> è '-'.	S
TU-532	Contesto <i>RoomDetailComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-533	Contesto <i>RoomDetailComponent</i> ; caso 'mostra il titolo panoramica della pagina': verifica che heading contiene 'Panoramica allarmi e dispositivi'.	heading contiene 'Panoramica allarmi e dispositivi'.	S
TU-534	Contesto <i>RoomDetailComponent</i> ; caso 'renderizza la tabella endpoint embedded': verifica che endpointTable è valorizzato.	endpointTable è valorizzato.	S
TU-535	Contesto <i>SidebarComponent</i> ; caso 'mostra un link per ogni navItem con indirizzi corretti': verifica che il valore di <code>links.length</code> è 3 e <code>links[0].textContent</code> contiene 'test1'.	il valore di <code>links.length</code> è 3; <code>links[0].textContent</code> contiene 'test1'; il valore di <code>links[0].getAttribute('href')</code> è '/path1'; <code>links[1].textContent</code> contiene 'test2'.	S
TU-536	Contesto <i>SidebarComponent</i> ; caso 'mostra messaggio se navItems è vuoto': verifica che content contiene 'Menu di navigazione vuoto!'.	content contiene 'Menu di navigazione vuoto!'.	S
TU-537	Contesto <i>SidebarComponent</i> ; caso 'non mostra controllo di chiusura sidebar': verifica che button è null.	button è null.	S
TU-538	Contesto <i>SidebarComponent</i> ; caso 'emette navItemSelected con la route selezionata': verifica che spy viene invocato 1 volta e spy viene invocato con 'alarms/alarm-management'.	spy viene invocato 1 volta; spy viene invocato con 'alarms/alarm-management'.	S
TU-539	Contesto <i>TopbarComponent</i> ; caso 'dovrebbe essere creato': verifica che component è valorizzato.	component è valorizzato.	S
TU-540	Contesto <i>TopbarComponent</i> ; caso 'dovrebbe emettere hamburgerClicked quando viene premuto il menu': verifica che spy viene invocato.	spy viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-541	Contesto <i>TopbarComponent</i> ; caso 'dovrebbe emettere <code>profileClicked</code> quando viene premuto il nome utente': verifica che <code>spy</code> viene invocato.	<code>spy</code> viene invocato.	S
TU-542	Contesto <i>TopbarComponent</i> ; caso 'emette <code>profileClicked</code> anche quando utente non <code>admin</code> ': verifica che <code>spy</code> viene invocato.	<code>spy</code> viene invocato.	S
TU-543	Contesto <i>TopbarComponent</i> ; caso 'evidenzia in giallo il profilo quando attivo': verifica che il valore di <code>profileButton.classList.contains('bg-amber-300')</code> è true.	il valore di <code>profileButton.classList.contains('bg-amber-300')</code> è true.	S
TU-544	Contesto <i>TopbarComponent</i> ; caso 'mostra avviso <code>MyVimar</code> quando richiesto': verifica che <code>warning</code> è valorizzato e <code>warning.textContent?.toLowerCase()</code> contiene 'account da associare a <code>myvimar</code> '.	<code>warning</code> è valorizzato; <code>warning.textContent?.toLowerCase()</code> contiene 'account da associare a <code>myvimar</code> '.	S
TU-545	Contesto <i>MyVimarAccountStatusComponent</i> ; caso 'should <code>create</code> ': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-546	Contesto <i>MyVimarAccountStatusComponent</i> ; caso 'mostra pulsante di collegamento quando account non collegato': verifica che <code>text</code> contiene 'Nessun account <code>MyVimar</code> collegato.' e <code>text</code> contiene 'Collega account <code>MyVimar</code> '.	<code>text</code> contiene 'Nessun account <code>MyVimar</code> collegato.'; <code>text</code> contiene 'Collega account <code>MyVimar</code> '.	S
TU-547	Contesto <i>MyVimarAccountStatusComponent</i> ; caso 'mostra email e pulsante di rimozione quando account collegato': verifica che <code>text</code> contiene 'admin@example.com' e <code>text</code> contiene 'Rimuovi account'.	<code>text</code> contiene 'admin@example.com'; <code>text</code> contiene 'Rimuovi account'.	S
TU-548	Contesto <i>MyVimarAccountStatusComponent</i> ; caso 'emette <code>linkClicked</code> quando viene premuto il pulsante collega': verifica che <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-549	Contesto <i>MyVimarAccountStatusComponent</i> ; caso <code>'emette unlinkClicked'</code> quando viene premuto il pulsante <code>rimuovi'</code> : verifica che <code>emitSpy</code> viene invocato 1 volta.	<code>emitSpy</code> viene invocato 1 volta.	S
TU-550	Contesto <i>MyVimarPageComponent</i> ; caso <code>'should create'</code> : verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-551	Contesto <i>MyVimarPageComponent</i> ; caso <code>'carica lo stato account in ngOnInit'</code> : verifica che <code>serviceStub.getLinkedAccount</code> viene invocato 1 volta.	<code>serviceStub.getLinkedAccount</code> viene invocato 1 volta.	S
TU-552	Contesto <i>MyVimarPageComponent</i> ; caso <code>'delegates onLinkAccount al service'</code> : verifica che <code>serviceStub.initiateOAuth</code> viene invocato 1 volta.	<code>serviceStub.initiateOAuth</code> viene invocato 1 volta.	S
TU-553	Contesto <i>MyVimarPageComponent</i> ; caso <code>'onUnlinkAccount richiama il service e attiva refresh'</code> : verifica che <code>serviceStub.unlinkAccount</code> viene invocato 1 volta e <code>serviceStub.getLinkedAccount</code> viene invocato 2 volte.	<code>serviceStub.unlinkAccount</code> viene invocato 1 volta; <code>serviceStub.getLinkedAccount</code> viene invocato 2 volte.	S
TU-554	Contesto <i>MyVimarPageComponent</i> ; caso <code>'imposta errore quando unlinkAccount fallisce'</code> : verifica che <code>component.error</code> contiene <code>'Errore durante la rimozione'</code> e il valore di <code>component.isLoading</code> è <code>false</code> .	<code>component.error</code> contiene <code>'Errore durante la rimozione'</code> ; il valore di <code>component.isLoading</code> è <code>false</code> .	S
TU-555	Contesto <i>NotificationBadgeComponent</i> ; caso <code>'should create'</code> : verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-556	Contesto <i>NotificationBadgeComponent</i> ; caso <code>'nasconde il badge quando count è 0'</code> : verifica che <code>badge</code> è <code>null</code> .	<code>badge</code> è <code>null</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-557	Contesto <i>NotificationBadgeComponent</i> - mostra il badge quando count; caso '0 con testo e label accessibile': verifica che con count positivo il badge viene mostrato con testo numerico e attributo aria-label coerente.	il badge è visibile, mostra testo '3' e aria-label '3 notifiche non lette'.	S
TU-558	Contesto <i>NotificationBadgeComponent</i> ; caso 'normalizza valori negativi nascondendo il badge': verifica che badge è null.	badge è null.	S
TU-559	Contesto <i>NotificationBadgeComponent</i> ; caso 'limita la visualizzazione a 99+ per count elevati': verifica che il valore di <code>badge.textContent.trim()</code> è '99+' e il valore di <code>badge.getAttribute('aria-label')</code> è '120 notifiche non lette'.	il valore di <code>badge.textContent.trim()</code> è '99+'; il valore di <code>badge.getAttribute('aria-label')</code> è '120 notifiche non lette'.	S
TU-560	Contesto <i>NotificationItemComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-561	Contesto <i>NotificationItemComponent</i> ; caso 'renderizza titolo e tempo relativo': verifica che <code>title.textContent</code> contiene 'Rilevato movimento in soggiorno' e il valore di <code>time.textContent.trim()</code> è '1m fa'.	<code>title.textContent</code> contiene 'Rilevato movimento in soggiorno'; il valore di <code>time.textContent.trim()</code> è '1m fa'.	S
TU-562	Contesto <i>NotificationItemComponent</i> ; caso 'espone il timestamp originale in datetime per accessibilità e machine-readability': verifica che il valore di <code>time.getAttribute('datetime')</code> è <code>notification.sentAt</code> .	il valore di <code>time.getAttribute('datetime')</code> è <code>notification.sentAt</code> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-563	Contesto <i>NotificationItemComponent</i> ; caso 'aggiorna il contenuto quando cambia input notification': verifica che title.textContent contiene 'Temperatura elevata in cucina' e il valore di time.textContent.trim() è '40s fa'.	title.textContent contiene 'Temperatura elevata in cucina'; il valore di time.textContent.trim() è '40s fa'.	S
TU-564	Contesto <i>NotificationItemComponent</i> ; caso 'emette <i>removeClicked</i> quando si clicca la X': verifica che emitSpy viene invocato con 'n-1' e emitSpy viene invocato 1 volta.	emitSpy viene invocato con 'n-1'; emitSpy viene invocato 1 volta.	S
TU-565	Contesto <i>NotificationPageComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-566	Contesto <i>NotificationPageComponent</i> ; caso 'renderizza uno stato vuoto accessibile quando non ci sono notifiche': verifica che emptyState è valorizzato e emptyState.textContent contiene 'Nessuna notifica disponibile'.	emptyState è valorizzato; emptyState.textContent contiene 'Nessuna notifica disponibile'; items ha lunghezza 0.	S
TU-567	Contesto <i>NotificationPageComponent</i> ; caso 'renderizza la lista notifiche e passa i dati ai figli': verifica che summary.textContent contiene '2 notifiche non lette' e items ha lunghezza 2.	summary.textContent contiene '2 notifiche non lette'; items ha lunghezza 2; items[0].componentInstance.notification() coincide con notificationA; items[1].componentInstance.notification() coincide con notificationB.	S
TU-568	Contesto <i>NotificationPageComponent</i> ; caso 'usa il singolare nel riepilogo quando unreadCount è 1': verifica che summary.textContent contiene '1 notifica non letta'.	summary.textContent contiene '1 notifica non letta'.	S
TU-569	Contesto <i>NotificationPageComponent</i> ; caso 'aggiorna il DOM quando vm\$ emette un nuovo snapshot coerente': verifica che items ha lunghezza 1 e items ha lunghezza 2.	items ha lunghezza 1; items ha lunghezza 2.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-570	Contesto <i>NotificationPageComponent</i> ; caso 'invoca <i>clearAllNotifications</i> quando si clicca Cancellate tutte': verifica che <i>notificationServiceStub.clearAllNotifications</i> viene invocato con [<i>notificationA, notificationB, </i>].	<i>notificationServiceStub.clearAllNotifications</i> viene invocato con [<i>notificationA, notificationB, </i>].	S
TU-571	Contesto <i>NotificationPageComponent</i> ; caso 'evidenzia leggermente la notifica richiesta da query param focus': verifica che il valore di <i>items[0].componentInstance.isHighlighted()</i> è false e il valore di <i>items[1].componentInstance.isHighlighted()</i> è true.	il valore di <i>items[0].componentInstance.isHighlighted()</i> è false; il valore di <i>items[1].componentInstance.isHighlighted()</i> è true.	S
TU-572	Contesto <i>NotificationTopbarPanelComponent</i> ; caso 'mostrato vuoto quando non ci sono notifiche': verifica che <i>empty</i> è valorizzato e <i>empty.textContent?.toLowerCase()</i> contiene 'nessuna notifica'.	<i>empty</i> è valorizzato; <i>empty.textContent?.toLowerCase()</i> contiene 'nessuna notifica'.	S
TU-573	Contesto <i>NotificationTopbarPanelComponent</i> ; caso 'emette <i>viewAllClicked</i> quando si clicca su Vedi tutte': verifica che <i>emitSpy</i> viene invocato 1 volta.	<i>emitSpy</i> viene invocato 1 volta.	S
TU-574	Contesto <i>NotificationTopbarPanelComponent</i> ; caso 'emette <i>removeClicked</i> quando si clicca sulla X di una notifica': verifica che <i>removeSpy</i> viene invocato con 'n-1' e <i>removeSpy</i> viene invocato 1 volta.	<i>removeSpy</i> viene invocato con 'n-1'; <i>removeSpy</i> viene invocato 1 volta.	S
TU-575	Contesto <i>NotificationTopbarPanelComponent</i> ; caso 'emette <i>notificationSelected</i> quando si clicca su un item notifica in preview': verifica che <i>selectedSpy</i> viene invocato con 'n-1' e <i>selectedSpy</i> viene invocato 1 volta.	<i>selectedSpy</i> viene invocato con 'n-1'; <i>selectedSpy</i> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-576	Contesto <i>NotificationTopbarPanelComponent</i> ; caso 'emette <i>clearAllClicked</i> quando si clicca <i>Cancella tutte</i> ': verifica che <i>clearAllSpy</i> viene invocato 1 volta.	<i>clearAllSpy</i> viene invocato 1 volta.	S
TU-577	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga a first-access quando isFirstAccess è true': verifica che il valore di component.isLoading è false e component.errorType è null.	il valore di component.isLoading è false; component.errorType è null; routerMock.navigate viene invocato con ['/auth/first-access'].	S
TU-578	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga a dashboard quando isFirstAccess è false': verifica che routerMock.navigate viene invocato con ['/dashboard'].	routerMock.navigate viene invocato con ['/dashboard'].	S
TU-579	Contesto <i>AuthBaseComponent</i> ; caso 'handleSuccess naviga su returnUrl quando presente': verifica che routerMock.navigateByUrl viene invocato con '/vimar-link' e routerMock.navigate non viene invocato con ['/dashboard'].	routerMock.navigateByUrl viene invocato con '/vimar-link'; routerMock.navigate non viene invocato con ['/dashboard'].	S
TU-580	Contesto <i>AuthBaseComponent</i> ; caso 'handleError imposta USERNAME_OR_PASSWORD_WRONG su 400 e 401': verifica che il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG.	il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG.	S
TU-581	Contesto <i>AuthBaseComponent</i> ; caso 'handleError imposta NEW_PASSWORD_EQUALS_TEMP su 409': verifica che il valore di component.errorType è AuthErrorType.NEW_PASSWORD_EQUALS_TEMP.	il valore di component.errorType è AuthErrorType.NEW_PASSWORD_EQUALS_TEMP.	S
TU-582	Contesto <i>AuthBaseComponent</i> ; caso 'handleError usa fallback USERNAME_OR_PASSWORD_WRONG per errori non Http': verifica che il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG e il valore di component.isLoading è false.	il valore di component.errorType è AuthErrorType.USERNAME_OR_PASSWORD_WRONG; il valore di component.isLoading è false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-583	Contesto <i>AuthBaseComponent</i> ; caso 'espone il messaggio per credenziali errate': verifica che il valore di <code>component.loginErrorMessage</code> è 'Utente non trovato: username o password errati.'	il valore di <code>component.loginErrorMessage</code> è 'Utente non trovato: username o password errati.'	S
TU-584	Contesto <i>FirstAccessComponent</i> ; caso 'crea il componente': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-585	Contesto <i>FirstAccessComponent</i> ; caso 'renderizza il contenuto informativo del primo accesso': verifica che <code>content</code> contiene 'Primo accesso' e <code>content</code> contiene 'Imposta la tua password personale'.	<code>content</code> contiene 'Primo accesso'; <code>content</code> contiene 'Imposta la tua password personale'; <code>content</code> contiene 'password temporanea'; <code>content</code> contiene 'Conferma e continua'.	S
TU-586	Contesto <i>FirstAccessComponent</i> ; caso 'imposta errore di validazione quando la nuova password coincide con la temporanea': verifica che il valore di <code>component.firstAccessForm.errors?.['sameAsTemporaryPassword']</code> è true e <code>authServiceMock.setFirstAccessPassword</code> non viene invocato.	il valore di <code>component.firstAccessForm.errors?.['sameAsTemporaryPassword']</code> è true; <code>authServiceMock.setFirstAccessPassword</code> non viene invocato.	S
TU-587	Contesto <i>FirstAccessComponent</i> ; caso 'invia <code>setFirstAccessPassword</code> e naviga su dashboard in caso di successo': verifica che <code>authServiceMock.setFirstAccessPassword</code> viene invocato con 'mrossi' e <code>routerMock.navigate</code> viene invocato con ['/dashboard'].	<code>authServiceMock.setFirstAccessPassword</code> viene invocato con 'mrossi'; <code>routerMock.navigate</code> viene invocato con ['/dashboard']; il valore di <code>component.isLoading</code> è false.	S
TU-588	Contesto <i>FirstAccessComponent</i> ; caso 'imposta errore USERNAME_OR_TEMP_PASSWORD_WRONG in caso di errore API': verifica che il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_TEMP_PASSWORD_WRONG</code> e il valore di <code>component.isLoading</code> è false.	il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_TEMP_PASSWORD_WRONG</code> ; il valore di <code>component.isLoading</code> è false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-589	Contesto <i>FirstAccessComponent</i> ; caso 'renderizza un messaggio quando username o password temporanea sono errati': verifica che content contiene 'Username o password temporanea errati.'	content contiene 'Username o password temporanea errati.'	S
TU-590	Contesto <i>FirstAccessComponent</i> ; caso 'marca il form touched quando submit su form invalido': verifica che markAllAsTouchedSpy viene invocato e authServiceMock.setFirstAccessPassword non viene invocato.	markAllAsTouchedSpy viene invocato; authServiceMock.setFirstAccessPassword non viene invocato.	S
TU-591	Contesto <i>FirstAccessComponent</i> ; caso 'non invia la richiesta quando la nuova password è troppo corta': verifica che component.firstAccessForm.controls.newPassword.errors?.['minlength'] è valorizzato e authServiceMock.setFirstAccessPassword non viene invocato.	component.firstAccessForm.controls.newPassword.errors?.['minlength'] è valorizzato; authServiceMock.setFirstAccessPassword non viene invocato.	S
TU-592	Contesto <i>FirstAccessComponent</i> ; caso 'renderizza il messaggio di validazione quando la nuova password coincide con la temporanea': verifica che content contiene 'La nuova password deve essere diversa da quella temporanea.'	content contiene 'La nuova password deve essere diversa da quella temporanea.'	S
TU-593	Contesto <i>LoginComponent</i> ; caso 'crea il componente': verifica che component è valorizzato.	component è valorizzato.	S
TU-594	Contesto <i>LoginComponent</i> ; caso 'non invoca login se il form non rispetta i validator': verifica che markAllAsTouchedSpy viene invocato e authServiceMock.login non viene invocato.	markAllAsTouchedSpy viene invocato; authServiceMock.login non viene invocato.	S
TU-595	Contesto <i>LoginComponent</i> ; caso 'invoca login e naviga su dashboard se autenticazione ok': verifica che authServiceMock.login viene invocato con 'mrossi' e routerMock.navigate viene invocato con ['/dashboard'].	authServiceMock.login viene invocato con 'mrossi'; routerMock.navigate viene invocato con ['/dashboard']; component.errorType è null; il valore di component.isLoading è false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-596	Contesto <i>LoginComponent</i> ; caso 'invoca login anche con valori presenti solo nel DOM (autofill browser)': verifica che <code>authServiceMock.login</code> viene invocato con 'mrossi'.	<code>authServiceMock.login</code> viene invocato con 'mrossi'.	S
TU-597	Contesto <i>LoginComponent</i> ; caso 'invoca login e naviga su <code>returnUrl</code> se presente': verifica che <code>authServiceMock.login</code> viene invocato con 'mrossi' e <code>routerMock.navigateByUrl</code> viene invocato con '/vimar-link'.	<code>authServiceMock.login</code> viene invocato con 'mrossi'; <code>routerMock.navigateByUrl</code> viene invocato con '/vimar-link'.	S
TU-598	Contesto <i>LoginComponent</i> ; caso 'imposta errore <code>USERNAME_OR_PASSWORD_WRONG</code> su 401': verifica che il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</code> e il valore di <code>component.isLoading</code> è false.	il valore di <code>component.errorType</code> è <code>AuthErrorType.USERNAME_OR_PASSWORD_WRONG</code> ; il valore di <code>component.isLoading</code> è false.	S
TU-599	Contesto <i>LoginComponent</i> ; caso 'espone il messaggio già al primo tentativo fallito': verifica che il valore di <code>component.isLoading</code> è false e il valore di <code>component.loginErrorMessage</code> è 'Utente non trovato: username o password errati.'.	il valore di <code>component.isLoading</code> è false; il valore di <code>component.loginErrorMessage</code> è 'Utente non trovato: username o password errati.'.	S
TU-600	Contesto <i>LoginComponent</i> ; caso 'renderizza il testo errore quando <code>errorType</code> è valorizzato': verifica che <code>content</code> contiene 'Utente non trovato: username o password errati.'.	<code>content</code> contiene 'Utente non trovato: username o password errati.'.	S
TU-601	Contesto <i>CreateUserForm</i> ; caso 'should create': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-602	Contesto <i>CreateUserForm</i> ; caso 'submit emette <code>formSubmit</code> quando il form è valido': verifica che <code>submitSpy</code> viene invocato con { name: 'Mario', surname: 'Rossi', username: 'mrossi', } e <code>submitSpy</code> viene invocato 1 volta.	<code>submitSpy</code> viene invocato con { name: 'Mario', surname: 'Rossi', username: 'mrossi', }; <code>submitSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-603	Contesto <i>CreateUserForm</i> ; caso 'submit non emette e marca touched quando il form è invalido' : verifica che il valore di <code>component.form.invalid</code> è true e il valore di <code>component.form.controls['name'].touched</code> è true.	il valore di <code>component.form.invalid</code> è true; il valore di <code>component.form.controls['name'].touched</code> è true; il valore di <code>component.form.controls['surname'].touched</code> è true; il valore di <code>component.form.controls['username'].touched</code> è true.	S
TU-604	Contesto <i>CreateUserForm</i> ; caso 'applica validazioni required e minLength' : verifica che il valore di <code>component.form.controls['name'].hasError('required')</code> è true e il valore di <code>component.form.controls['surname'].hasError('minlength')</code> è true.	il valore di <code>component.form.controls['name'].hasError('required')</code> è true; il valore di <code>component.form.controls['surname'].hasError('minlength')</code> è true; il valore di <code>component.form.controls['username'].hasError('minlength')</code> è true.	S
TU-605	Contesto <i>CreateUserForm</i> ; caso 'reset pulisce il form' : verifica che <code>component.form.value</code> coincide con <code>{ name: "", surname: "", username: "", }</code> .	<code>component.form.value</code> coincide con <code>{ name: "", surname: "", username: "", }</code> .	S
TU-606	Contesto <i>CreateUserForm</i> ; caso 'requestClose emette closeRequest' : verifica che <code>closeSpy</code> viene invocato 1 volta.	<code>closeSpy</code> viene invocato 1 volta.	S
TU-607	Contesto <i>CreateUserForm</i> ; caso 'requestOpen emette openRequest' : verifica che <code>openSpy</code> viene invocato 1 volta.	<code>openSpy</code> viene invocato 1 volta.	S
TU-608	Contesto <i>UserCreatedDialogComponent</i> ; caso 'should create' : verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-609	Contesto <i>UserCreatedDialogComponent</i> ; caso 'mostra nome utente e password temporanea' : verifica che <code>textContent</code> contiene 'Mario' e <code>textContent</code> contiene 'mrossi'.	<code>textContent</code> contiene 'Mario'; <code>textContent</code> contiene 'mrossi'; <code>textContent</code> contiene 'TempPass123'.	S
TU-610	Contesto <i>UserCreatedDialogComponent</i> ; caso 'closeDialog emette evento closed' : verifica che <code>closedSpy</code> viene invocato 1 volta.	<code>closedSpy</code> viene invocato 1 volta.	S
TU-611	Contesto <i>UserCreatedDialogComponent</i> ; caso 'click sul bottone chiudi invoca closeDialog' : verifica che <code>closeSpy</code> viene invocato 1 volta.	<code>closeSpy</code> viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-612	Contesto <i>UserListComponent</i> ; caso 'should create' : verifica che component è valorizzato.	component è valorizzato.	S
TU-613	Contesto <i>UserListComponent</i> ; caso 'onDelete apre il dialog di conferma' : verifica che component.pendingDeleteUser() coincide con user.	component.pendingDeleteUser() coincide con user.	S
TU-614	Contesto <i>UserListComponent</i> ; caso 'onDeleteConfirm emette deleteUser con id numerico' : verifica che emitSpy viene invocato con 1 e emitSpy viene invocato 1 volta.	emitSpy viene invocato con 1; emitSpy viene invocato 1 volta; component.pendingDeleteUser() è null.	S
TU-615	Contesto <i>UserListComponent</i> ; caso 'onDeleteCancel non emette deleteUser e chiude il dialog' : verifica che emitSpy non viene invocato e component.pendingDeleteUser() è null.	emitSpy non viene invocato; component.pendingDeleteUser() è null.	S
TU-616	Contesto <i>UserManagementPage</i> ; caso 'should create' : verifica che component è valorizzato e <i>getUsersMock</i> viene invocato 1 volta.	component è valorizzato; <i>getUsersMock</i> viene invocato 1 volta.	S
TU-617	Contesto <i>UserManagementPage</i> ; caso 'mostra di default solo la tabella e non il form OSS' : verifica che il valore di component.isCreateFormOpen() è false e nativeElement.textContent contiene 'Elenco operatori sanitari'.	il valore di component.isCreateFormOpen() è false; nativeElement.textContent contiene 'Elenco operatori sanitari'; panel non è null; panel?.className contiene 'max-h-0'.	S
TU-618	Contesto <i>UserManagementPage</i> ; caso 'apre e chiude il form OSS con i pulsanti dedicati' : verifica che il valore di component.isCreateFormOpen() è true e il valore di component.isCreateFormOpen() è false.	il valore di component.isCreateFormOpen() è true; il valore di component.isCreateFormOpen() è false.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-619	Contesto <code>UserManagementPage</code> ; caso <code>'onFormSubmit</code> in successo imposta <code>createdResponse</code> e attiva <code>refresh lista</code> ': verifica che <code>createUserMock</code> viene invocato con <code>{ name: 'Mario', surname: 'Rossi', username: 'mrossi', }</code> e il valore di <code>component.createdResponse()?.tempPassword</code> è <code>'TempPass123'</code> .	<code>createUserMock</code> viene invocato con <code>{ name: 'Mario', surname: 'Rossi', username: 'mrossi', }</code> ; il valore di <code>component.createdResponse()?.tempPassword</code> è <code>'TempPass123'</code> ; <code>component.createdUser()</code> coincide con <code>{ name: 'Mario', surname: 'Rossi', username: 'mrossi', }</code> ; <code>component.formError()</code> è null.	S
TU-620	Contesto <code>UserManagementPage</code> ; caso <code>'onFormSubmit</code> con 409 imposta <code>USERNAME_ALREADY_IN_USE</code> ': verifica che il valore di <code>component.formError()</code> è <code>UserManagementErrorType.USERNAME_ALREADY_IN_USE</code> .	il valore di <code>component.formError()</code> è <code>UserManagementErrorType.USERNAME_ALREADY_IN_USE</code> .	S
TU-621	Contesto <code>UserManagementPage</code> ; caso <code>'onFormSubmit</code> con errore generico imposta <code>OTHER_ERROR</code> ': verifica che il valore di <code>component.formError()</code> è <code>UserManagementErrorType.OTHER_ERROR</code> .	il valore di <code>component.formError()</code> è <code>UserManagementErrorType.OTHER_ERROR</code> .	S
TU-622	Contesto <code>UserManagementPage</code> ; caso <code>'onUserDeleted</code> delega al service e attiva <code>refresh lista</code> ': verifica che <code>deleteUserMock</code> viene invocato con 1 e <code>getUsersMock</code> viene invocato 2 volte.	<code>deleteUserMock</code> viene invocato con 1; <code>getUsersMock</code> viene invocato 2 volte.	S
TU-623	Contesto <code>UserManagementPage</code> ; caso <code>'onDialogClosed</code> azzera <code>createdResponse</code> ': verifica che <code>component.createdResponse()</code> è null e <code>component.createdUser()</code> è null.	<code>component.createdResponse()</code> è null; <code>component.createdUser()</code> è null.	S
TU-624	Contesto <code>UserManagementPage</code> ; caso <code>'users\$</code> ritorna <code>[]</code> se il caricamento utenti fallisce': verifica che <code>users</code> coincide con <code>[]</code> .	<code>users</code> coincide con <code>[]</code> .	S
TU-625	Contesto <code>AssignOperatorDialogComponent</code> ; caso <code>'should create</code> ': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-626	Contesto <i>AssignOperatorDialogComponent</i> ; caso 'dovrebbe resettare userId in <i>ngOnInit</i> ': verifica che compo- nent.form.controls.userId.value è null.	component.form.controls.userId.value è null.	S
TU-627	Contesto <i>AssignOperatorDialogComponent</i> ; caso 'non dovrebbe emettere submitted se il form è invalido': verifica che il valore di component.form.invalid è true e il valore di compo- nent.form.controls.userId.touched è true.	il valore di component.form.invalid è true; il valore di compo- nent.form.controls.userId.touched è true; submittedSpy non viene invocato.	S
TU-628	Contesto <i>AssignOperatorDialogComponent</i> ; caso 'dovrebbe emettere submitted con userId quando il form è valido': verifica che submittedSpy viene invocato con { userId: 1 } e submittedSpy viene invocato 1 volta.	submittedSpy viene invocato con { userId: 1 }; submittedSpy viene invocato 1 volta.	S
TU-629	Contesto <i>AssignOperatorDialogComponent</i> ; caso 'dovrebbe emettere cancelled quando <i>onCancel</i> viene chiamato': verifica che cancelledSpy viene invocato 1 volta.	cancelledSpy viene invocato 1 volta.	S
TU-630	Contesto <i>AssignWardDialogComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-631	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe resettare plantId a null in <i>ngOnInit</i> ': verifica che compo- nent.form.controls.plantId.value è null.	component.form.controls.plantId.value è null.	S
TU-632	Contesto <i>AssignWardDialogComponent</i> ; caso 'non dovrebbe emettere submitted se il form è invalido': verifica che il valore di component.form.invalid è true e il valore di compo- nent.form.controls.plantId.touched è true.	il valore di component.form.invalid è true; il valore di compo- nent.form.controls.plantId.touched è true; submittedSpy non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-633	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe emettere submitted con plantId quando il form è valido': verifica che submittedSpy viene invocato con { plantId : '101' } e submittedSpy viene invocato 1 volta.	submittedSpy viene invocato con { plantId : '101' }; submittedSpy viene invocato 1 volta.	S
TU-634	Contesto <i>AssignWardDialogComponent</i> ; caso 'dovrebbe emettere cancelled quando <i>onCancel</i> viene chiamato': verifica che cancelledSpy viene invocato 1 volta.	cancelledSpy viene invocato 1 volta.	S
TU-635	Contesto <i>WardCardComponent</i> ; caso 'should create': verifica che component è valorizzato.	component è valorizzato.	S
TU-636	Contesto <i>WardCardComponent</i> ; caso 'toggleExpanded dovrebbe alternare lo stato espanso': verifica che il valore di component.isExpanded() è false e il valore di component.isExpanded() è true.	il valore di component.isExpanded() è false; il valore di component.isExpanded() è true.	S
TU-637	Contesto <i>WardCardComponent</i> ; caso 'onEditWardClick dovrebbe emettere editWard con il ward corrente': verifica che editWardSpy viene invocato con ward e editWardSpy viene invocato 1 volta.	editWardSpy viene invocato con ward ; editWardSpy viene invocato 1 volta.	S
TU-638	Contesto <i>WardCardComponent</i> ; caso 'onDeleteWardClick dovrebbe emettere deleteWard con ward.id ': verifica che deleteWardSpy viene invocato con 1 e deleteWardSpy viene invocato 1 volta.	deleteWardSpy viene invocato con 1; deleteWardSpy viene invocato 1 volta.	S
TU-639	Contesto <i>WardCardComponent</i> ; caso 'onAssignOperatorClick dovrebbe emettere assignOperator con ward.id ': verifica che assignOperatorSpy viene invocato con 1 e assignOperatorSpy viene invocato 1 volta.	assignOperatorSpy viene invocato con 1; assignOperatorSpy viene invocato 1 volta.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-640	Contesto <i>WardCardComponent</i> ; caso ' <i>onRemoveOperatorClick</i> dovrebbe emettere payload completo': verifica che <i>removeOperatorSpy</i> viene invocato con { wardId: 1, userId: 2, } e <i>removeOperatorSpy</i> viene invocato 1 volta.	<i>removeOperatorSpy</i> viene invocato con { wardId: 1, userId: 2, }; <i>removeOperatorSpy</i> viene invocato 1 volta.	S
TU-641	Contesto <i>WardCardComponent</i> ; caso ' <i>onAssignPlantClick</i> dovrebbe emettere <i>assignPlant</i> con ward.id': verifica che <i>assignPlantSpy</i> viene invocato con 1 e <i>assignPlantSpy</i> viene invocato 1 volta.	<i>assignPlantSpy</i> viene invocato con 1; <i>assignPlantSpy</i> viene invocato 1 volta.	S
TU-642	Contesto <i>WardCardComponent</i> ; caso ' <i>onRemovePlantClick</i> dovrebbe emettere payload completo': verifica che <i>removePlantSpy</i> viene invocato con { wardId: 1, plantId: '102', } e <i>removePlantSpy</i> viene invocato 1 volta.	<i>removePlantSpy</i> viene invocato con { wardId: 1, plantId: '102', }; <i>removePlantSpy</i> viene invocato 1 volta.	S
TU-643	Contesto <i>WardCardComponent</i> ; caso ' <i>non dovrebbe emettere eventi legati al ward quando ward è null</i> ': verifica che <i>editWardSpy</i> non viene invocato e <i>deleteWardSpy</i> non viene invocato.	<i>editWardSpy</i> non viene invocato; <i>deleteWardSpy</i> non viene invocato; <i>assignOperatorSpy</i> non viene invocato; <i>removeOperatorSpy</i> non viene invocato.	S
TU-644	Contesto <i>WardFormDialogComponent</i> ; caso ' <i>should create</i> ': verifica che component è valorizzato.	component è valorizzato.	S
TU-645	Contesto <i>WardFormDialogComponent</i> ; caso ' <i>dovrebbe precompilare il nome in edit mode</i> ': verifica che il valore di <i>component.isEditMode()</i> è true e il valore di <i>component.form.controls.name.value</i> è 'Cardiologia'.	il valore di <i>component.isEditMode()</i> è true; il valore di <i>component.form.controls.name.value</i> è 'Cardiologia'.	S
TU-646	Contesto <i>WardFormDialogComponent</i> ; caso ' <i>non dovrebbe emettere submitted se nome vuoto</i> ': verifica che il valore di <i>component.form.invalid</i> è true e il valore di <i>component.form.controls.name.touched</i> è true.	il valore di <i>component.form.invalid</i> è true; il valore di <i>component.form.controls.name.touched</i> è true; <i>submittedSpy</i> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-647	Contesto <i>WardFormDialogComponent</i> ; caso 'non dovrebbe emettere submitted se il nome supera 100 caratteri': verifica che il valore di <code>component.form.invalid</code> è true e <code>component.form.controls.name.errors?.['maxlength']</code> è valorizzato.	il valore di <code>component.form.invalid</code> è true; <code>component.form.controls.name.errors?.['maxlength']</code> è valorizzato; <code>submittedSpy</code> non viene invocato.	S
TU-648	Contesto <i>WardFormDialogComponent</i> ; caso 'dovrebbe emettere submitted con nome troncato se form valido': verifica che <code>submittedSpy</code> viene invocato con <code>{ name: 'Psicopazzia' }</code> e <code>submittedSpy</code> viene invocato 1 volta.	<code>submittedSpy</code> viene invocato con <code>{ name: 'Psicopazzia' }</code> ; <code>submittedSpy</code> viene invocato 1 volta.	S
TU-649	Contesto <i>WardFormDialogComponent</i> ; caso 'dovrebbe emettere cancelled quando <i>onCancel</i> viene chiamato': verifica che <code>cancelledSpy</code> viene invocato 1 volta.	<code>cancelledSpy</code> viene invocato 1 volta.	S
TU-650	Contesto <i>WardManagementPageComponent</i> ; caso 'should create': verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-651	Contesto <i>WardManagementPageComponent</i> ; caso 'ngOnInit dovrebbe chiamare <i>loadWards</i> e sincronizzare <i>snapshots/error</i> ': verifica che <code>storeStub.loadWards</code> viene invocato 1 volta e <code>component.wardsSnapshot()</code> coincide con <code>[]</code> .	<code>storeStub.loadWards</code> viene invocato 1 volta; <code>component.wardsSnapshot()</code> coincide con <code>[]</code> ; <code>component.snackbarMessage()</code> è null; <code>component.wardsSnapshot()</code> ha lunghezza 2.	S
TU-652	Contesto <i>WardManagementPageComponent</i> ; caso 'quando wards diventa vuoto resetta selezioni e step mobile': verifica che <code>component.selectedWardId()</code> è null e <code>component.selectedApartmentId()</code> è null.	<code>component.selectedWardId()</code> è null; <code>component.selectedApartmentId()</code> è null; il valore di <code>component.mobileStep()</code> è 'wards'.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-653	Contesto <i>WardManagementPage-Component</i> ; caso 'selectedApartment si riallinea al ward selezionato' : verifica che <code>component.selectedApartment()</code> coincide con <code>{ id: '101', name: 'App. 101' }</code> e <code>component.selectedApartment()</code> è null.	<code>component.selectedApartment()</code> coincide con <code>{ id: '101', name: 'App. 101' }</code> ; <code>component.selectedApartment()</code> è null.	S
TU-654	Contesto <i>WardManagementPage-Component</i> ; caso 'ngOnDestroy dovrebbe interrompere aggiornamenti da stream' : verifica che il valore di <code>component.snackbarMessage()</code> è 'Errore iniziale' .	il valore di <code>component.snackbarMessage()</code> è 'Errore iniziale' .	S
TU-655	Contesto <i>WardManagementPage-Component</i> ; caso 'dismissSnackbar dovrebbe azzerare il messaggio' : verifica che <code>component.snackbarMessage()</code> è null.	<code>component.snackbarMessage()</code> è null.	S
TU-656	Contesto <i>WardManagementPage-Component</i> ; caso 'onCreateWard dovrebbe aprire dialog in create mode' : verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.wardDialogMode()</code> è 'create' .	il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.wardDialogMode()</code> è 'create' .	S
TU-657	Contesto <i>WardManagementPage-Component</i> ; caso 'onCreateWardSubmit dovrebbe delegare a store e chiudere dialog' : verifica che <code>storeStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> e <code>storeStub.createWard</code> viene invocato 1 volta.	<code>storeStub.createWard</code> viene invocato con <code>{ name: 'Oncologia' }</code> ; <code>storeStub.createWard</code> viene invocato 1 volta; il valore di <code>component.wardDialogMode()</code> è 'closed' .	S
TU-658	Contesto <i>WardManagementPage-Component</i> ; caso 'onEditWard e onEditWardSubmit dovrebbero aggiornare il reparto selezionato' : verifica che il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> e il valore di <code>component.wardDialogMode()</code> è 'edit' .	il valore di <code>component.selectedWardId()</code> è <code>wardA.id</code> ; il valore di <code>component.wardDialogMode()</code> è 'edit' ; <code>storeStub.updateWard</code> viene invocato con 1; il valore di <code>component.wardDialogMode()</code> è 'closed' .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-659	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onEditWardSubmit non dovrebbe chiamare store se selectedWard è null'</i> : verifica che <i>storeStub.updateWard</i> non viene invocato.	<i>storeStub.updateWard</i> non viene invocato.	S
TU-660	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onCloseWardDialog dovrebbe chiudere dialog e pulire selezione'</i> : verifica che il valore di <i>component.selectedWardId()</i> è <i>wardA.id</i> e il valore di <i>component.wardDialogMode()</i> è <i>'closed'</i> .	il valore di <i>component.selectedWardId()</i> è <i>wardA.id</i> ; il valore di <i>component.wardDialogMode()</i> è <i>'closed'</i> .	S
TU-661	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onCloseAssignOperatorDialog e onCloseAssignPlantDialog ripuliscono stato di caricamento e lookup'</i> : verifica che <i>component.operatorWardId()</i> è <i>null</i> e <i>component.availableOperatorsFromFetch()</i> è <i>null</i> .	<i>component.operatorWardId()</i> è <i>null</i> ; <i>component.availableOperatorsFromFetch()</i> è <i>null</i> ; il valore di <i>component.isLoadingAvailableOperators()</i> è <i>false</i> ; <i>component.plantWardId()</i> è <i>null</i> .	S
TU-662	Contesto <i>WardManagementPage-Component</i> ; caso <i>'selectWard dovrebbe aggiornare ward attivo e step mobile'</i> : verifica che il valore di <i>component.selectedWardId()</i> è <i>2</i> e il valore di <i>component.mobileStep()</i> è <i>'apartments'</i> .	il valore di <i>component.selectedWardId()</i> è <i>2</i> ; il valore di <i>component.mobileStep()</i> è <i>'apartments'</i> .	S
TU-663	Contesto <i>WardManagementPage-Component</i> ; caso <i>'selectWard sul ward già selezionato mantiene la selezione dell'appartamento'</i> : verifica che il valore di <i>component.selectedWardId()</i> è <i>wardA.id</i> e il valore di <i>component.selectedApartmentId()</i> è <i>'101'</i> .	il valore di <i>component.selectedWardId()</i> è <i>wardA.id</i> ; il valore di <i>component.selectedApartmentId()</i> è <i>'101'</i> ; il valore di <i>component.mobileStep()</i> è <i>'apartments'</i> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-664	Contesto <i>WardManagementPage-Component</i> ; caso <i>'selectApartment e step helpers dovrebbero aggiornare lo stato ui'</i> : verifica che il valore di <code>component.selectedApartmentId()</code> è '101' e il valore di <code>component.mobileStep()</code> è 'wards'.	il valore di <code>component.selectedApartmentId()</code> è '101'; il valore di <code>component.mobileStep()</code> è 'wards'; il valore di <code>component.mobileStep()</code> è 'apartments'; il valore di <code>component.mobileStep()</code> è 'operators'.	S
TU-665	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onEditSelectedWard non dovrebbe fare nulla senza ward selezionato'</i> : verifica che <code>component.selectedWardId()</code> è null e il valore di <code>component.wardDialogMode()</code> è 'closed'.	<code>component.selectedWardId()</code> è null; il valore di <code>component.wardDialogMode()</code> è 'closed'.	S
TU-666	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onAssignOperatorSubmit non dovrebbe chiamare store senza wardId'</i> : verifica che <code>storeStub.assignOperator</code> non viene invocato.	<code>storeStub.assignOperator</code> non viene invocato.	S
TU-667	Contesto <i>WardManagementPage-Component</i> ; caso <i>'flow plant dovrebbe impostare wardId, submit e chiudere'</i> : verifica che il valore di <code>component.plantWardId()</code> è 2 e <code>storeStub.getAvailablePlantsForWard</code> viene invocato con 2.	il valore di <code>component.plantWardId()</code> è 2; <code>storeStub.getAvailablePlantsForWard</code> viene invocato con 2; <code>component.availablePlants()</code> coincide con <code>[{ id: '103', name: 'App. 103' }]</code> ; <code>storeStub.assignPlant</code> viene invocato con 2.	S
TU-668	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onAssignPlantSubmit non dovrebbe chiamare store senza wardId'</i> : verifica che <code>storeStub.assignPlant</code> non viene invocato.	<code>storeStub.assignPlant</code> non viene invocato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-669	Contesto <i>WardManagementPage-Component</i> ; caso <i>'confirmState + onConfirmDialogConfirmed</i> dovrebbero gestire i 3 rami': verifica che <i>component.confirmState()</i> coincide con <i>{ kind: 'delete-ward', wardId: 1 }</i> e <i>storeStub.deleteWard</i> viene invocato con 1.	<i>component.confirmState()</i> coincide con <i>{ kind: 'delete-ward', wardId: 1 }</i> ; <i>storeStub.deleteWard</i> viene invocato con 1; <i>component.confirmState()</i> è null; <i>component.confirmState()</i> coincide con <i>{ kind: 'remove-operator', wardId: 1, userId: 2, }</i> .	S
TU-670	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onConfirmDialogConfirmed</i> non dovrebbe fare nulla con stato nullo': verifica che <i>storeStub.deleteWard</i> non viene invocato e <i>storeStub.removeOperator</i> non viene invocato.	<i>storeStub.deleteWard</i> non viene invocato; <i>storeStub.removeOperator</i> non viene invocato; <i>storeStub.removePlant</i> non viene invocato.	S
TU-671	Contesto <i>WardManagementPage-Component</i> ; caso <i>'onConfirmDialogCancelled</i> dovrebbe pulire <i>confirmState</i> ': verifica che <i>component.confirmState()</i> è null.	<i>component.confirmState()</i> è null.	S
TU-672	Contesto <i>WardManagementPage-Component</i> ; caso <i>'getConfirmMessage</i> dovrebbe restituire il testo corretto per ogni stato': verifica che il valore di <i>component.getConfirmMessage()</i> è <i>'Confermi questa operazione?'</i> e il valore di <i>component.getConfirmLabel()</i> è <i>'Conferma'</i> .	il valore di <i>component.getConfirmMessage()</i> è <i>'Confermi questa operazione?'</i> ; il valore di <i>component.getConfirmLabel()</i> è <i>'Conferma'</i> ; il valore di <i>component.getConfirmMessage()</i> è <i>'Confermi l\{\}eliminazione del reparto?'</i> ; il valore di <i>component.getConfirmMessage()</i> è <i>'Confermi la rimozione dell\{\}operatore dal reparto?'</i> .	S
TU-673	Contesto <i>WardManagementPage-Component</i> ; caso <i>'getOperatorDisplayName</i> dovrebbe comporre ed effettuare il trim di nome e cognome': verifica che il valore di <i>component.getOperatorDisplayName('Mario', 'Rossi')</i> è <i>'Mario Rossi'</i> e il valore di <i>component.getOperatorDisplayName('Mario', '')</i> è <i>'Mario'</i> .	il valore di <i>component.getOperatorDisplayName('Mario', 'Rossi')</i> è <i>'Mario Rossi'</i> ; il valore di <i>component.getOperatorDisplayName('Mario', '')</i> è <i>'Mario'</i> .	S

Codice	Descrizione	Esito atteso	Stato del test
TU-674	Contesto <i>WardManagementPage-Component</i> ; caso 'availablePlants dovrebbe deduplicare e escludere quelli già assegnati al ward selezionato' : verifica che <code>component.availablePlants()</code> coincide con <code>[]</code> .	<code>component.availablePlants()</code> coincide con <code>[]</code> .	S
TU-675	Contesto <i>WardManagementPage-Component</i> ; caso 'onAssignPlant non usa fallback locale se fetch ad-hoc fallisce' : verifica che <code>component.availablePlants()</code> coincide con <code>[]</code> e il valore di <code>component.isLoadingAvailablePlants()</code> è <code>false</code> .	<code>component.availablePlants()</code> coincide con <code>[]</code> ; il valore di <code>component.isLoadingAvailablePlants()</code> è <code>false</code> .	S
TU-676	Contesto <i>WardRowComponent</i> ; caso 'should create' : verifica che <code>component</code> è valorizzato.	<code>component</code> è valorizzato.	S
TU-677	Contesto <i>WardRowComponent</i> ; caso 'accetta plant nullo' : verifica che <code>component.plant()</code> è <code>null</code> .	<code>component.plant()</code> è <code>null</code> .	S
TU-678	Contesto <i>WardRowComponent</i> ; caso 'accetta plant valorizzato con id stringa' : verifica che <code>component.plant()</code> coincide con <code>{ id: '101', name: 'App' }</code> .	<code>component.plant()</code> coincide con <code>{ id: '101', name: 'App' }</code> .	S
TU-679	Contesto <i>Plot</i> ; caso 'should be defined' : verifica che un'istanza di <code>Plot</code> è valorizzata.	nuova istanza <code>Plot</code> è definita.	S
TU-680	Contesto <i>Plot</i> ; caso 'should return correct title, metric, unit, labels e series' : verifica che <code>Plot</code> restituisce le proprietà passate al costruttore.	<code>getTitle()</code> , <code>getMetric()</code> , <code>getUnit()</code> , <code>getLabels()</code> , <code>getSeries()</code> ritornano i valori corretti.	S
TU-681	Contesto <i>Plot</i> ; caso 'should return undefined suggestion when not set' : verifica che <code>getSuggestion()</code> è <code>undefined</code> prima di <code>setSuggestion</code> .	<code>getSuggestion()</code> ritorna <code>undefined</code> .	S
TU-682	Contesto <i>Plot</i> ; caso 'should set and return suggestion' : verifica che <code>setSuggestion()</code> aggiorna il valore e <code>getSuggestion()</code> lo ritorna.	dopo <code>setSuggestion()</code> , <code>getSuggestion()</code> ritorna il valore impostato.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-683	Contesto <i>ValidationPipe</i> ; caso 'should be defined': verifica che il pipe è valorizzato.	ValidationPipe è definito.	S
TU-684	Contesto <i>ValidationPipe</i> ; caso 'should return value if no metatype': verifica che il pipe passa il valore quando non c'è metatype.	transform ritorna il valore non modificato.	S
TU-685	Contesto <i>ValidationPipe</i> ; caso 'should skip validation for primitive types': verifica che il pipe skip primitive types come String.	transform ritorna il valore per tipologie primitive.	S
TU-686	Contesto <i>ValidationPipe</i> ; caso 'should pass validation for valid DTO': verifica che il pipe accetta DTO validi.	transform ritorna il DTO passato.	S
TU-687	Contesto <i>ValidationPipe</i> ; caso 'should throw error for invalid DTO': verifica che il pipe rigetta DTO con proprietà non valide.	transform lancia BadRequestException.	S
TU-688	Contesto <i>ValidationPipe</i> ; caso 'should throw error when required field is missing': verifica che il pipe rigetta DTO senza campi obbligatori.	transform lancia BadRequestException.	S
TU-689	Contesto <i>AlarmRulesService</i> ; caso 'should return all alarm rules': verifica che il servizio ritorna la lista di tutte le regole.	getAllAlarmRules non è un array vuoto.	S
TU-690	Contesto <i>AlarmRulesService</i> ; caso 'should return empty array when no rules exist': verifica che il servizio ritorna array vuoto quando non ci sono regole.	getAllAlarmRules ritorna un array vuoto.	S
TU-691	Contesto <i>AlarmRulesService</i> ; caso 'should return alarm rule by id': verifica che il servizio ritorna una regola specifica.	getAlarmRuleById(id) ritorna la regola con id corretto.	S
TU-692	Contesto <i>AlarmRulesService</i> ; caso 'should return null when rule not found': verifica che il servizio ritorna null quando regola non esiste.	getAlarmRuleById(invalid) ritorna null.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-693	Contesto <i>AlarmRulesService</i> ; caso 'should create alarm rule successfully' : verifica che il servizio crea una nuova regola.	<code>createAlarmRule()</code> aggiunge la regola al database.	S
TU-694	Contesto <i>AlarmRulesService</i> ; caso 'should update alarm rule successfully' : verifica che il servizio aggiorna una regola esistente.	<code>updateAlarmRule()</code> modifica le proprietà della regola.	S
TU-695	Contesto <i>AlarmRulesService</i> ; caso 'should delete alarm rule successfully' : verifica che il servizio cancella una regola.	<code>deleteAlarmRule()</code> rimuove la regola dal database.	S
TU-696	Contesto <i>AlarmRulesService</i> ; caso 'should emit alarm.activated event when alarm matches' : verifica che il servizio emette evento quando soglia è superata.	<code>checkAlarmRule</code> emette <code>alarm.activated</code> event.	S
TU-697	Contesto <i>AlarmEventsService</i> ; caso 'should return alarm event by id' : verifica che il servizio ritorna un evento specifico.	<code>getAlarmEventById(id)</code> ritorna l'evento con id corretto.	S
TU-698	Contesto <i>AlarmEventsService</i> ; caso 'should return all alarm events' : verifica che il servizio ritorna lista di tutti gli eventi.	<code>getAllAlarmEvents</code> non è un array vuoto.	S
TU-699	Contesto <i>AlarmEventsService</i> ; caso 'should return managed alarm events by user' : verifica che il servizio filtra gli allarmi risolti dell'utente.	<code>getManagedAlarmEventsByUserId()</code> ritorna solo allarmi con <code>resolution_time</code> .	S
TU-700	Contesto <i>AlarmEventsService</i> ; caso 'should return unmanaged alarm events by user' : verifica che il servizio filtra gli allarmi attivi dell'utente.	<code>getUnmanagedAlarmEventsByUserId()</code> ritorna solo allarmi con <code>resolution_time</code> null.	S
TU-701	Contesto <i>AlarmEventsService</i> ; caso 'should resolve alarm event and emit event' : verifica che il servizio risolve un allarme e emette evento.	<code>resolveAlarmEvent()</code> aggiorna <code>resolution_time</code> e emette <code>alarm.resolved</code> .	S
TU-702	Contesto <i>WardDomain</i> ; caso 'should be defined' : verifica che <code>Ward</code> può essere istanziato.	<code>new Ward(1, 'Neuro')</code> è definito.	S

Codice	Descrizione	Esito atteso	Stato del test
TU-703	Contesto <i>AnalyticsStrategy</i> ; caso 'plant-anomalies returns plot with correct data': verifica che la strategia di anomalie assume correttamente i dati.	execute() ritorna Plot con serie e label allineate.	S
TU-704	Contesto <i>AnalyticsStrategy</i> ; caso 'sensor-long-presence counts sessions over 30 minutes': verifica che la strategia conta sessioni prolungate > 30 min.	execute() conta solo una sessione per evento continuativo.	S
TU-705	Contesto <i>AnalyticsStrategy</i> ; caso 'ward-resolved-alarm aggregates total and resolved': verifica che la strategia aggrega allarmi totali e risolti.	execute() ritorna due serie: una totale, una risolta.	S

Tabella 16: Tabella dei test di unita

5 Cruscotto di qualità

5.1 MPC1 e MPC2 - Planned Value e Earned Value

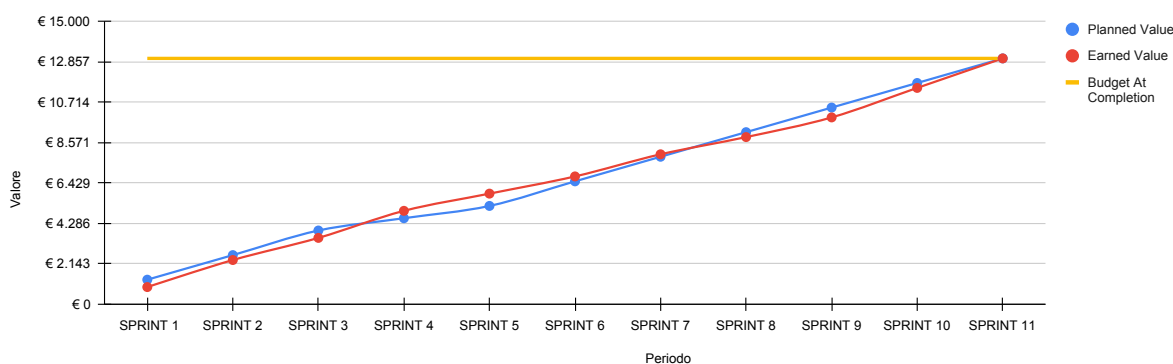


Figura 1: Andamento PV e EV in base al periodo

L'andamento del grafico denota un progressivo aumento del lavoro completato. Fino allo Sprint 3 rimane al di sotto della percentuale prevista. Grazie ad una successiva ripianificazione la percentuale di lavoro svolto è di poco al di sopra del pianificato. Durante i periodi degli Sprint 8 e 9, il valore di lavoro completato è ritornato leggermente al di sotto del preventivo, a causa di alcune sovrastime nella pianificazione. Allo Sprint 11 entrambe le curve hanno raggiunto il Budget At Completion, confermando il completamento del progetto nei tempi e nei costi previsti.

5.2 MPC3 e MPC7 - Actual Cost e Estimate To Complete

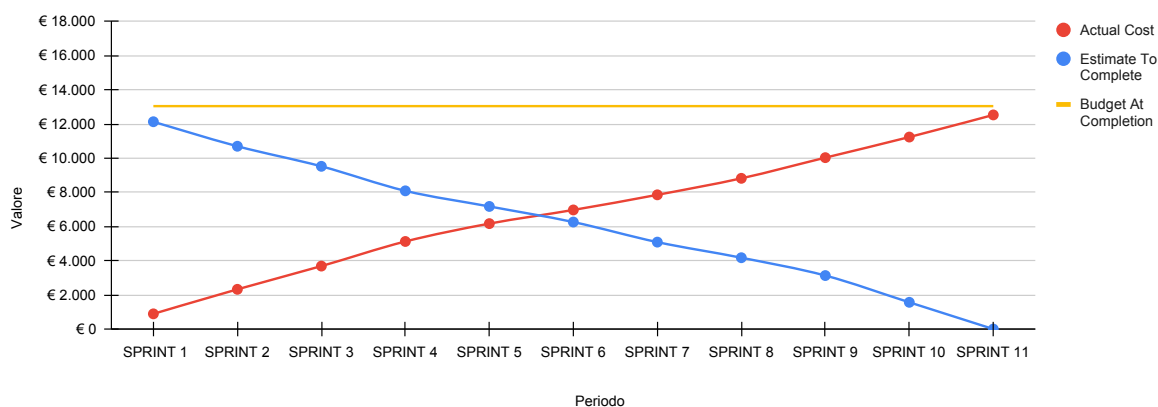


Figura 2: Andamento AC e ETC in base al periodo

Il grafico mostra che il budget impiegato è coerente con la percentuale di lavoro completata, senza evidenziare particolari anomalie nell'andamento. Allo Sprint 11 l'Estimate To Complete si è azzerato e il costo consuntivato finale si è attestato leggermente al di sotto del BAC, indicando una gestione complessivamente efficiente delle risorse economiche.

5.3 MPC4 e MPC5 - Cost Performance Index e Schedule Performance Index

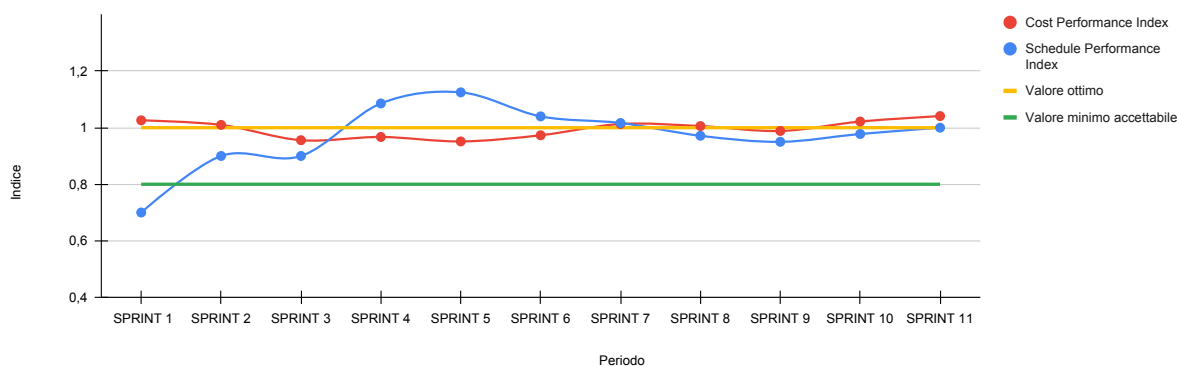


Figura 3: Andamento CPI e SPI in base al periodo

Dal grafico si desume che, nella fase iniziale, il gruppo ha gestito correttamente il budget, sebbene le attività risultassero significativamente in ritardo. Successivamente, la situazione si è stabilizzata su valori superiori alla soglia minima accettabile fino a circa il quarto sprint, momento in cui il trend si è invertito: le attività sono state completate in anticipo rispetto alla pianificazione, mentre il budget impiegato ha registrato un lieve scostamento in eccesso rispetto alle previsioni. A seguito della ripianificazione effettuata al termine dello Sprint 7, si nota nei periodi successivi una riduzione e stabilizzazione del budget speso (CPI), perdendo però qualche punto nell'SPI che rimane comunque accettabile. Allo Sprint 11 il progetto si è concluso con CPI e SPI entrambi prossimi a 1.00, confermando il rispetto dei tempi e dei costi preventivati.

5.4 MPC6 - Estimate At Completion

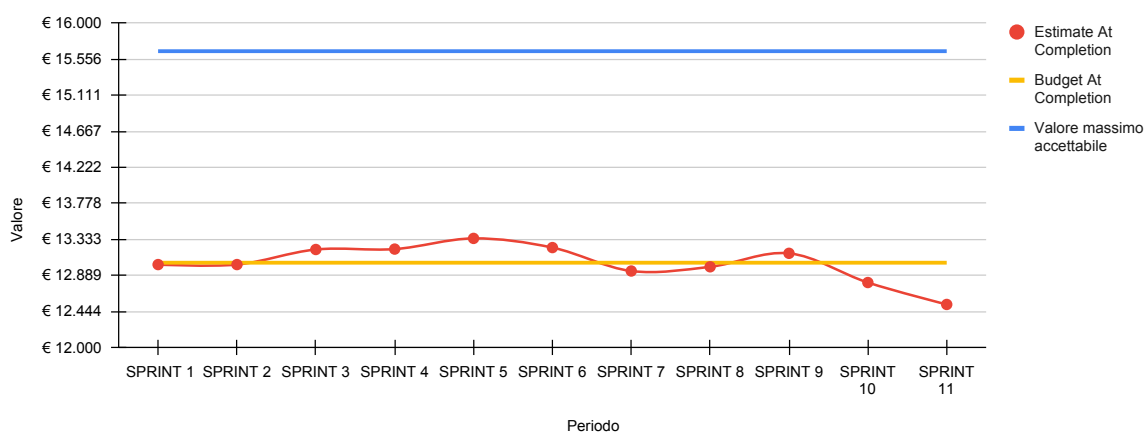


Figura 4: Andamento EAC in base al periodo

Il grafico rappresenta quanto riportato nella figura 1 (MPC1 e MPC2), ovvero una buona gestione del budget iniziale, con una previsione del costo al completamento in linea con il preventivo iniziale. Al termine dello Sprint 5, momento in cui lo sfioramento di budget ha raggiunto il valore massimo, è stata effettuata una lieve ripianificazione che ha contribuito alla riduzione dello scostamento dai costi previsti. Dallo Sprint 7 allo Sprint 8 il costo stimato al completamento è rimasto al di sotto del preventivo iniziale, subendo un leggero aumento durante lo Sprint 9. Allo Sprint 10 l'EAC si è attestato sensibilmente al di sotto del BAC, confermando una gestione del budget più efficiente, come da precedenti previsioni. Allo Sprint 11 l'EAC si è ulteriormente ridotto, chiudendo il progetto con un costo finale inferiore al preventivo iniziale e senza sfioramenti significativi.

5.5 MPC8 - Requirements Stability Index

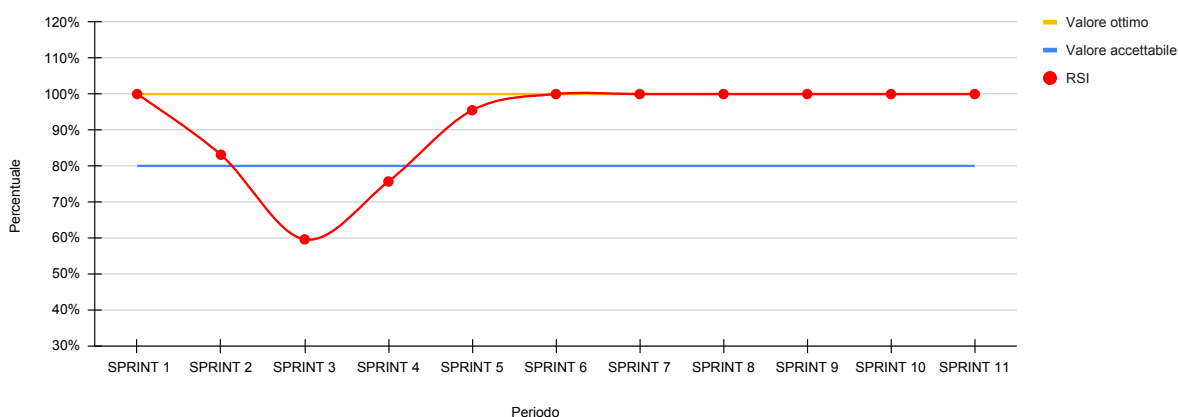


Figura 5: Andamento RSI in base al periodo

Il crollo della stabilità nello Sprint 3 è dovuto ad un iniziale errore nella raccolta dei requisiti che ha portato inevitabilmente a rivedere quasi tutti quelli individuati fino a quel momento. Dallo Sprint 4 in poi i requisiti hanno avuto una stabilità pressoché ottima. Allo Sprint 11 l'RSI ha confermato il valore ottimo del 100%, attestando che la base dei requisiti si è stabilizzata completamente senza ulteriori variazioni significative.

5.6 MPC9 - Indice di Gulpease

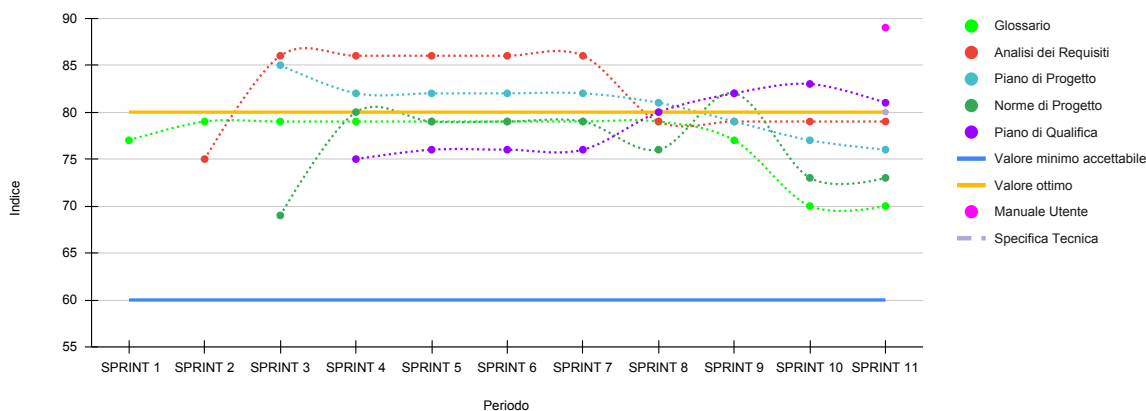


Figura 6: Andamento IG dei documenti principali, in base al periodo

Durante tutti gli sprint la leggibilità dei documenti è rimasta buona e al di sopra della soglia di accettabilità. Allo Sprint 10 si è registrato un calo dell'indice di Gulpease per il Glossario e le Norme di Progetto, dovuto all'aggiunta di contenuti correlati alla progettazione e alla codifica. Entrambi i documenti rimangono comunque al di sopra della soglia di accettabilità. Allo Sprint 11 il Piano di Qualifica e il Manuale Utente hanno registrato un incremento, raggiungendo entrambi valori superiori al valore ottimo, mentre la Specifica Tecnica ha anch'essa raggiunto un indice prossimo al valore ottimo.

5.7 MPC10 - Indice di Frammentazione

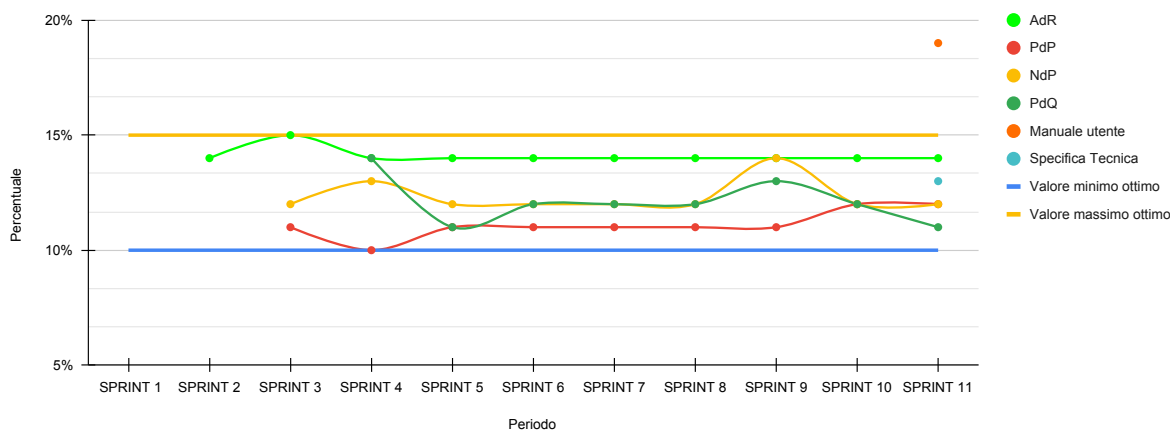


Figura 7: Andamento IF dei documenti principali, in base al periodo

Durante tutti gli sprint la struttura dei documenti è rimasta buona, al di sopra della soglia di accettabilità. Si rende evidente il fatto che il documento "Glossario" non è stato incluso nel controllo della metrica MPC10: è stata fatta questa scelta poiché quest'ultimo, essendo un documento per definizione molto frammentato (ad ogni termine viene associata una breve spiegazione), avrebbe ottenuto un valore superiore al 20%. La verifica avrebbe dunque evidenziato un problema nel Glossario quando, in realtà, non è presente. Allo Sprint 11 tutti i documenti monitorati hanno confermato valori stabili e coerenti con il range ottimo, attestandosi tra il 12% e il 15%.

5.8 MPC12 - Percentuale Metriche Soddisfatte

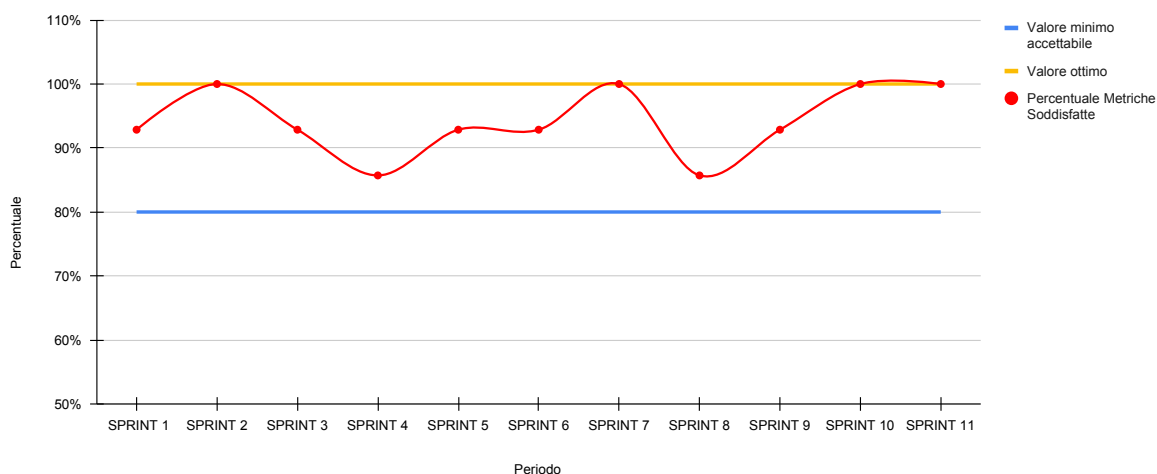


Figura 8: Andamento PMS in base al periodo

Dal grafico è evidente l'impegno del gruppo nei confronti delle metriche di qualità di processo e del loro soddisfacimento. Un lieve peggioramento è stato notato durante lo Sprint 4, soprattutto a causa dei rischi inattesi e dell'instabilità dei requisiti durante tale periodo. Dopo averne preso consapevolezza, il gruppo ha adottato opportune azioni correttive, riuscendo a migliorare progressivamente le performance negli sprint successivi. Il progetto si è concluso allo Sprint 11 con il 100% delle metriche soddisfatte, attestando il pieno raggiungimento degli obiettivi di qualità di processo.

5.9 MPC13 - Percentuale Rischi Inattesi

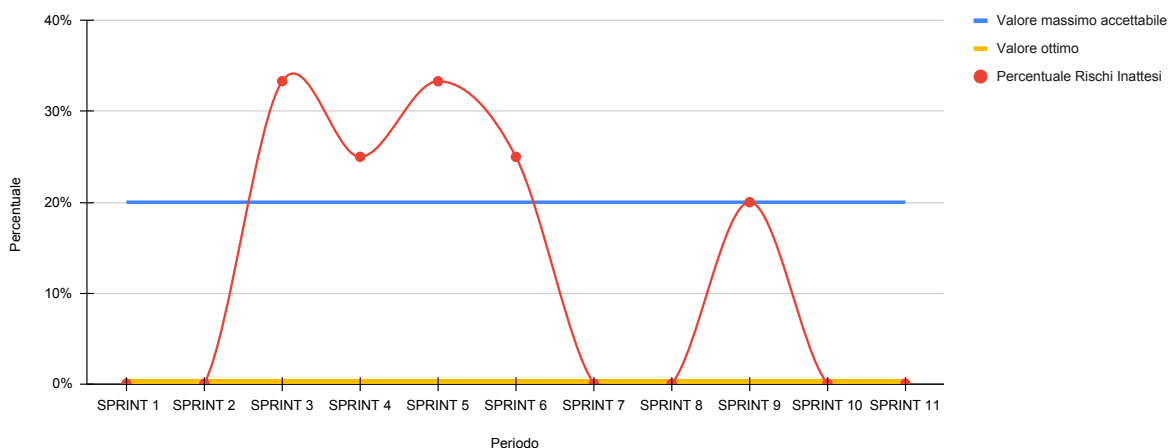


Figura 9: Andamento PRI in base al periodo

La percentuale rispetto al totale dei rischi inattesi è superiore al valore massimo accettabile nella maggior parte dei periodi. Ciò testimonia una capacità ancora non matura di previsione dei rischi da parte del gruppo. Si può notare però che agli Sprint 6, 7 e 8 è diminuita, subendo un rialzo allo Sprint 9. Allo Sprint 10 la percentuale di rischi inattesi si è mantenuta al di sotto del valore massimo accettabile, confermando il miglioramento nella capacità di previsione dei rischi da parte del gruppo. Allo Sprint 11 la percentuale

si è azzerata, raggiungendo il valore ottimo e confermando la piena maturità del gruppo nella gestione e previsione dei rischi al termine del progetto.

5.10 MPC14 - Labor Efficiency

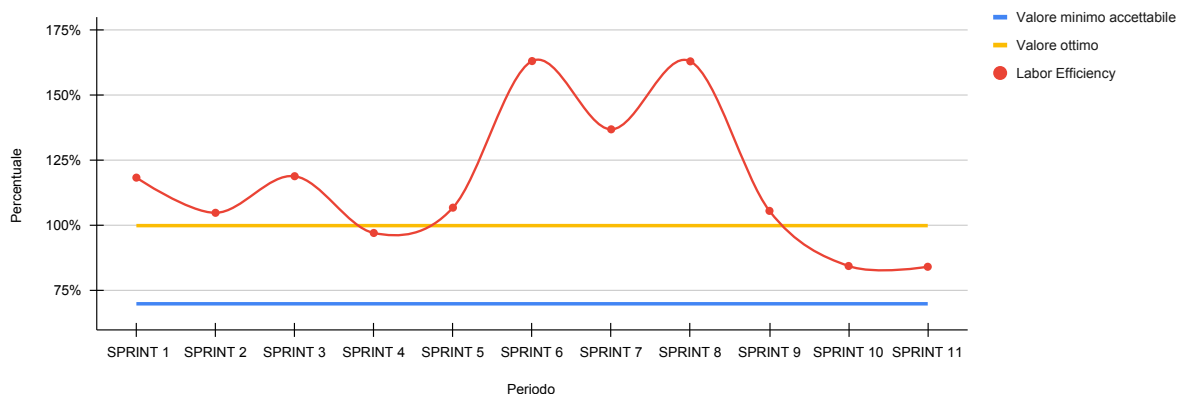


Figura 10: Andamento LE in base al periodo

Dal grafico è possibile notare, fin dal primo Sprint, un'eccellente efficienza nel lavoro svolto, con un picco importante durante lo Sprint 6 e 8. È opportuno sottolineare che tale andamento non indica un'eccessiva efficienza operativa del gruppo, bensì evidenzia che le ore preventivate sono risultate superiori rispetto a quelle effettivamente impiegate. Allo Sprint 10 si è registrato un calo della Labor Efficiency al di sotto del valore ottimo, dovuto a una stima delle ore più accurata rispetto ai periodi precedenti: le ore effettivamente impiegate si sono avvicinate a quelle preventivate, riducendo il surplus che aveva caratterizzato gli sprint precedenti. Allo Sprint 11 la Labor Efficiency si è mantenuta al di sopra della soglia minima accettabile, confermando che il gruppo ha concluso il progetto con un utilizzo delle risorse coerente con quanto preventivato.

5.11 MPD1 - Percentuale Requisiti Obbligatori Soddisfatti

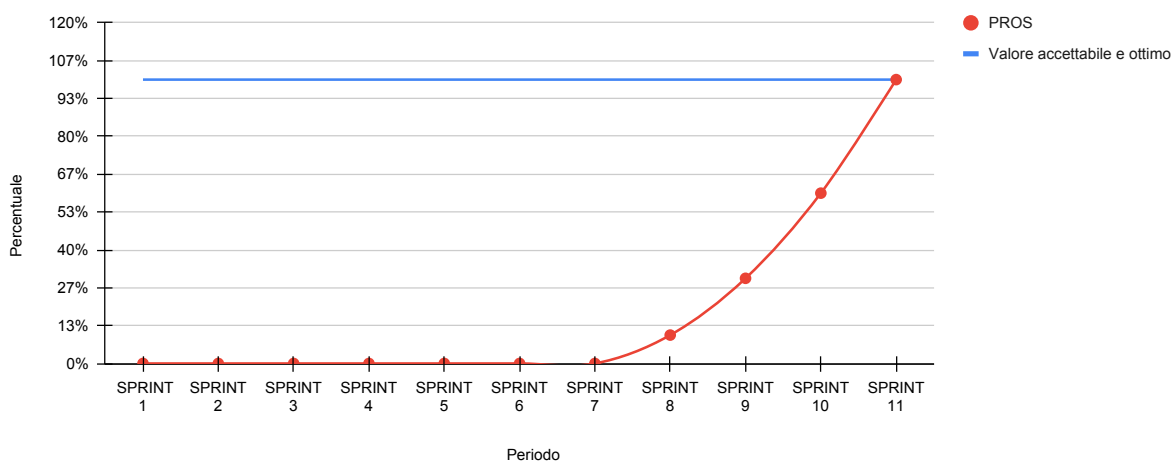


Figura 11: Andamento PROS in base al periodo

A partire dallo Sprint 8 la percentuale cresce in maniera esponenziale, raggiungendo il valore ottimo allo Sprint 11. Allo Sprint 11, la totalità dei requisiti obbligatori risulta soddisfatta, confermando il completamento con successo di tutti gli obiettivi fondamentali del progetto.

5.12 MPD2 - Percentuale Requisiti Opzionali Soddisfatti

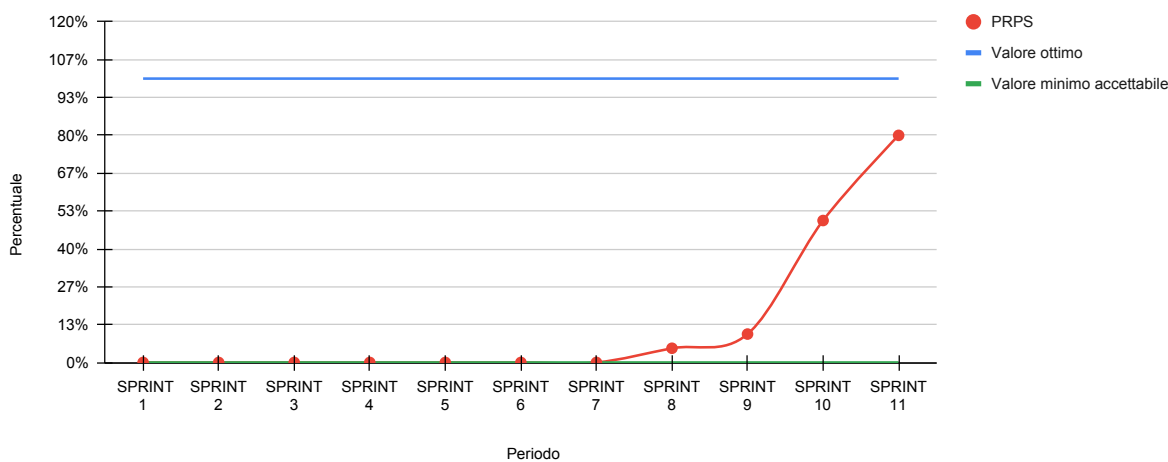


Figura 12: Andamento PRPS in base al periodo

A partire dallo Sprint 8 si osserva una crescita progressiva, tuttavia allo Sprint 11 il valore raggiunto si attesta intorno all'80%, al di sotto del valore ottimo del 100%. Ciò indica che, pur avendo dedicato attenzione ai requisiti opzionali nelle fasi finali, il gruppo ha privilegiato il soddisfacimento dei requisiti obbligatori, lasciando una quota di funzionalità opzionali non implementata al termine del progetto.

5.13 MPD3 - Percentuale Requisiti Desiderabili Soddisfatti

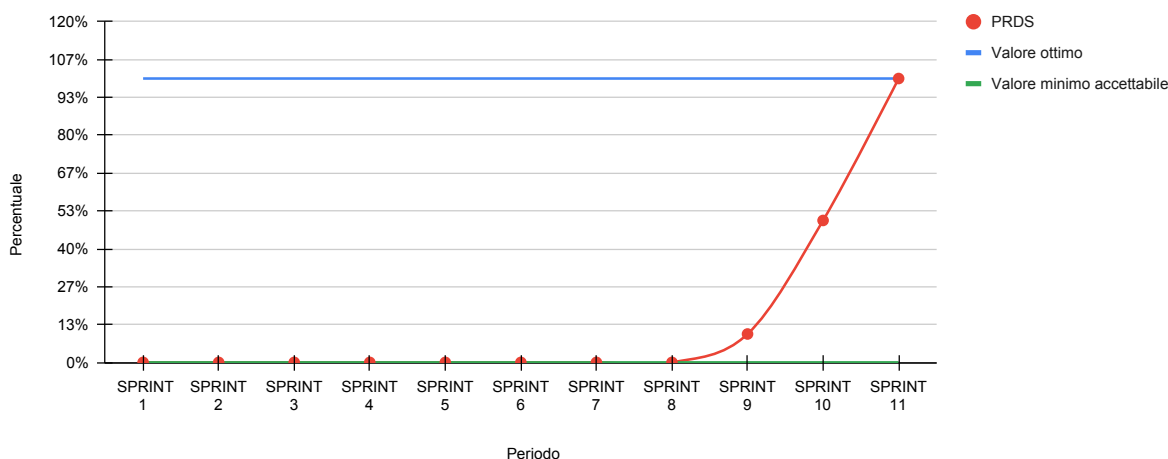


Figura 13: Andamento PRDS in base al periodo

A partire dallo Sprint 9 si osserva una crescita rapida e sostenuta, e allo Sprint 11 il valore raggiunge il valore ottimo, confermando che tutti i requisiti desiderabili sono stati soddisfatti al termine del progetto.

5.14 MPD4 - Line Coverage

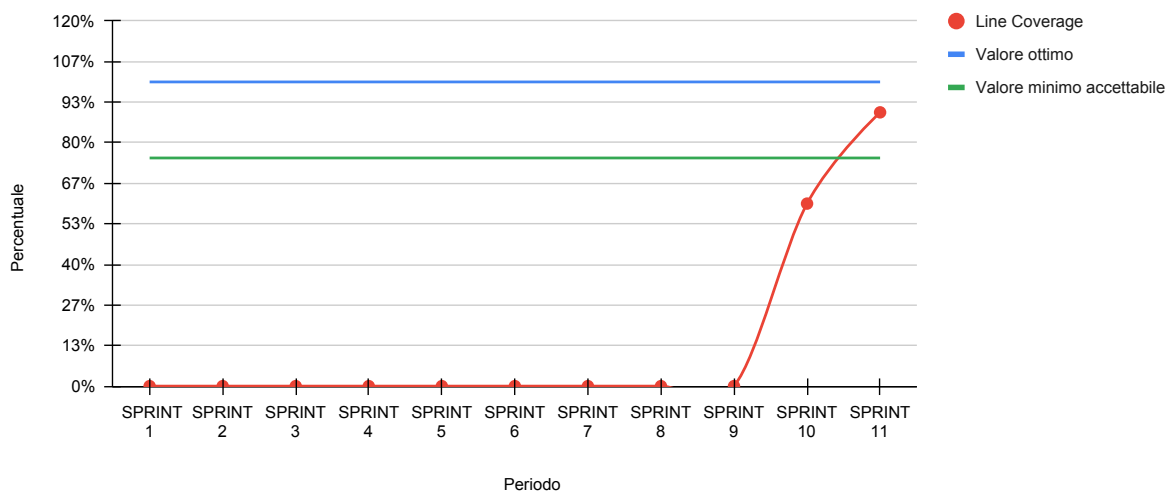


Figura 14: Andamento LC in base al periodo

A partire dallo Sprint 9 la copertura cresce rapidamente, superando il valore minimo accettabile già allo Sprint 10 e avvicinandosi al valore ottimo allo Sprint 11, pur senza raggiungerlo pienamente. Questo andamento testimonia un impegno crescente nella scrittura dei test nelle fasi conclusive del progetto.

5.15 MPD5 - Branch Coverage

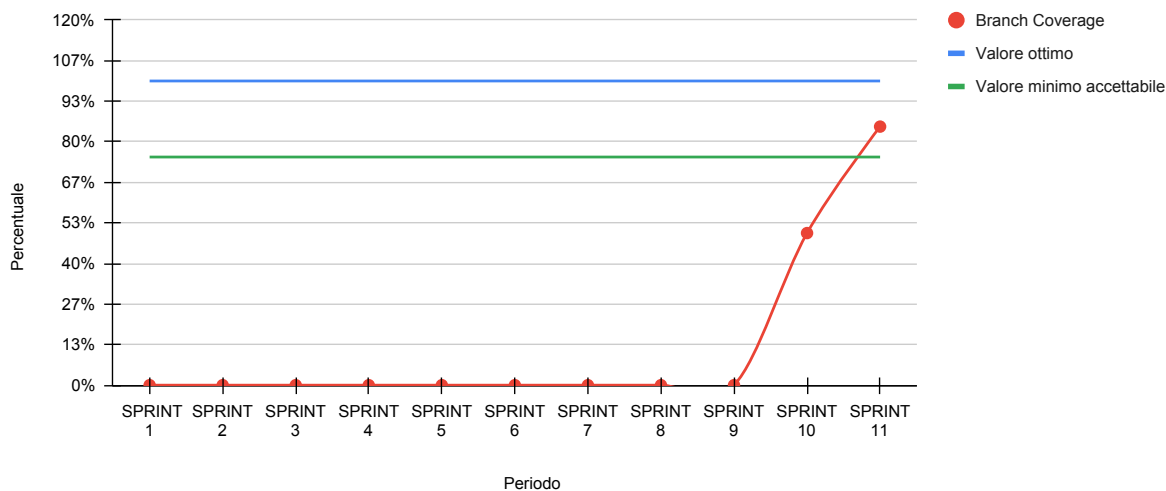


Figura 15: Andamento BC in base al periodo

A partire dallo Sprint 9 la copertura cresce in modo significativo, superando il valore minimo accettabile allo Sprint 10 e attestandosi poco al di sotto del valore ottimo allo Sprint 11. Il risultato finale indica una buona maturità nella copertura dei rami del codice, raggiunta nelle fasi conclusive del progetto.

5.16 MPD6 - Statement Coverage

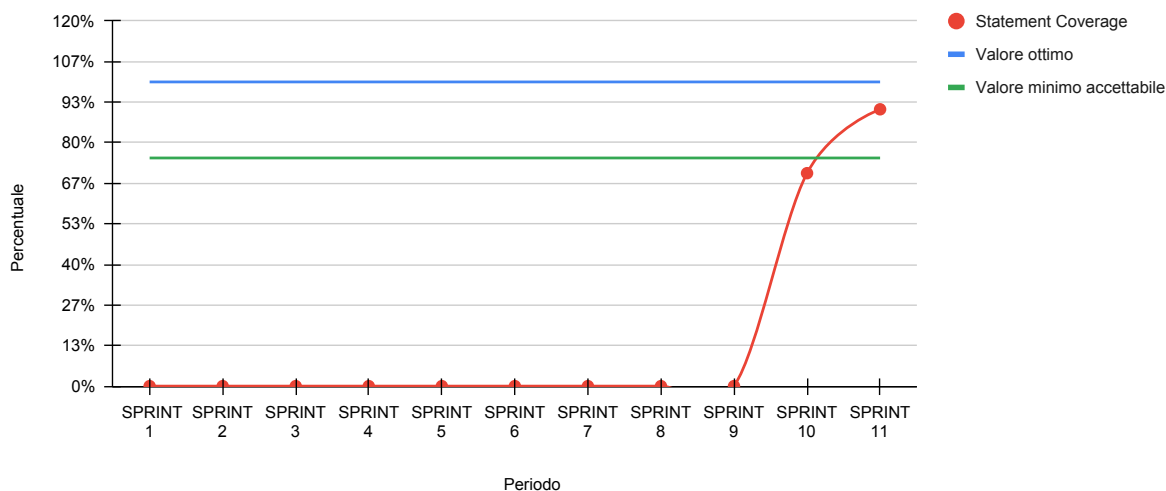


Figura 16: Andamento SC in base al periodo

A partire dallo Sprint 9 la copertura cresce rapidamente, raggiungendo il valore minimo accettabile allo Sprint 10 e avvicinandosi al valore ottimo allo Sprint 11. L'andamento complessivo è in linea con quello della *Line Coverage* e della *Branch Coverage*, confermando un approccio uniforme e sistematico alla scrittura dei test nelle fasi finali del progetto.

5.17 MPD7 - Profondità Massima di Navigazione

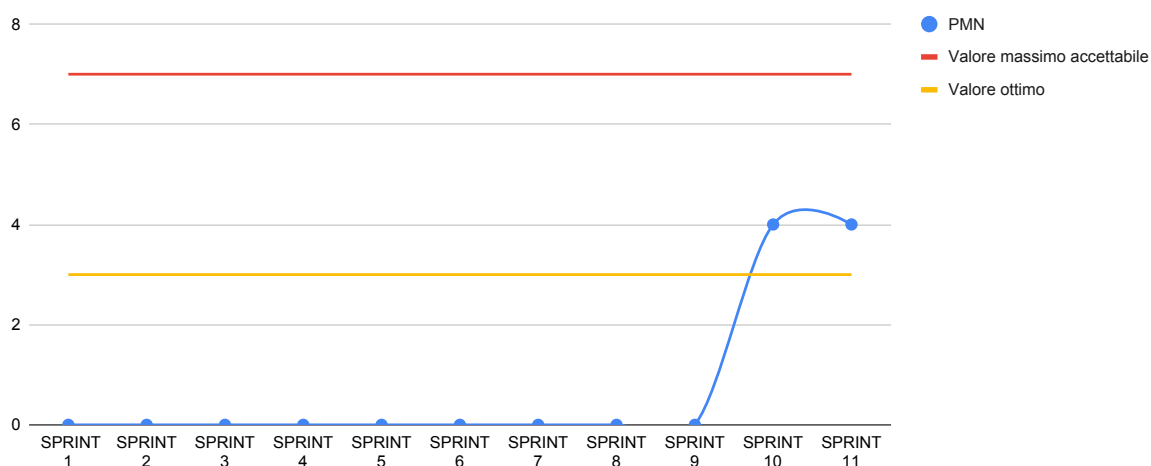


Figura 17: Andamento PMN in base al periodo

L'operazione presa come riferimento per il calcolo della seguente metrica è l'operazione di collegamento dell'account MyVimar al profilo dell'Amministratore, ovvero l'operazione con il maggior numero di passaggi all'interno del Sistema. A partire dallo Sprint 9 il valore cresce, attestandosi a 4 allo Sprint 10, leggermente al di sopra del valore ottimo di 3 ma ampiamente al di sotto del valore massimo accettabile di 7. Allo Sprint 11 il valore si stabilizza a 4, mantenendosi entro i limiti accettabili e indicando una struttura di navigazione complessivamente adeguata.

5.18 MPD8 - Tempo Medio di Risposta

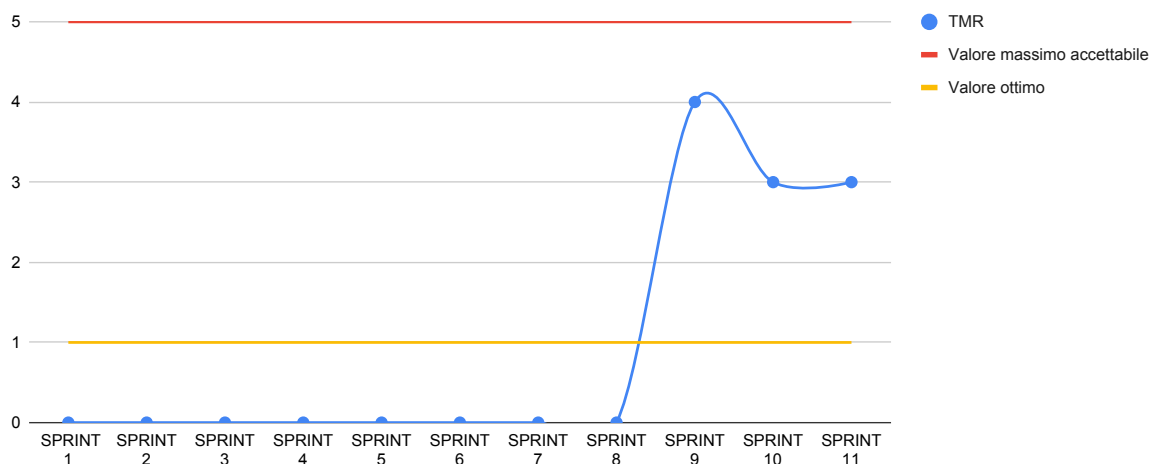


Figura 18: Andamento TMR in base al periodo

A partire dallo Sprint 8 il valore cresce rapidamente, raggiungendo un picco di 4 allo Sprint 9, al di sopra del valore ottimo di 1 secondo ma al di sotto del valore massimo accettabile di 5 secondi. Negli sprint successivi si osserva un miglioramento, con il valore che scende a 3 allo Sprint 10 e si mantiene stabile a 3 allo Sprint 11. Sebbene il valore finale risulti ancora superiore al valore ottimo, il trend discendente nelle fasi conclusive del progetto testimonia un miglioramento nelle prestazioni del Sistema.

5.19 MPD9 - Complessità Ciclomatica

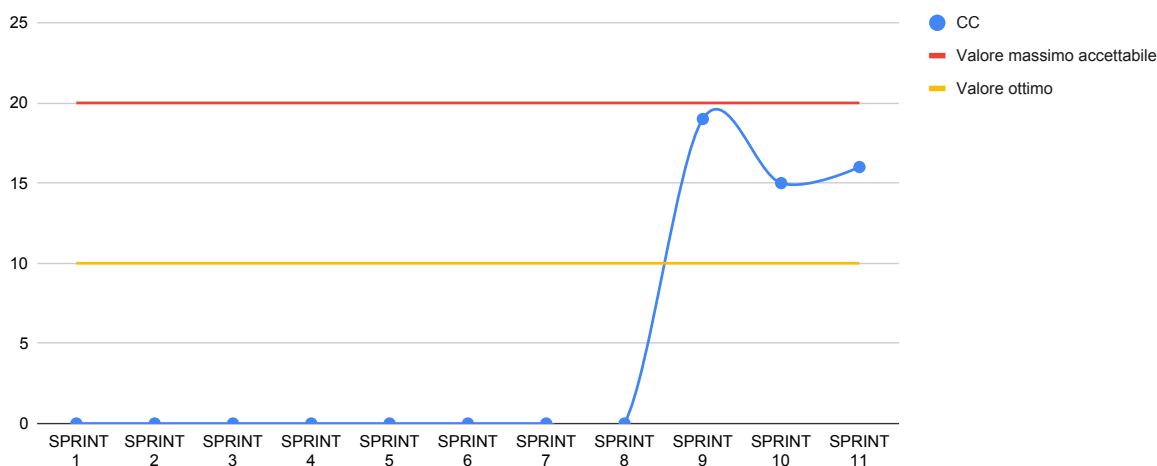


Figura 19: Andamento CC in base al periodo

A partire dallo Sprint 8 il valore cresce rapidamente, raggiungendo un picco di circa 19 allo Sprint 9, al di sopra del valore ottimo di 10 ma al di sotto del valore massimo accettabile di 20. Negli sprint successivi si osserva una riduzione, con il valore che scende a 15 allo Sprint 10 e risale leggermente a 16 allo Sprint 11, mantenendosi comunque entro il valore massimo accettabile. L'andamento complessivo indica che il codice prodotto presenta una complessità nella norma, sebbene vi sia ancora margine di miglioramento per avvicinarsi al valore ottimo.

5.20 MPD10 - Compatibilità Cross-Browser

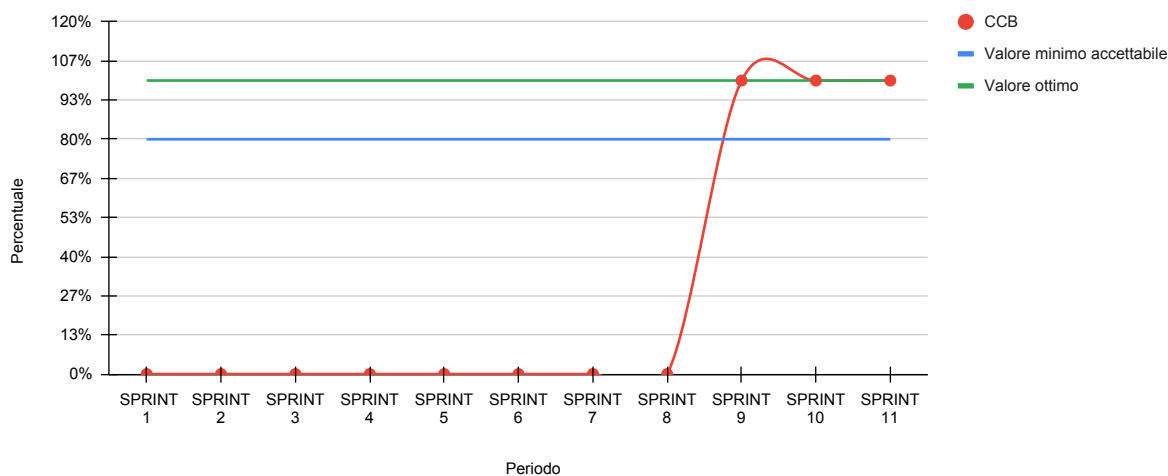


Figura 20: Andamento CCB in base al periodo

A partire dallo Sprint 8 il valore cresce rapidamente, superando sia il valore minimo accettabile che il valore ottimo già allo Sprint 9, dove si attesta intorno al 100%. Negli sprint successivi il valore si stabilizza al di sopra del valore ottimo, confermando un'ottima compatibilità del prodotto con i principali *browser* al termine del progetto.