



Specifica Tecnica

Informazioni sul documento

Versione	2.0.0
Data di Creazione	2017-02-01
Data ultima modifica	2017-04-07
Stato	Approvato
Redazione	Jordan Gottardo
Verifica	Giulia Petenazzi
Approvazione	Giovanni Prete
Uso	Esterno
Lista di distribuzione	Professor Tullio Vardanega Professor Riccardo Cardin <i>Zephyrus</i> <i>RiskApp</i>
Email di riferimento	zephyrus.swe@gmail.com

Registro delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
2.0.0	2017-04-07	Giovanni Prete	<i>Responsabile</i>	Approvazione del documento
1.1.0	2017-02-28	Giulia Petenazzi	<i>Verificatore</i>	Verifica del documento
1.0.3	2017-04-05	Jordan Gottardo	<i>Progettista</i>	Correzione post RP: modificata sottosezione "React" in sezione 2
1.0.2	2017-03-23	Jordan Gottardo	<i>Progettista</i>	Correzione post RP: rimozione sezione 7.2 "Tracciamento requisito-package"
1.0.1	2017-03-23	Jordan Gottardo	<i>Progettista</i>	Correzione post RP: rimozione sezione 7.1 "Tracciamento package-requisiti"
1.0.0	2017-02-28	Leonardo Brutesco	<i>Responsabile</i>	Approvazione del documento
0.3.0	2017-02-28	Damo Giovanni	<i>Verificatore</i>	Verifica del documento
0.2.4	2017-02-27	Giulia Petenazzi	<i>Progettista</i>	Stesura la sezione "Diagrammi di Attività"
0.2.3	2017-02-26	Giulia Petenazzi	<i>Progettista</i>	Stesura sezione "Mockup"
0.2.2	2017-02-25	Jordan Gottardo	<i>Progettista</i>	Stesura sezione "Tracciamento"
0.2.1	2017-02-28	Daniel De Gaspari	<i>Progettista</i>	Corretti gli errori grammaticali e sintattici segnalati dal verificatore nelle sezioni "Introduzione" in "Descrizione architettura" e "DeGeOP::ViewPkg" in "Componenti"
0.2.0	2017-02-24	Damo Giovanni	<i>Verificatore</i>	Verifica del documento in forma parziale
0.1.4	2017-02-24	Giovanni Prete	<i>Progettista</i>	Stesura sottosezioni "Contestualizzazione" in "Descrizione design pattern" terminando la sezione "Descrizione design pattern"
0.1.3	2017-02-23	Daniel De Gaspari	<i>Progettista</i>	Stesura sezione "Descrizione design pattern"
0.1.2	2017-02-23	Giovanni Prete	<i>Progettista</i>	Stesura sezione "Server" in "Descrizione architettura"

Versione	Data	Autore	Ruolo	Descrizione
0.1.1	2017-02-23	Marco Pasqualini	<i>Progettista</i>	Stesura sezione "Stime di fattibilità e di bisogno di risorse"
0.1.0	2017-02-22	Damo Giovanni	<i>Verificatore</i>	Verifica del documento in forma parziale
0.0.4	2017-02-22	Jordan Gottardo	<i>Progettista</i>	Stesura sezione "Componenti"
0.0.3	2017-02-21	Daniel De Gaspari	<i>Progettista</i>	Stesura sezione "Tecnologie utilizzate"
0.0.2	2017-02-21	Marco Pasqualini	<i>Progettista</i>	Stesura sezione "Introduzione" e "Descrizione Architettura"
0.0.1	2017-02-01	Daniel De Gaspari	<i>Progettista</i>	Creazione template e indice

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	2
2	Tecnologie utilizzate	3
2.1	Introduzione	3
2.2	Alexa Voice Service	4
2.3	Bing Map	4
2.4	CSS3	4
2.5	HammerJS	5
2.6	HTML5	5
2.7	JavaScript ES6	6
2.8	JSON	6
2.9	JSX	7
2.10	Node.js	7
2.11	OpenLayers	7
2.12	Open Street Map	8
2.13	React	9
2.14	ReactColor	10
2.15	React-Redux	10
2.16	React Toolbox	10
2.17	Redux	11
2.18	SVG	11
3	Descrizione architettura	12
3.1	Introduzione	12
3.2	Server	12
3.2.1	Chiamate REST	12
3.2.1.1	Customer	12
3.2.1.2	Graph	13
3.2.1.3	Asset	13
3.2.1.4	Node	13
3.2.1.5	Edge	13
3.3	Client	14
3.3.1	Design architettonicale di DeGeOP	14
4	Componenti	15
4.1	DeGeOP	15
4.1.1	Informazioni sul package	15
4.2	DeGeOP:StorePkg	16
4.2.1	Informazioni sul package	16
4.3	DeGeOP:StorePkg::StoreContentsPkg	18
4.3.1	Informazioni sul package	18
4.3.2	Classi	18

4.3.2.1	Customer	18
4.3.2.2	Options	18
4.3.2.3	StoreDeGeOP	18
4.4	DeGeOP::StorePkg::ProcessPkg	19
4.4.1	Informazioni sul package	19
4.4.2	Classi	19
4.4.2.1	Asset	19
4.4.2.2	Edge	19
4.4.2.3	ExitNode	19
4.4.2.4	MachineNode	19
4.4.2.5	Node	20
4.4.2.6	Process	20
4.4.2.7	QueueNode	20
4.4.2.8	ResourceNode	20
4.4.2.9	SourceNode	20
4.4.2.10	TransportEdge	20
4.5	DeGeOP::StorePkg::AnalysisPkg	21
4.5.1	Informazioni sul package	21
4.5.2	Classi	21
4.5.2.1	Analysis	21
4.5.2.2	Scenario	21
4.6	DeGeOP::StorePkg::PolygonPkg	22
4.6.1	Informazioni sul package	22
4.6.2	Classi	22
4.6.2.1	ConcretePolygon	22
4.6.2.2	ConcretePolygonFactory	22
4.6.2.3	Coordinate	22
4.6.2.4	Polygon	22
4.6.2.5	PolygonFactory	22
4.7	DeGeOP::ReducerPkg	23
4.7.1	Informazioni sul package	23
4.7.2	Classi	24
4.7.2.1	AnalysisReducer	24
4.7.2.2	AssetReducer	24
4.7.2.3	EdgeReducer	24
4.7.2.4	NodeReducer	24
4.7.2.5	OptionReducer	24
4.7.2.6	Reducer	24
4.7.2.7	ScenarioReducer	25
4.8	DeGeOP::CallManagerPkg	26
4.8.1	Informazioni sul package	26
4.8.2	Classi	26
4.8.2.1	CallManager	26
4.8.2.2	DataFromServer	27
4.8.2.3	DataToServer	27
4.9	DeGeOP::ActionPkg	28
4.9.1	Informazioni sul package	28
4.9.2	Classi	28
4.9.2.1	EdgeAction	28

4.10	DeGeOP::ActionPkg::ActionElementsPkg	29
4.10.1	Informazioni sul package	29
4.10.2	Classi	29
4.10.2.1	Action	29
4.10.2.2	AnalysisAction	29
4.10.2.3	AssetAction	29
4.10.2.4	NodeAction	29
4.10.2.5	OptionAction	30
4.10.2.6	ScenarioAction	30
4.11	DeGeOP::ActionPkg::ActionCreatorsPkg	31
4.11.1	Informazioni sul package	31
4.11.2	Classi	31
4.11.2.1	AnalysisActionCreator	31
4.11.2.2	AssetActionCreator	31
4.11.2.3	EdgeActionCreator	31
4.11.2.4	NodeActionCreator	31
4.11.2.5	OptionActionCreator	32
4.11.2.6	ScenarioActionCreator	32
4.12	DeGeOP::ViewPkg	33
4.12.1	Informazioni sul package	33
4.13	DeGeOP::ViewPkg::DeGeOPViewPkg	34
4.13.1	Informazioni sul package	34
4.13.2	Classi	35
4.13.2.1	ConcreteDeGeOPViewBuilder	35
4.13.2.2	DeGeOPView	35
4.13.2.3	DeGeOPViewBuilder	35
4.13.2.4	Director	35
4.14	DeGeOP::ViewPkg::MapComponentsPkg	36
4.14.1	Informazioni sul package	36
4.14.2	Classi	36
4.14.2.1	ButtonWrapper	36
4.14.2.2	MapWrapper	37
4.14.2.3	MessageWrapper	37
4.14.2.4	PolygonOperationWrapper	37
4.15	DeGeOP::ViewPkg::SidebarPkg	38
4.15.1	Informazioni sul package	38
4.15.2	Classi	39
4.15.2.1	Sidebar	39
4.15.2.2	SidebarFacade	39
4.16	DeGeOP::ViewPkg::SidebarPkg::FactorySidebarPkg	40
4.16.1	Informazioni sul package	40
4.16.2	Classi	41
4.16.2.1	AbstractAnalysisSidebarFactory	41
4.16.2.2	AbstractAssetSidebarFactory	41
4.16.2.3	AbstractEdgeSidebarFactory	41
4.16.2.4	AbstractNodeSidebarFactory	41
4.16.2.5	AbstractScenarioSidebarFactory	41
4.16.2.6	AbstractSidebarFactory	42
4.16.2.7	AnalysisSidebarFactory	42

4.16.2.8	EditAssetSidebarFactory	42
4.16.2.9	EditEdgeSidebarFactory	42
4.16.2.10	EditNodeSidebarFactory	42
4.16.2.11	EditScenarioSidebarFactory	42
4.16.2.12	InsertAssetSidebarFactory	42
4.16.2.13	InsertEdgeSidebarFactory	43
4.16.2.14	InsertNodeSidebarFactory	43
4.16.2.15	InsertScenarioSidebarFactory	43
4.16.2.16	ViewAssetSidebarFactory	43
4.16.2.17	ViewEdgeSidebarFactory	43
4.16.2.18	ViewNodeSidebarFactory	43
4.16.2.19	ViewScenarioSidebarFactory	43
4.17	DeGeOP::ViewPkg::SidebarPkg::ContentPkg	44
4.17.1	Informazioni sul package	44
4.17.2	Classi	45
4.17.2.1	AbstractAnalysisContent	45
4.17.2.2	AbstractAssetContent	45
4.17.2.3	AbstractContent	45
4.17.2.4	AbstractEdgeContent	45
4.17.2.5	AbstractNodeContent	45
4.17.2.6	AbstractScenarioContent	46
4.17.2.7	AnalysisContent	46
4.17.2.8	EditAssetContent	46
4.17.2.9	EditEdgeContent	46
4.17.2.10	EditNodeContent	46
4.17.2.11	EditScenarioContent	46
4.17.2.12	InsertAssetContent	46
4.17.2.13	InsertEdgeContent	46
4.17.2.14	InsertNodeContent	46
4.17.2.15	InsertScenarioContent	47
4.17.2.16	ViewAssetContent	47
4.17.2.17	ViewEdgeContent	47
4.17.2.18	ViewNodeContent	47
4.17.2.19	ViewScenarioContent	47
4.18	DeGeOP::ViewPkg::SidebarPkg::ButtonsPkg	48
4.18.1	Informazioni sul package	48
4.18.2	Classi	49
4.18.2.1	AbstractAnalysisButtons	49
4.18.2.2	AbstractAssetButtons	49
4.18.2.3	AbstractButtons	49
4.18.2.4	AbstractEdgeButtons	49
4.18.2.5	AbstractNodeButtons	49
4.18.2.6	AbstractScenarioButtons	50
4.18.2.7	AnalysisButtons	50
4.18.2.8	EditAssetButtons	50
4.18.2.9	EditEdgeButtons	50
4.18.2.10	EditNodeButtons	50
4.18.2.11	EditScenarioButtons	50
4.18.2.12	InsertAssetButtons	50

4.18.2.13	InsertEdgeButtons	50
4.18.2.14	InsertNodeButtons	50
4.18.2.15	InsertScenarioButtons	51
4.18.2.16	ViewAssetButtons	51
4.18.2.17	ViewEdgeButtons	51
4.18.2.18	ViewNodeButtons	51
4.18.2.19	ViewScenarioButtons	51
5	Stime di fattibilità e bisogno di risorse	52
5.1	JavaScript	52
5.2	React	52
6	Attività	53
6.1	Introduzione	53
6.2	Visualizzazione di default	54
6.3	Aggiunta asset	57
6.4	Aggiunta nodo	60
6.5	Aggiunta arco	62
6.6	Aggiunta scenario	64
6.7	Gestione Analisi	66
6.8	Selezione asset	68
6.9	Selezione nodo	70
6.10	Selezione arco	72
6.11	Selezione scenario	73
6.12	Modifica asset	75
6.13	Modifica nodo	77
6.14	Modifica arco	79
6.15	Modifica scenario	80
6.16	Eliminazione asset	82
6.17	Eliminazione nodo	83
6.18	Eliminazione arco	85
6.19	Eliminazione scenario	86
6.20	Errore di connessione	87
7	Tracciamento	89
7.1	Tracciamento classi-requisiti	89
7.2	Tracciamento requisiti-classi	109
A	Descrizione design pattern	129
A.1	Introduzione	129
A.2	Pattern Creazionali	130
A.2.1	Abstract Factory Method	130
A.2.1.1	Descrizione	130
A.2.1.2	Contestualizzazione	130
A.2.2	Builder	132
A.2.2.1	Descrizione	132
A.2.2.2	Contestualizzazione	132
A.2.3	Factory Method	133
A.2.3.1	Descrizione	133
A.2.3.2	Contestualizzazione	133

A.2.4	Singleton	134
A.2.4.1	Descrizione	134
A.2.4.2	Contestualizzazione	134
A.3	Pattern Strutturali	136
A.3.1	Facade	136
A.3.1.1	Descrizione	136
A.3.1.2	Contestualizzazione	136
A.4	Pattern Architetturali	137
A.4.1	Redux	137
A.4.1.1	Descrizione	137
A.4.1.2	Contestualizzazione	137

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è definire ad alto livello l'architettura del sistema, le sue componenti e l'interazione tra esse. Verrà quindi esplicitato il tracciamento tra le componenti software individuate e i requisiti presenti nel documento *Analisi dei requisiti v3.0.0*.

Progettando ad alto livello si è cercato di individuare il giusto trade-off tra:

- l'astrazione che cattura le proprietà salienti del prodotto a livello concettuale, cercando quindi di rimanere indipendenti dallo specifico linguaggio di programmazione utilizzato;
- l'implementazione effettiva della nostra architettura, indissolubilmente legata alle tecnologie utilizzate.

1.2 Scopo del prodotto

Lo scopo del prodotto consiste nella creazione di un'interfaccia web contenente una mappa geografica su cui potranno essere rappresentati:

- il processo produttivo aziendale;
- gli scenari di danno;
- i risultati dell'analisi dei rischi.

Il prodotto verrà utilizzato da agenti assicuratori per l'inserimento delle informazioni utili allo svolgimento dell'analisi dei rischi dell'assicurando.

L'interfaccia dovrà essere in grado di connettersi ai sistemi preesistenti di *RiskApp* per la memorizzazione e gestione dei dati inseriti. A causa del requisito di integrabilità, che è stato deciso di soddisfare, l'applicazione da sviluppare sarà parte integrante dell'attuale applicazione del proponente.

1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione dei documenti viene allegato il *Glossario v2.0.0*, nel quale verranno raccolte le spiegazioni di terminologia tecnica o ambigua, abbreviazioni ed acronimi. Per evidenziare un termine presente in tale documento, esso verrà marcato con il pedice *G*. Solo la prima occorrenza del termine in ogni sezione sarà marcata per non appesantire la lettura del documento.

Tutti i termini del glossario evidenziati sono link ipertestuali al glossario stesso; affinché funzionino correttamente è necessario che la posizione delle directory e dei file forniti non venga alterata.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- *Norme di progetto v3.0.0*.

1.4.2 Riferimenti informativi

- **capitolato d'appalto C3:** DeGeOP: A Designer and Geo-localizer Web App for Organizational Plants. Reperibile all'indirizzo:
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C3.pdf> ;
- *Analisi dei requisiti v3.0.0;*
- **Guide to the Software Engineering Body of Knowledge:** IEEE Computer Society. Software Engineering Coordinating Committee (Versione 2004):
 - Chapter 3: Software Design;
- **slide del corso di Ingegneria del Software**
<http://www.math.unipd.it/~tullio/IS-1/2016/>;
- **Design Patterns** - Elementi per il riuso di software a oggetti - Gamma, Helm, Johnson, Vlissides;
- **Documentazione di Redux.** Reperibile all'indirizzo:
<http://redux.js.org/>;
- **Documentazione di REST.** Reperibile all'indirizzo:
https://en.wikipedia.org/wiki/Representational_state_transfer.

2 Tecnologie utilizzate

2.1 Introduzione

In questa sezione vengono descritte le tecnologie su cui si basa lo sviluppo del progetto. Per ognuna di esse verrà indicato l'ambito di utilizzo della tecnologia, i vantaggi e eventuali svantaggi che ne derivano. La scelta delle tecnologie non è stata vincolata in alcun modo dal proponente, anche se alcune decisioni a riguardo sono state prese tenendo conto delle tecnologie da loro già utilizzate.

Tecnologia	Utilizzo
Alexa Voice Service	Libreria per i servizi di riconoscimento vocale
Bing Map	Libreria per la fornitura di mappe satellitari
CSS3	Linguaggio per la formattazione delle pagine web
HammerJS	Libreria per gestire le gesture su tablet
HTML5	Linguaggio per la costruzione di pagine web
JavaScript ES6	Linguaggio principale in cui è sviluppata l'applicazione
JSON	Formato dati utilizzato per lo scambio di informazioni
JSX	Estensione <i>JavaScript</i> per l'integrazione del codice <i>HTML</i>
Node.js	Ambiente operativo per utilizzare JavaScript lato server
OpenLayers	Libreria per la gestione della mappa e del grafo
Open Street Map	Libreria per la fornitura di mappe in formato vettoriale
React	Costruzione dell'interfaccia grafica
ReactColor	Libreria per gestire la palette di colori
React-Redux	libreria per interfacciare <i>React</i> con Redux
React Toolbox	Libreria che implementa la specifica di Material Design
Redux	Libreria per l'implementazione dell'architettura
SVG	Scalable Vector Graphics

Tabella 1: Panoramica generale delle tecnologie usate nel progetto

2.2 Alexa Voice Service

Descrizione	Alexa è una libreria per il riconoscimento vocale sviluppata e offerta da Amazon; è un servizio in continua evoluzione e la logica che l'accompagna è sempre più intelligente, oltre a essere gratuito e di facile integrazione.
Vantaggi	- gratuito.
Svantaggi	- servizio giovane, non ancora completo.
Utilizzo	Alexa verrà utilizzato per offrire il servizio di riconoscimento vocale della nostra applicazione.

2.3 Bing Map

Descrizione	Libreria che fornisce mappe in modalità satellitare ad una buona risoluzione. Di proprietà della Microsoft.
Vantaggi	- una delle più complete mappe satellitari dal punto di vista della copertura.
Svantaggi	- oltre le 125.000 transazioni è necessario pagare.
Utilizzo	Bing Map viene utilizzato in modo indiretto dalla libreria OpenLayers per fornire una vista satellitare nella mappa dell'applicazione.

2.4 CSS3

Descrizione	È un linguaggio utilizzato per la presentazione di documenti HTML. Lo standard viene definito dal W3C .
Vantaggi	- assicura maggiore manutenibilità e riutilizzo grazie alla separazione tra presentazione e struttura; - raccomandato dal W3C.
Svantaggi	- specifica ufficiale non completata; - non supportato pienamente da tutti i browser.
Utilizzo	Viene utilizzato per definire il layout dell'applicazione.

2.5 HammerJS

Descrizione	È una libreria JavaScript per riconoscere le gesture tipiche dei tablet come drag, pinch, zoom.
Vantaggi	- leggero; - gestisce tutte le gesture richieste dall'analisi dei requisiti come spiegato nella sezione 4.12 .
Utilizzo	Viene utilizzato per riconoscere le gesture del tablet in caso l'applicazione venga utilizzata con questo tipo di device.

2.6 HTML5

Descrizione	È un <i>linguaggio di markup</i> utilizzato per definire la struttura delle pagine web. Lo standard viene definito dal W3C.
Vantaggi	- fornisce un set di <i>tag</i> più vasto rispetto alle vecchie versioni, con nuovi tag semanticci; - raccomandato dal W3C.
Svantaggi	- non supportato pienamente da tutti i browser.
Utilizzo	Viene utilizzato per definire il layout dell'applicazione.

2.7 JavaScript ES6

Descrizione	JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi, utilizzato principalmente nella programmazione Web lato client. Le caratteristiche più importanti di questo linguaggio sono:
	<ul style="list-style-type: none"> • eventi: quando l'utente interagisce con la pagina Web in vari modi, come ad esempio mouse e tastiera, viene generato un evento; JavaScript gestisce tali eventi, i quali possono avviare un'azione registrata in un gestore di eventi; • tipizzazione dinamica: il programmatore non è tenuto a specificare il tipo degli oggetto che utilizza; • paradigma a prototipi: stile di programmazione orientato ad oggetti in cui l'ereditarietà è implementata tramite il riuso di oggetti esistenti, basandosi sul loro prototipo.
	In particolare, il <i>gruppoG</i> si baserà sull'utilizzo della specifica JavaScript ES6, che definisce significativi cambiamenti sintattici per la scrittura di applicazioni complesse in modo più semplice.
Vantaggi	<ul style="list-style-type: none"> - facilità di utilizzo; - larga disponibilità di documentazione; - conoscenza pregressa del linguaggio; - maggior supporto da parte dei browser rispetto alle alternative, come ad esempio ActionScript.
Svantaggi	<ul style="list-style-type: none"> - variazione di interpretazione a seconda del browser; - la tipizzazione dinamica è frequentemente fonte di errori.
Utilizzo	JavaScript è il linguaggio base con cui si svilupperà l'applicazione <i>DeGeOP</i> . Di conseguenza è anche il linguaggio utilizzato maggiormente dalle librerie esterne da noi sfruttate.

2.8 JSON

Descrizione	Formato dati utilizzato per lo scambio di informazioni tra il client (ovvero il nostro prodotto) e il server (ovvero il prodotto di <i>RiskApp</i>).
Vantaggi	<ul style="list-style-type: none"> - standard per lo scambio di dati; - meno verboso di alternative come XML.
Utilizzo	Viene utilizzato per lo scambio di dati tra l'applicazione <i>DeGeOPe</i> e il server di <i>RiskApp</i> .

2.9 JSX

Descrizione	JSX è un linguaggio orientato agli oggetti staticamente tipizzato. È un'estensione di JavaScript. I file in linguaggio JSX vengono poi tradotti in JavaScript.
Vantaggi	<ul style="list-style-type: none"> - permette di utilizzare tag in stile HTML all'interno delle componenti React; - facilità di utilizzo; - viene compilato, quindi permette di scoprire gli errori a tempo di compilazione; - il suo utilizzo in combinazione con React è altamente consigliato; - maggiore leggibilità.
Utilizzo	Viene utilizzato come sintassi all'interno di React.

2.10 Node.js

Descrizione	Ambiente operativo per utilizzare JavaScript in ambito server.
Vantaggi	<ul style="list-style-type: none"> - combinato con npm permette di creare un ambiente di sviluppo molto facilitato.
Utilizzo	Viene utilizzato per far avviare la nostra applicazione.

2.11 OpenLayers

Descrizione	E' una libreria JavaScript per visualizzare mappe interattive nei browser web. OpenLayers offre API G ai programmati per poter accedere a diverse fonti d'informazioni cartografiche in Internet: mappe del progetto OpenStreetMap, mappe sotto licenze non-libere (Google Maps, Bing, Yahoo), Web Feature Service, ecc. E' coperto da licenza BSD.
Vantaggi	<ul style="list-style-type: none"> - buona documentazione; - maggiori funzionalità e flessibilità rispetto ai concorrenti, come ad esempio Leaflet.
Svantaggi	- più pesante di alcuni concorrenti, come Leaflet.
Utilizzo	Viene utilizzato per gestire la mappa.

2.12 Open Street Map

Descrizione	E' una libreria JavaScript per fornire informazioni geografiche in formato vettoriale.
Vantaggi	- utile alla nostra applicazione in quanto mostra il perimetro di molti edifici.
Svantaggi	- mancanza della vista satellitare.
Utilizzo	OpenStreetMap viene utilizzato in modo indiretto dalla libreria OpenLayers per fornire una vista vettoriale nella mappa dell'applicazione.

2.13 React

Descrizione	E' una libreria JavaScript <i>open source</i> mantenuta da Facebook e Instagram utile alla costruzione di interfacce grafiche. Per fare ciò, React utilizza componenti indipendenti e riusabili che ereditano dalla classe base astratta React.Component. Le componenti devono implementare il metodo <i>render()</i> che si occupa di rappresentare la <i>componente</i> sul browser. Le caratteristiche più importanti di questa libreria sono:
	<ul style="list-style-type: none">• One-way-data-flow: meccanismo tramite il quale le proprietà (un insieme di valori immutabili passato al render di un componente) non possono essere direttamente modificate. Queste proprietà possono però essere modificate da una <i>callback</i>;• Virtual DOM: virtualizzazione operata da React per effettuare un re-rendering efficiente dei componenti. Consiste in:<ul style="list-style-type: none">- replicare il DOM in memoria;- individuare le differenze tra il DOM reale e il DOM virtuale;- aggiornare le informazioni del DOM reale sulla base delle differenze precedentemente individuate.• utilizzo di JSX.
Vantaggi	<ul style="list-style-type: none">- facile da testare in quanto il DOM virtuale è implementato interamente in JavaScript;- agevola il riuso del codice grazie all'uso delle componenti, le quali possono essere combinate e collegate tra loro;- gestione automatica degli aggiornamenti dell'interfaccia grafica.
Svantaggi	<ul style="list-style-type: none">- riduzione per far dialogare le componenti, React suggerisce di portare lo stato nelle componenti di più alto livello possibile. In questo modo le componenti di più basso livello sono prevalente componenti presentazionali, mentre le componenti di più alto livello risultano essere cariche di informazioni;
Svantaggi	<ul style="list-style-type: none">- manca di librerie per la gestione del model perché si occupa solamente della costruzione dell'interfaccia grafica. È necessaria esperienza per la scelta di librerie aggiuntive.
Utilizzo	React viene utilizzata per la costruzione dell'interfaccia grafica dell'applicazione.

2.14 ReactColor

Descrizione	E' una libreria JavaScript per creare una paletta di colori RGB.
Vantaggi	- buona documentazione.
Utilizzo	Viene utilizzata per la creazione di un color picker.

2.15 React-Redux

Descrizione	Libreria che facilita l'integrazione tra Redux e React.
Vantaggi	- facilita l'integrazione tra Redux e React.
Utilizzo	Le classi JavaScript vengono passate ad una funzione della libreria per ottenere una nuova classe che sfrutta React-Redux.

2.16 React Toolbox

Descrizione	E' una libreria JavaScript composta da un insieme di componenti React che implementano la specifica del Material Design di Google.
Vantaggi	- vasta varietà di elementi grafici; - elementi grafici e temi facilmente personalizzabili; - ben documentata; - aiuta la separazione tra presentazione e contenuto grazie all'utilizzo usa i moduli CSS_G al posto dello stile inline, a differenza ad esempio di Material-UI.
Utilizzo	React Toolbox viene utilizzata per implementare alcune le componenti grafiche secondo la specifica del Material Design.

2.17 Redux

Descrizione	Libreria per l'implementazione dell'architettura che si occupa di gestire le interazioni tra la business logic e la presentazione. Per fare ciò:
	<ul style="list-style-type: none">• implementa un <i>design pattern</i> architettonicale da usare il sostituzione a MVC, come descritto in Redux;• offre delle API apposite per la gestione degli elementi del design pattern descritto al punto precedente.
Vantaggi	- integrabile facilmente con React; - largamente utilizzato; - ben documentata.
Utilizzo	Redux viene utilizzato per implementare l'architettura di <i>DeGeOP</i> .

2.18 SVG

Descrizione	Standard per la scrittura di immagini in formato vettoriale.
Vantaggi	- standard aperto.
Utilizzo	SVG viene utilizzato per aggiungere oggetti personalizzati alla mappa.

3 Descrizione architettura

3.1 Introduzione

Il prodotto *DeGeOP*, creato dal *gruppo_G* *Zephyrus*, sarà integrabile nell'attuale applicazione del proponente *RiskApp*.

Per esporre l'architettura dell'applicazione si procederà con approccio top-down, partendo cioè da una visione generale delle componenti che distinguono il sistema, per poi analizzare in dettaglio la conformazione di tali componenti.

Il sistema attuale di *RiskApp* presenta un'architettura client-server.

3.2 Server

DeGeOP si conoscerà al server di *RiskApp* esclusivamente utilizzando le *API_G* REST fornite dal proponente stesso e riportate in questa sezione. Il gruppo non ha quindi accesso all'implementazione del server di *RiskApp*.

3.2.1 Chiamate REST

L'interfaccia REST proposta da *RiskApp* fornisce l'accesso alle seguenti entità:

- Customer
- Graph
- *Asset_G*
- Node
- Edge

Per ognuna di queste è possibile fare una chiamata REST usando i verbi http. (GET, OPTION, POST, PUT, UPDATE, DELETE)

Tutte queste entità sono identificate univocamente da uno uuid formato come una stringa di cifre esadecimale con la seguente struttura:

XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Il verbo OPTION viene utilizzato all'interno del server di Riskapp per ottenere informazioni sulla struttura (nomi dei campi dati e relativi tipi di dato) delle entità ricevute a seguito di una chiamata GET o necessarie per le chiamate POST e PUT. Inoltre per determinati campi, la chiamata OPTION restituisce tutti i possibili valori, nel caso siano limitati al lato server, per il campo stesso.

3.2.1.1 Customer

- **descrizione:** l'entità customer contiene le informazioni del cliente;
- **chiamata:** /mitsuko/v01/customer/<uuid>/uuid/
 - **operazioni utilizzate:** GET.

3.2.1.2 Graph

- **descrizione:** l'entità grafo proposta da *RiskApp* contiene le informazioni relative al processo produttivo e ai suoi *nodi_G*;
- **chiamata:** /mitsuko/v01/graph/<uuid>/uuid/
 - **operazioni utilizzate:** GET.

3.2.1.3 Asset

- **descrizione:** l'entità asset rappresenta un asset del cliente con le relative informazioni;
- **chiamata:** /mitsuko/v01/customer/<uuid>/asset/new/
 - **operazioni utilizzate:** POST.
- **chiamata:** /mitsuko/v01/asset/<uuid>/uuid/
 - **operazioni utilizzate:** GET, PUT, OPTIONS, UPDATE, DELETE.

3.2.1.4 Node

- **descrizione:** l'entità node rappresenta un elemento di interesse strategico all'interno di un asset;
- **chiamata:** /mitsuko/v01/graph/<uuid>/node/new/
 - **operazioni utilizzate:** POST.
- **chiamata:** /mitsuko/v01/node/<uuid>/uuid/
 - **operazioni utilizzate:** GET, PUT, OPTIONS, UPDATE, DELETE.

3.2.1.5 Edge

- **descrizione:** l'entità edge rappresenta un collegamento fra due nodi;
- **chiamata:** /mitsuko/v01/graph/<uuid>/edge/new/
 - **operazioni utilizzate:** POST.
- **chiamata:** /mitsuko/v01/edge/<uuid>/uuid/
 - **operazioni utilizzate:** GET, PUT, OPTIONS, UPDATE, DELETE.

3.3 Client

Il proponente ha fornito l'ambiente di sviluppo contenente la loro attuale applicazione, denominata in figura come "RiskApp", su cui il gruppo potrà integrare DeGeOP. Il prodotto sarà sviluppato come una single-page accessibile cliccando sulla scheda di menu "Process and analysis".

3.3.1 Design architetturale di DeGeOP

E' stato scelto di utilizzare il design architetturale Redux (per maggiori informazioni su questo pattern, si veda l'appendice A.4.1). Per una migliore gestione e uso di questo pattern si è deciso di rafforzare l'uso classico di Redux, incapsulando le varie componenti in una struttura a classi. Rispetto all'architettura base, ovvero Redux, sono quindi presenti altre componenti:

- **ActionCreators**: struttura il cui compito è creare le "azioni", ossia le operazioni che si occupano di interagire con lo *store_G*. Le actions come fornite da Redux permettono di definire azioni potenzialmente invalide o la cui esecuzione potrebbe portare ad errori. È quindi nata la necessità di incapsulare il concetto di azione e devolvere il compito della sua creazione ad una *componente_G* propria;
- **Reducer_G**: classe di utilità che accetta *action_G* generiche e le reindirizza alle giuste implementazioni per gestire l'azione in analisi. Anche in questo caso l'incapsulazione è stata adottata per fornire una migliore interfaccia di utilizzo dei reducer.

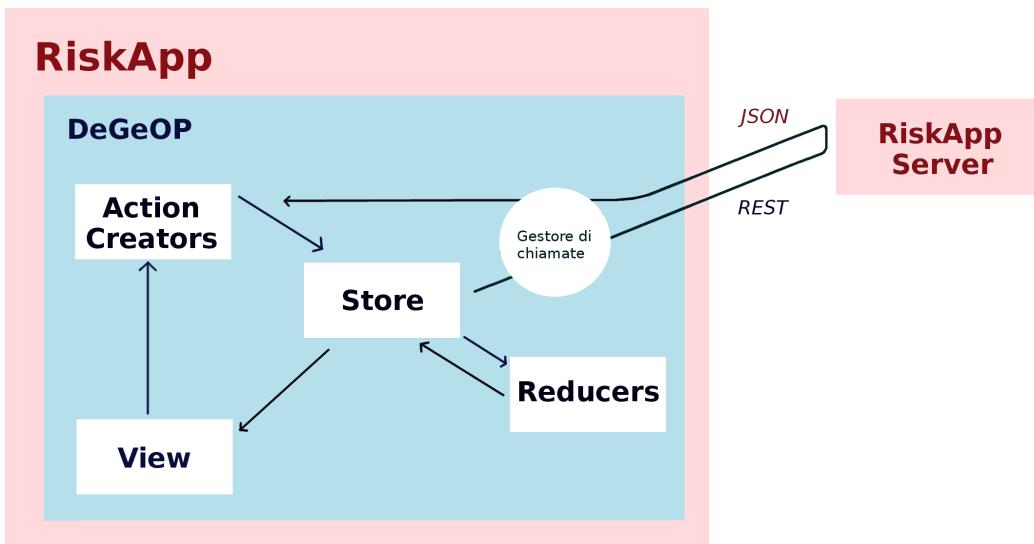


Figura 1: Architettura di base

4 Componenti

4.1 DeGeOP

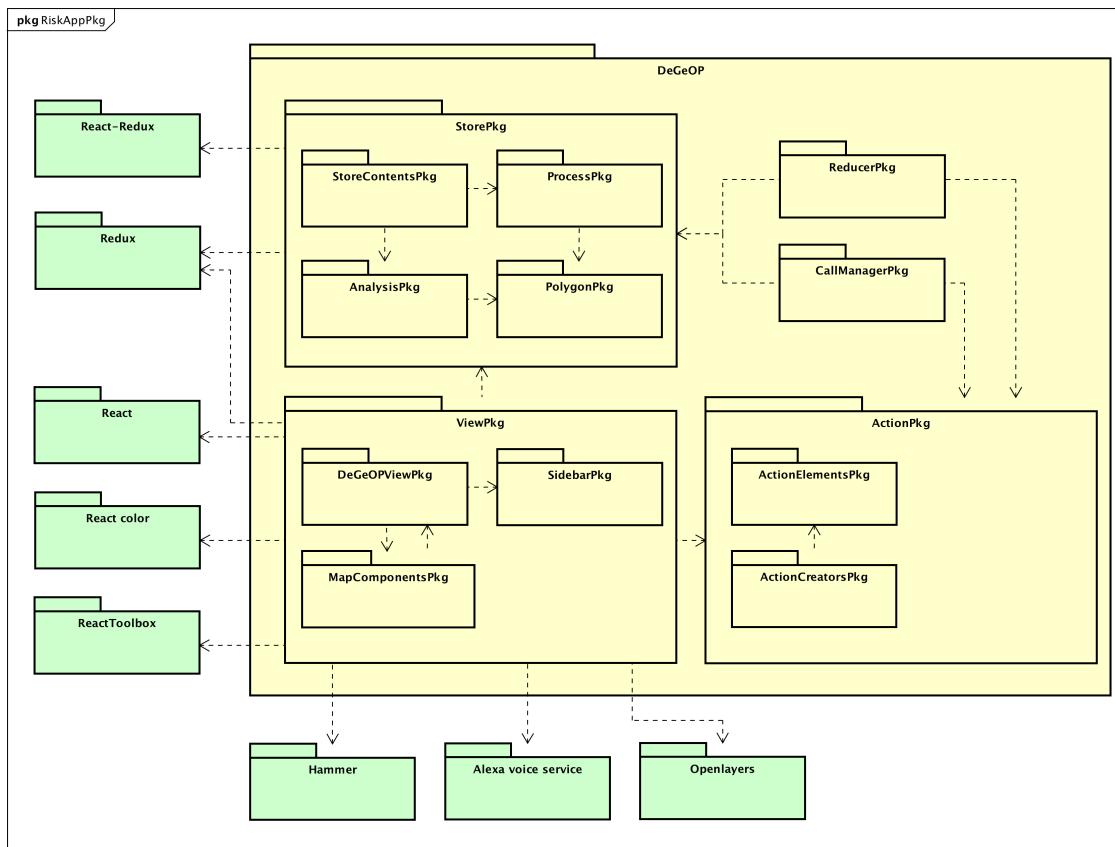


Figura 2: Schema componente DeGeOP

4.1.1 Informazioni sul package

- **descrizione:** racchiude tutte le componenti necessarie per il front-end del prodotto;
- **package contenuti:**
 - DeGeOP::[ActionPkg](#);
 - DeGeOP::[CallManagerPkg](#);
 - DeGeOP::[ReducerPkg](#);
 - DeGeOP::[StorePkg](#);
 - DeGeOP::[ViewPkg](#).

4.2 DeGeOP::StorePkg

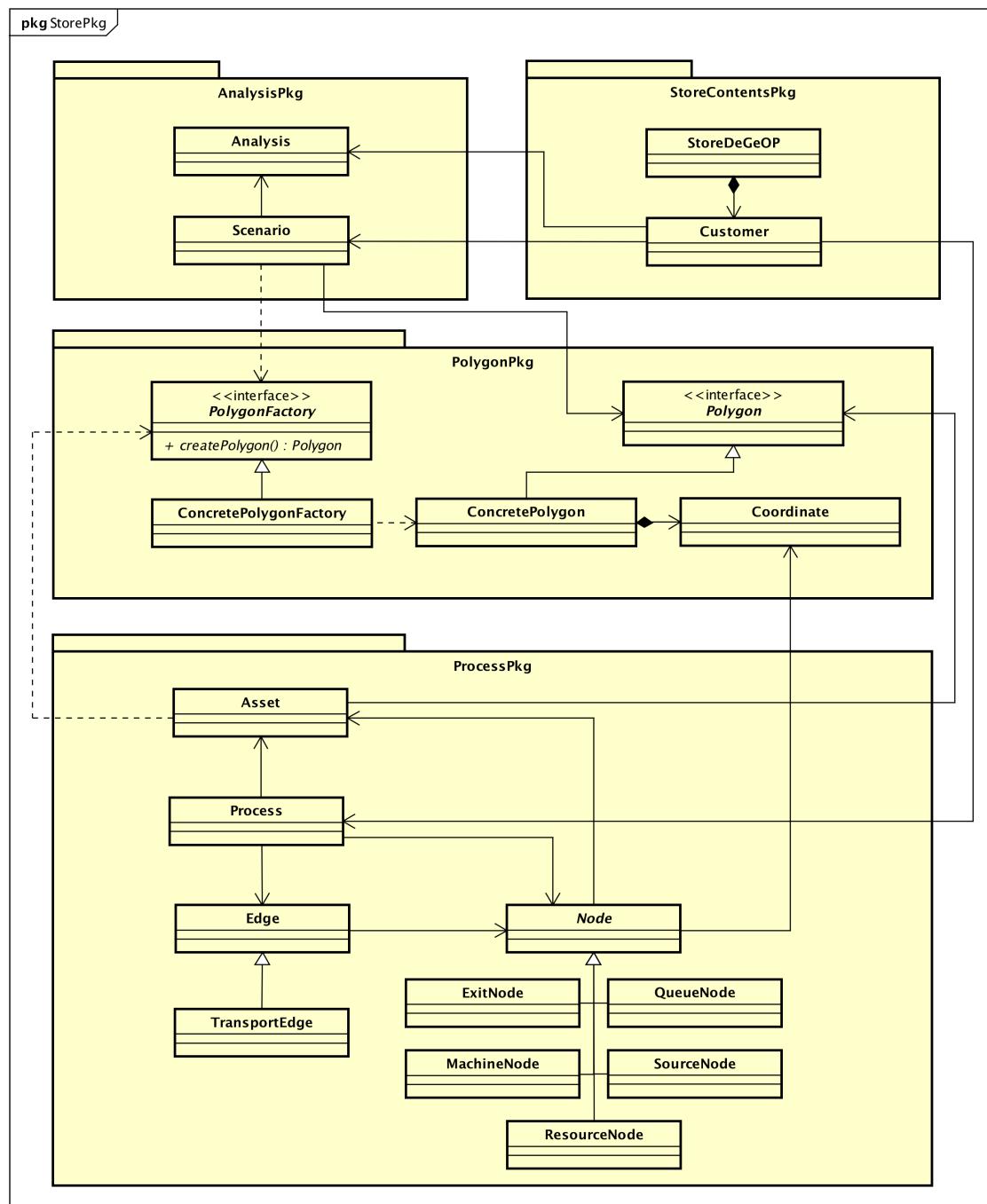


Figura 3: Schema componente DeGeOP::StorePkg

4.2.1 Informazioni sul package

- **descrizione:** racchiude le componenti utilizzate per la memorizzazione e rappresentazione dei dati;
- **padre:** [DeGeOP](#);
- **package contenuti:**

- StorePkg::[AnalysisPkg](#);
- StorePkg::[PolygonPkg](#);
- StorePkg::[ProcessPkg](#);
- StorePkg::[StoreContentsPkg](#).

- **interazioni con altri package:**

- IN CallManagerPkg: subscribe sullo store;
- IN ReducerPkg: applicazione di cambiamenti di stato;
- IN ViewPkg: subscribe sullo store;
- OUT React-Redux: utilizzo di Provider per evitare di passare lo store come proprietà alle componenti React;
- OUT Redux: creazione Store utilizzando il metodo createStore().

4.3 DeGeOP::StorePkg::StoreContentsPkg

4.3.1 Informazioni sul package

- **descrizione:** racchiude le componenti che implementano il concetto di store dell'architettura Redux;
- **padre:** [StorePkg](#);
- **interazioni con altri package:**
 - OUT AnalysisPkg: riferimento ad analisi di danno;
 - OUT ProcessPkg: riferimento a processo;
 - OUT React-Redux: utilizzo del Provider;
 - OUT Redux: creazione store.
- **classi contenute:**
 - Customer;
 - Options;
 - StoreDeGeOP.

4.3.2 Classi

4.3.2.1 Customer

- **descrizione:** rappresenta l'assicurando;
- **utilizzo:** viene utilizzato nello Store per memorizzare l'assicurando;
- **relazioni con altre classi:**
 - OUT Analysis;
 - OUT Scenario.

4.3.2.2 Options

- **descrizione:** classe contenente le associazioni tra codice e valore dei campi dati degli altri elementi dello store;
- **utilizzo:** viene utilizzata per associare codice e valore dei campi dati degli altri elementi dello store.

4.3.2.3 StoreDeGeOP

- **descrizione:** rappresenta una classe che incapsula uno Store creato utilizzando Redux;
- **utilizzo:** viene utilizzato per memorizzare lo stato dell'applicazione. Le componenti che effettuano il subscribe sullo Store verranno notificate ad ogni cambiamento di stato dello Store.

4.4 DeGeOP::StorePkg::ProcessPkg

4.4.1 Informazioni sul package

- **descrizione:** racchiude le componenti necessarie alla rappresentazione del processo produttivo dell'assicurando;
- **padre:** [StorePkg](#);
- **interazioni con altri package:**
 - IN StoreContentsPkg: riferimento a processo;
 - OUT PolygonPkg: riferimento ad un poligono.
- **classi contenute:**
 - Asset;
 - Edge;
 - ExitNode;
 - MachineNode;
 - Node;
 - Process;
 - QueueNode;
 - ResourceNode;
 - SourceNode;
 - TransportEdge.

4.4.2 Classi

4.4.2.1 Asset

- **descrizione:** rappresenta un fabbricato di interesse per il processo produttivo dell'assicurando;
- **utilizzo:** sono contenuti all'interno di Process.

4.4.2.2 Edge

- **descrizione:** rappresenta un arco che collega due nodi tra di loro; un arco indica che i nodi sono in correlazione tra di loro;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.3 ExitNode

- **descrizione:** rappresenta un nodo di tipo Uscita;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.4 MachineNode

- **descrizione:** rappresenta un nodo di tipo Macchina;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.5 Node

- **descrizione:** rappresenta un nodo contenuto all'interno di un Asset ;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.6 Process

- **descrizione:** rappresenta un processo produttivo dell'azienda dell'assicurando;
- **utilizzo:** è memorizzato nello Store.

4.4.2.7 QueueNode

- **descrizione:** rappresenta un nodo di tipo Coda;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.8 ResourceNode

- **descrizione:** rappresenta un nodo di tipo Risorsa;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.9 SourceNode

- **descrizione:** rappresenta un nodo di tipo Sorgente;
- **utilizzo:** è contenuto all'interno di Process.

4.4.2.10 TransportEdge

- **descrizione:** rappresenta un arco di tipo Trasporto;
- **utilizzo:** è contenuto all'interno di Process.

4.5 DeGeOP::StorePkg::AnalysisPkg

4.5.1 Informazioni sul package

- **descrizione:** racchiude le componenti necessarie alla rappresentazione dell'analisi di danno relative al processo produttivo dell'assicurando;
- **padre:** [StorePkg](#);
- **interazioni con altri package:**
 - IN StoreContentsPkg: riferimento ad analisi di danno;
 - OUT PolygonPkg: riferimento ad un poligono.
- **classi contenute:**
 - Analysis;
 - Scenario.

4.5.2 Classi

4.5.2.1 Analysis

- **descrizione:** rappresenta un risultato di un'analisi di danno relativo ad uno scenario;
- **utilizzo:** i metodi di questa classe saranno utilizzati per calcolare dati riassuntivi sull'analisi di danno;
- **relazioni con altre classi:**
 - IN Customer.

4.5.2.2 Scenario

- **descrizione:** rappresenta uno scenario di danno;
- **utilizzo:** viene utilizzata nello store per rappresentare gli scenari di danno;
- **relazioni con altre classi:**
 - IN Customer.

4.6 DeGeOP::StorePkg::PolygonPkg

4.6.1 Informazioni sul package

- **descrizione:** racchiude le componenti necessarie alla rappresentazione dell'area degli asset e degli scenari di danno;
- **padre:** [StorePkg](#);
- **interazioni con altri package:**
 - IN AnalysisPkg: riferimento ad un poligono;
 - IN ProcessPkg: riferimento ad un poligono.
- **classi contenute:**
 - [ConcretePolygon](#);
 - [ConcretePolygonFactory](#);
 - [Coordinate](#);
 - [Polygon](#);
 - [PolygonFactory](#).

4.6.2 Classi

4.6.2.1 ConcretePolygon

- **descrizione:** rappresenta un poligono;
- **utilizzo:** viene istanziata da [ConcretePolygonFactory](#); viene utilizzata in Asset e Scenario.

4.6.2.2 ConcretePolygonFactory

- **descrizione:** gestisce la creazione concreta dei poligoni;
- **utilizzo:** implementazione di [PolygonFactory](#); è la classe concreta da istanziare per gestire la creazione di un poligono.

4.6.2.3 Coordinate

- **descrizione:** rappresenta una coordinata geografica;
- **utilizzo:** è utilizzata all'interno di [Polygon](#) per delimitarne i suoi vertici.

4.6.2.4 Polygon

- **descrizione:** interfaccia che rappresenta il poligono;
- **utilizzo:** fornisce i metodi del poligono.

4.6.2.5 PolygonFactory

- **descrizione:** interfaccia che si occupa della costruzione dei poligoni;
- **utilizzo:** viene usata dalle classi Scenario e Asset per la costruzione dei poligoni.

4.7 DeGeOP::ReducerPkg

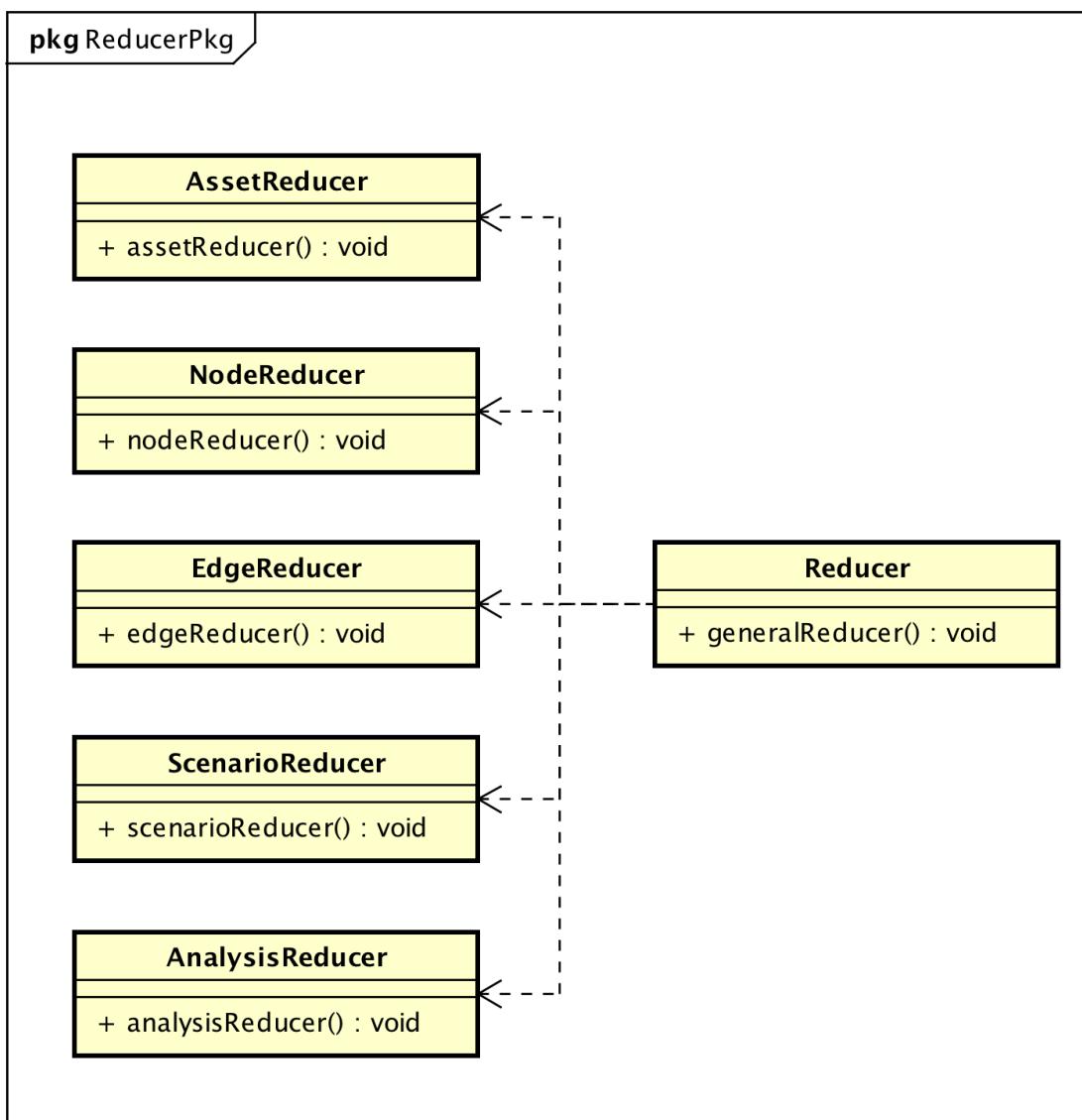


Figura 4: Schema componente DeGeOP::ReducerPkg

4.7.1 Informazioni sul package

- **descrizione:** racchiude le componenti necessarie all'implementazione dei reducer secondo l'architettura Redux;
- **padre:** [DeGeOP](#);
- **interazioni con altri package:**
 - OUT ActionPkg: utilizzo di azioni ;
 - OUT StorePkg: applicazione di cambiamenti di stato.
- **classi contenute:**

- AnalysisReducer;
- AssetReducer;
- EdgeReducer;
- NodeReducer;
- OptionReducer;
- Reducer;
- ScenarioReducer.

4.7.2 Classi

4.7.2.1 AnalysisReducer

- **descrizione:** rappresenta il reducer dell'analisi;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti le analisi;
- **relazioni con altre classi:**
 - IN Reducer.

4.7.2.2 AssetReducer

- **descrizione:** rappresenta il reducer dell'asset;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti l'asset.

4.7.2.3 EdgeReducer

- **descrizione:** rappresenta il reducer dell'arco;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti gli archi.

4.7.2.4 NodeReducer

- **descrizione:** rappresenta il reducer del nodo;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti i nodi.

4.7.2.5 OptionReducer

- **descrizione:** rappresenta il reducer delle option;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti l'asset.

4.7.2.6 Reducer

- **descrizione:** accetta una action generica in input e la reindirizza al giusto reducer;
- **utilizzo:** il suo metodo viene utilizzato per catturare un'azione e generare un nuovo stato sullo Store;
- **relazioni con altre classi:**
 - OUT AnalysisReducer;
 - OUT ScenarioReducer.

4.7.2.7 ScenarioReducer

- **descrizione:** rappresenta il reducer dello scenario;
- **utilizzo:** il suo metodo gestisce le operazioni sullo store riguardanti gli scenari;
- **relazioni con altre classi:**
 - IN Reducer.

4.8 DeGeOP::CallManagerPkg

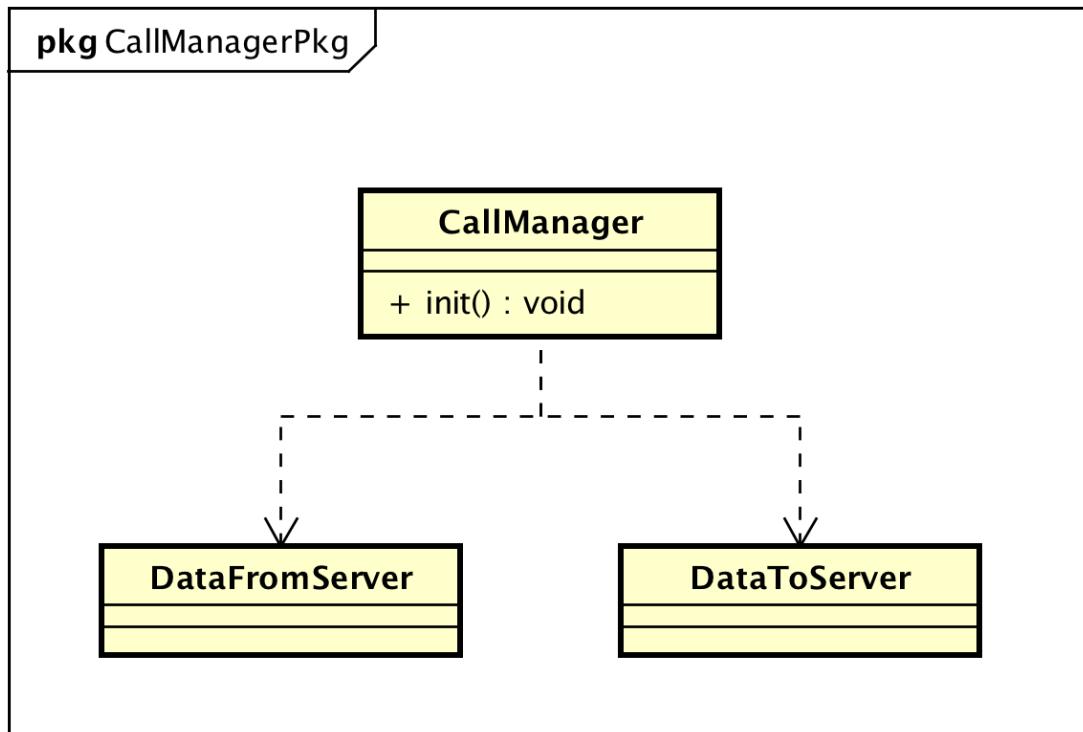


Figura 5: Schema componente DeGeOP::CallManagerPkg

4.8.1 Informazioni sul package

- **descrizione:** racchiude le componenti necessarie alla comunicazione dei dati verso il server;
- **padre:** [DeGeOP](#);
- **interazioni con altri package:**
 - OUT ActionPkg: dispatch di azioni;
 - OUT StorePkg: subscribe sullo store.
- **classi contenute:**
 - CallManager;
 - DataFromServer;
 - DataToServer.

4.8.2 Classi

4.8.2.1 CallManager

- **descrizione:** rappresenta il gestore delle chiamate da e verso il server;

- **utilizzo:** viene usato per effettuare chiamate REST da e verso il server. I suoi metodi si occupano di inizializzare lo Store emettendo Action quando l'applicazione viene aperta. Sfrutta i metodi della classe DataFromServer per effettuare la conversione da JSON ad oggetto logico. Inoltre, la classe effettuerà il subscribe allo Store per ricevere i cambiamenti di stato e mantenere aggiornate le informazioni sul server. La classe avrà un metodo che identifica cos'è cambiato nello Store e farà la chiamata REST appropriata, sfruttando la classe DataToServer per trasformare l'oggetto logico in JSON.

4.8.2.2 DataFromServer

- **descrizione:** rappresenta il gestore dei dati ricevuti dal server;
- **utilizzo:** gestisce i dati ricevuti dal server. I suoi metodi si occupano di effettuare la conversione dei JSON ricevuti dal server in oggetti logici.

4.8.2.3 DataToServer

- **descrizione:** rappresenta il gestore dei dati da inviare verso il server;
- **utilizzo:** gestisce i dati che dovranno essere inviati al server. I suoi metodi effettuano la conversione degli oggetti logici in JSON.

4.9 DeGeOP::ActionPkg

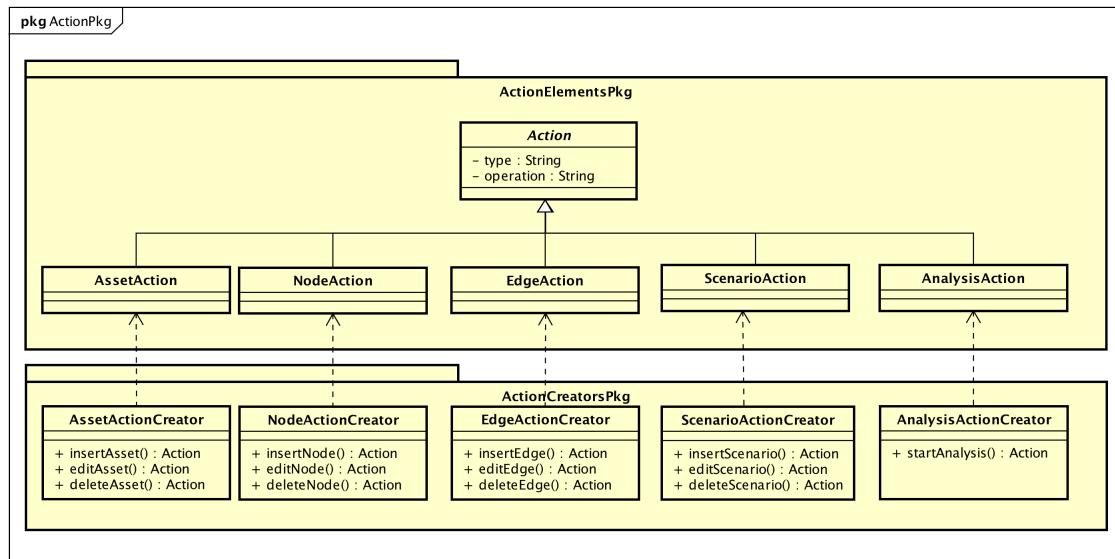


Figura 6: Schema componente DeGeOP::ActionPkg

4.9.1 Informazioni sul package

- **descrizione:** racchiude le componenti utilizzate per implementare le action dell'architettura Redux. Le action vengono create e ne viene fatto il dispatch verso lo store. Un reducer gestirà una action per produrre un cambiamento di stato sullo store;
- **padre:** [DeGeOP](#);
- **package contenuti:**
 - [ActionPkg::ActionCreatorsPkg](#);
 - [ActionPkg::ActionElementsPkg](#).
- **interazioni con altri package:**
 - IN [CallManagerPkg](#): dispatch di azioni;
 - IN [ReducerPkg](#): utilizzo di azioni ;
 - IN [ViewPkg](#): dispatch di azioni.
- **classi contenute:**
 - [EdgeAction](#).

4.9.2 Classi

4.9.2.1 EdgeAction

- **descrizione:** rappresenta un'azione relativa agli archi;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer.

4.10 DeGeOP::ActionPkg::ActionElementsPkg

4.10.1 Informazioni sul package

- **descrizione:** racchiude le componenti che rappresentano effettivamente le azioni;
- **padre:** [ActionPkg](#);
- **interazioni con altri package:**
 - IN ActionCreatorsPkg: creazione di azioni.
- **classi contenute:**
 - Action;
 - AnalysisAction;
 - AssetAction;
 - NodeAction;
 - OptionAction;
 - ScenarioAction.

4.10.2 Classi

4.10.2.1 Action

- **descrizione:** una classe astratta che rappresenta una generica azione di cui può essere fatto il dispatch;
- **utilizzo:** i suoi membri vengono usati dai reducer per completare una azione.

4.10.2.2 AnalysisAction

- **descrizione:** rappresenta un'azione relativa alle analisi di danno;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer;
- **relazioni con altre classi:**
 - IN AnalysisActionCreator.

4.10.2.3 AssetAction

- **descrizione:** rappresenta un'azione relativa agli asset;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer.

4.10.2.4 NodeAction

- **descrizione:** rappresenta un'azione relativa ai nodi;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer.

4.10.2.5 OptionAction

- **descrizione:** rappresenta un'azione relativa all'oggetto option;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer.

4.10.2.6 ScenarioAction

- **descrizione:** rappresenta un'azione relativa agli scenari di danno;
- **utilizzo:** l'azione viene creata da un apposito ActionCreator per essere poi inviata ad un reducer;
- **relazioni con altre classi:**
 - IN ScenarioActionCreator.

4.11 DeGeOP::ActionPkg::ActionCreatorsPkg

4.11.1 Informazioni sul package

- **descrizione:** racchiude le componenti che gestiscono la creazione delle azioni;
- **padre:** [ActionPkg](#);
- **interazioni con altri package:**
 - OUT ActionElementsPkg: creazione di azioni.
- **classi contenute:**
 - AnalysisActionCreator;
 - AssetActionCreator;
 - EdgeActionCreator;
 - NodeActionCreator;
 - OptionActionCreator;
 - ScenarioActionCreator.

4.11.2 Classi

4.11.2.1 AnalysisActionCreator

- **descrizione:** rappresenta la factory di azioni relative alle analisi di danno;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative alle analisi di danno;
- **relazioni con altre classi:**
 - OUT AnalysisAction.

4.11.2.2 AssetActionCreator

- **descrizione:** rappresenta la factory di azioni relative agli asset;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative agli asset.

4.11.2.3 EdgeActionCreator

- **descrizione:** rappresenta la factory di azioni relative agli archi;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative agli archi.

4.11.2.4 NodeActionCreator

- **descrizione:** rappresenta la factory di azioni relative ai nodi;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative ai nodi.

4.11.2.5 OptionActionCreator

- **descrizione:** rappresenta la factory di azioni relative agli asset;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative agli asset.

4.11.2.6 ScenarioActionCreator

- **descrizione:** rappresenta la factory di azioni relative agli scenari di danno;
- **utilizzo:** i suoi metodi sono chiamati dalla View e dal CallManager per la creazione di azioni relative agli scenari di danno;
- **relazioni con altre classi:**
 - OUT ScenarioAction.

4.12 DeGeOP::ViewPkg

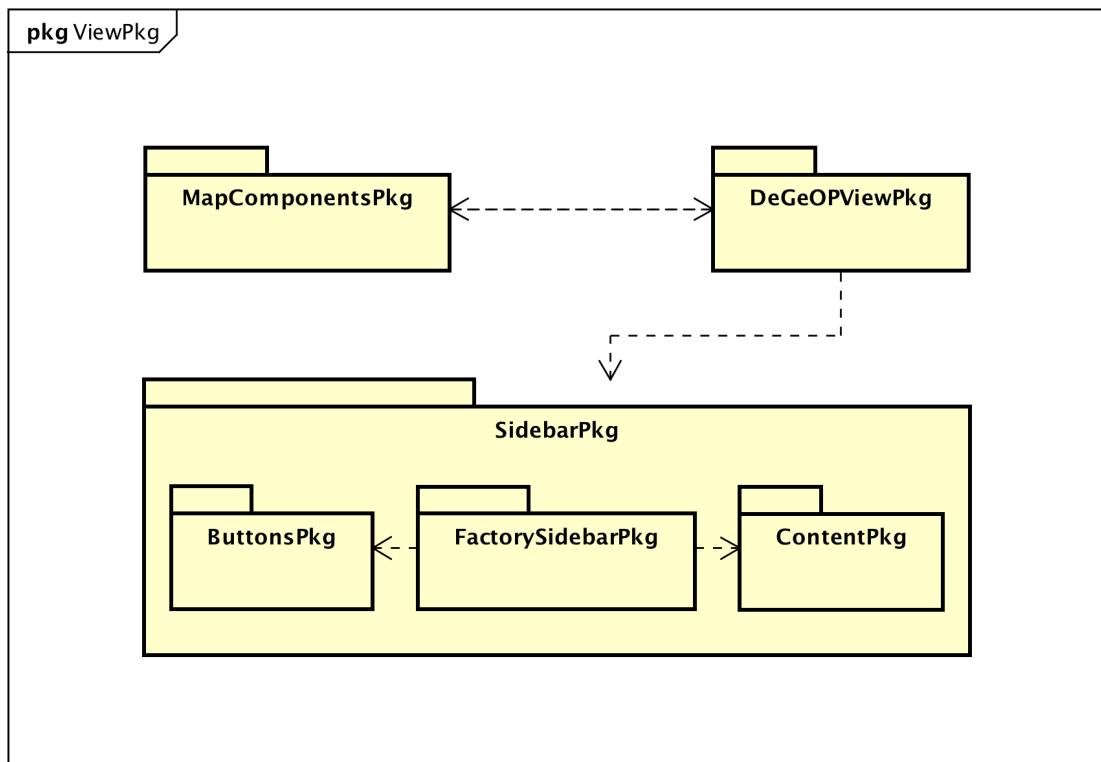


Figura 7: Schema componente DeGeOP::ViewPkg

4.12.1 Informazioni sul package

- **descrizione:** racchiude le componenti per la visualizzazione dell'interfaccia utente;
- **padre:** [DeGeOP](#);
- **package contenuti:**
 - `ViewPkg::DeGeOPViewPkg`;
 - `ViewPkg::MapComponentsPkg`;
 - `ViewPkg::SidebarPkg`.
- **interazioni con altri package:**
 - OUT ActionPkg: dispatch di azioni;
 - OUT Alexa voice service: gestore vocale;
 - OUT Hammer: gestione gesture ;
 - OUT Openlayers: gestione mappa;
 - OUT React: utilizzo componenti react;
 - OUT ReactToolbox: utilizzo componenti material design;
 - OUT Redux: utilizzo metodo dispatch;
 - OUT StorePkg: subscribe sullo store.

4.13 DeGeOP::ViewPkg::DeGeOPViewPkg

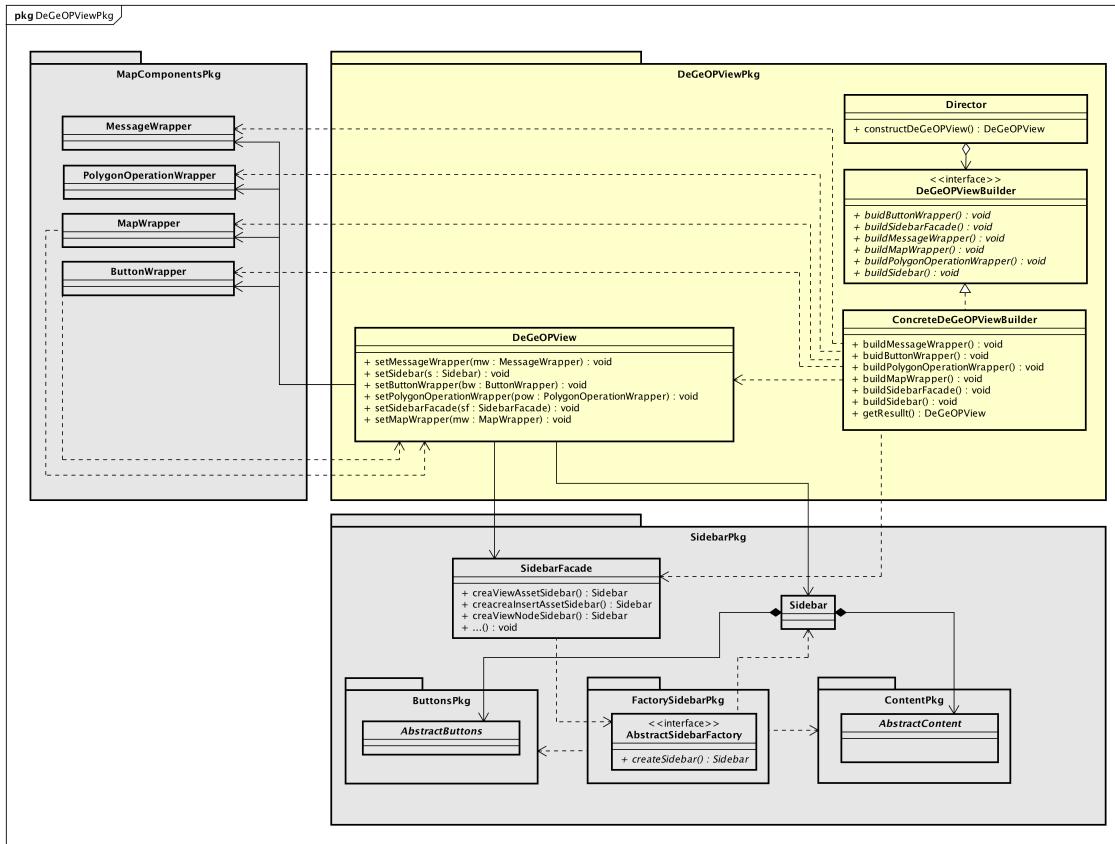


Figura 8: Schema componente DeGeOP::ViewPkg::DeGeOPViewPkg

4.13.1 Informazioni sul package

- descrizione:** racchiude la componente principale della view;
- padre:** [ViewPkg](#);
- interazioni con altri package:**
 - IN MapComponentsPkg: utilizzo di componenti grafiche;
 - OUT MapComponentsPkg: utilizzo di componenti grafiche;
 - OUT SidebarPkg: utilizzo della sidebar.
- classi contenute:**
 - **ConcreteDeGeOPViewBuilder**;
 - **DeGeOPView**;
 - **DeGeOPViewBuilder**;
 - **Director**.

4.13.2 Classi

4.13.2.1 ConcreteDeGeOPViewBuilder

- **descrizione:** rappresenta un'implementazione di DeGeOPViewBuilder;
- **utilizzo:** i suoi metodi vengono invocati dal director per creare le varie componenti di DeGeOPView;
- **relazioni con altre classi:**
 - OUT MapWrapper.

4.13.2.2 DeGeOPView

- **descrizione:** rappresenta l'oggetto grafico radice, che comprende l'intera View del prodotto;
- **utilizzo:** i suoi metodi setter sono richiamati da ConcreteDeGeOPViewBuilder per impostare i suoi campi dati. La classe effettua il subscribe sullo Store per ricevere gli aggiornamenti .

4.13.2.3 DeGeOPViewBuilder

- **descrizione:** interfaccia che rappresenta un builder per l'oggetto DeGeOPView;
- **utilizzo:** viene implementata per istanziare un builder concreto di DeGeOPView.

4.13.2.4 Director

- **descrizione:** rappresenta il director del design pattern builder usato per creare la View;
- **utilizzo:** è utilizzato per avviare la costruzione di DeGeOPView chiamando i metodi di un DeGeOPViewBuilder.

4.14 DeGeOP::ViewPkg::MapComponentsPkg

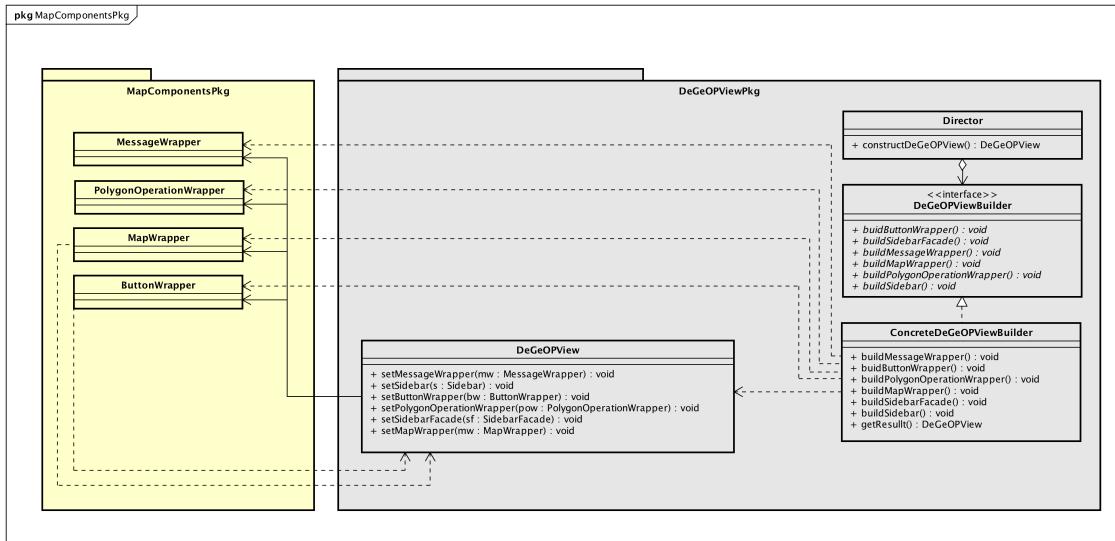


Figura 9: Schema componente DeGeOP::ViewPkg::MapComponentsPkg

4.14.1 Informazioni sul package

- **descrizione:** racchiude le componenti relative alla mappa e ai pulsanti sopra di essa;
- **padre:** [ViewPkg](#);
- **interazioni con altri package:**
 - IN DeGeOPViewPkg: utilizzo di componenti grafiche;
 - OUT DeGeOPViewPkg: utilizzo di componenti grafiche;
 - OUT Openlayers: gestione mappa.
- **classi contenute:**
 - ButtonWrapper;
 - MapWrapper;
 - MessageWrapper;
 - PolygonOperationWrapper.

4.14.2 Classi

4.14.2.1 ButtonWrapper

- **descrizione:** rappresenta una classe wrapper per visualizzare una serie di bottoni con cui è possibile eseguire varie operazioni;
- **utilizzo:** viene utilizzato per mostrare sulla mappa una serie di bottoni.

4.14.2.2 MapWrapper

- **descrizione:** rappresenta una classe wrapper per visualizzare la mappa;
- **utilizzo:** viene utilizzata per visualizzare una mappa e permettere all'utente di interagire con essa;
- **relazioni con altre classi:**
 - IN ConcreteDeGeOPViewBuilder.

4.14.2.3 MessageWrapper

- **descrizione:** rappresenta una classe wrapper per visualizzare un messaggio;
- **utilizzo:** viene utilizzato per mostrare messaggi di errore o di successo sulla mappa.

4.14.2.4 PolygonOperationWrapper

- **descrizione:** rappresenta una classe wrapper per visualizzare un bottone con cui è possibile effettuare operazioni sul perimetro di un poligono sulla mappa;
- **utilizzo:** invocando i suoi metodi è possibile iniziare a disegnare il poligono su mappa oppure cancellare l'ultimo segmento disegnato.

4.15 DeGeOP::ViewPkg::SidebarPkg

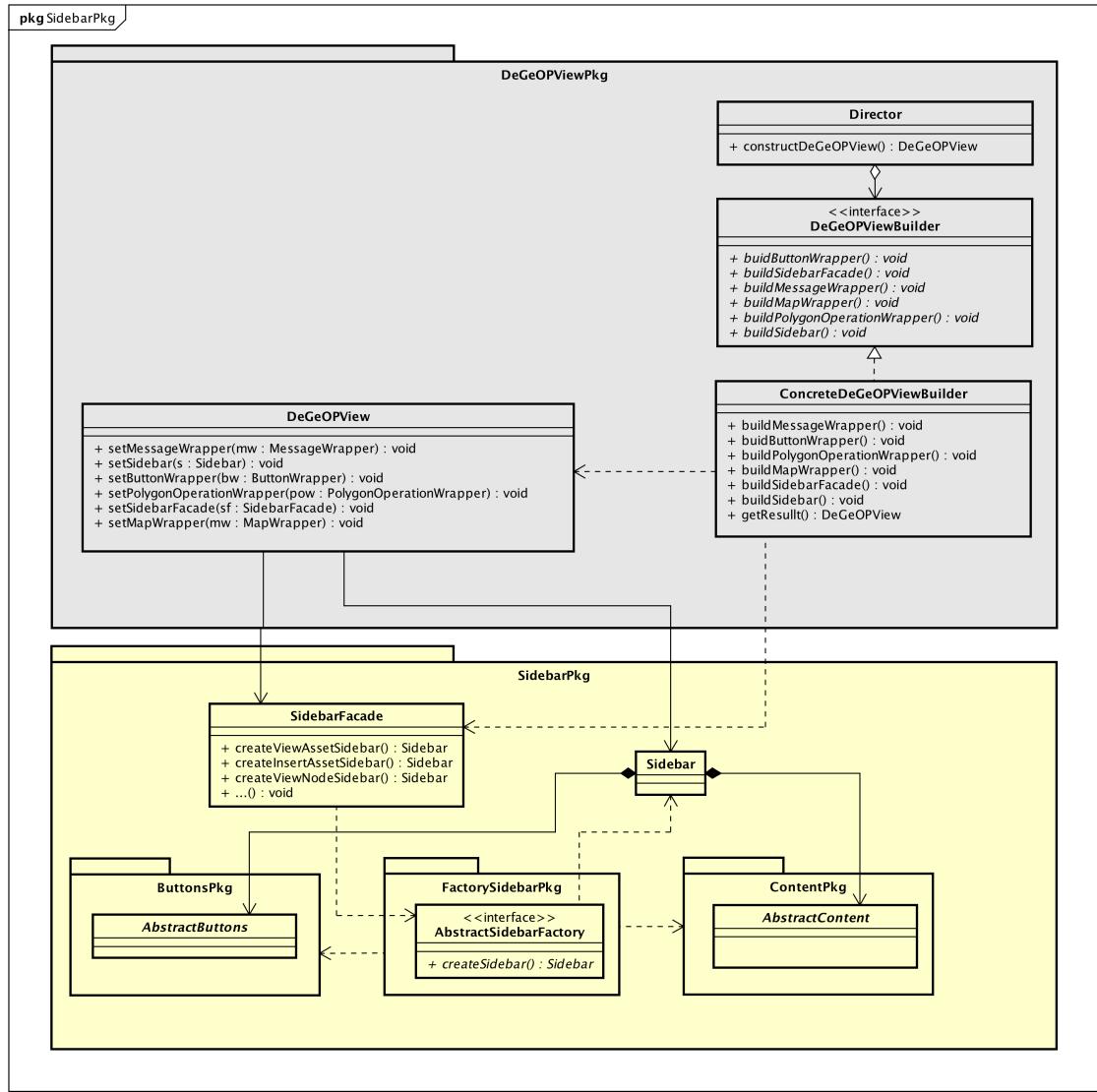


Figura 10: Schema componente DeGeOP::ViewPkg::SidebarPkg

4.15.1 Informazioni sul package

- **descrizione**: racchiude le componenti necessarie alla rappresentazione della sidebar;
- **padre**: [ViewPkg](#);
- **package contenuti**:
 - SidebarPkg::[ButtonsPkg](#);
 - SidebarPkg::[ContentPkg](#);
 - SidebarPkg::[FactorySidebarPkg](#).
- **interazioni con altri package**:
 - IN DeGeOPViewPkg: utilizzo della sidebar.

- **classi contenute:**

- Sidebar;
 - SidebarFacade.

4.15.2 Classi

4.15.2.1 Sidebar

- **descrizione:** rappresenta una sidebar contenente una serie di componenti grafiche;
- **utilizzo:** renderizza una sidebar composta da contenuto in cui l'utente può compilare i dati e da buttoni che permettono di eseguire varie operazioni, come ad esempio inserimenti e modifiche.

4.15.2.2 SidebarFacade

- **descrizione:** rappresenta un Facade tra DeGeOPView e Abstract;
- **utilizzo:** i suoi metodi vengono invocati per la creazione di Sidebar di vari tipologie senza che DeGeOPView necessiti di conoscere del tipo concreto delle Factory. Facade utilizza SidebarFactory concrete all'interno dei suoi metodi.

4.16 DeGeOP::ViewPkg::SidebarPkg::FactorySidebarPkg

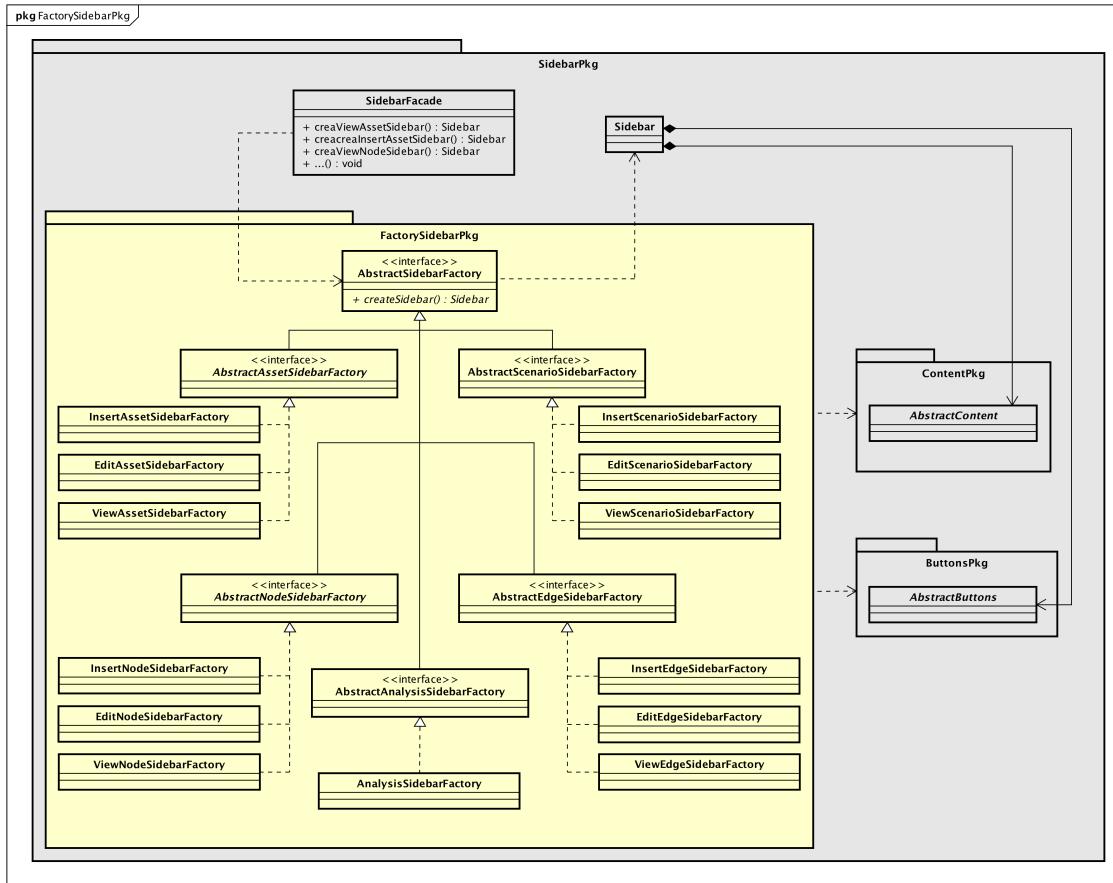


Figura 11: Schema componente DeGeOP::ViewPkg::SidebarPkg::FactorySidebarPkg

4.16.1 Informazioni sul package

- **descrizione:** Racchiude le componenti Factory per la sidebar;
- **padre:** [SidebarPkg](#);
- **interazioni con altri package:**
 - OUT ContentPkg: creazione contenuto della sidebar.
- **classi contenute:**
 - AbstractAnalysisSidebarFactory;
 - AbstractAssetSidebarFactory;
 - AbstractEdgeSidebarFactory;
 - AbstractNodeSidebarFactory;
 - AbstractScenarioSidebarFactory;
 - AbstractSidebarFactory;
 - AnalysisSidebarFactory;
 - EditAssetSidebarFactory;

- EditEdgeSidebarFactory;
- EditNodeSidebarFactory;
- EditScenarioSidebarFactory;
- InsertAssetSidebarFactory;
- InsertEdgeSidebarFactory;
- InsertNodeSidebarFactory;
- InsertScenarioSidebarFactory;
- ViewAssetSidebarFactory;
- ViewEdgeSidebarFactory;
- ViewNodeSidebarFactory;
- ViewScenarioSidebarFactory.

4.16.2 Classi

4.16.2.1 AbstractAnalysisSidebarFactory

- **descrizione:** abstract factory relativa alla sidebar analisi di danno;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractSidebar e le istanze di sidebar specifiche.

4.16.2.2 AbstractAssetSidebarFactory

- **descrizione:** abstract factory relativa alla sidebar asset;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractSidebar e le istanze di sidebar specifiche.

4.16.2.3 AbstractEdgeSidebarFactory

- **descrizione:** abstract factory relativa alla sidebar archi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractSidebar e le istanze di sidebar specifiche.

4.16.2.4 AbstractNodeSidebarFactory

- **descrizione:** abstract factory relativa alla sidebar nodi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractSidebar e le istanze di sidebar specifiche.

4.16.2.5 AbstractScenarioSidebarFactory

- **descrizione:** abstract factory relativa alla sidebar scenari;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractSidebar e le istanze di sidebar specifiche.

4.16.2.6 AbstractSidebarFactory

- **descrizione:** una classe d'interfaccia rappresentante una sidebar con il relativo contenuto;
- **utilizzo:** è utilizzata da SidebarFacade per la creazione di Sidebar.

4.16.2.7 AnalysisSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa alle analisi di danno;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar relativa alle analisi di danno.

4.16.2.8 EditAssetSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa alla modifica di un asset;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per la modifica di un asset.

4.16.2.9 EditEdgeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa alla modifica di un arco;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per la modifica di un arco.

4.16.2.10 EditNodeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di un nodo;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di un nodo.

4.16.2.11 EditScenarioSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di uno scenario di danno;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di uno scenario di danno.

4.16.2.12 InsertAssetSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di un asset;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di un asset.

4.16.2.13 InsertEdgeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di un arco;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di un arco.

4.16.2.14 InsertNodeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di un nodo;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di un nodo.

4.16.2.15 InsertScenarioSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di uno scenario di danno;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di uno scenario di danno.

4.16.2.16 ViewAssetSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa alla visualizzazione di un asset;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per la visualizzazione di un asset.

4.16.2.17 ViewEdgeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa alla visualizzazione di un arco;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per la visualizzazione di un arco.

4.16.2.18 ViewNodeSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di un nodo;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di un nodo.

4.16.2.19 ViewScenarioSidebarFactory

- **descrizione:** rappresenta una factory concreta di una sidebar relativa all'inserimento di uno scenario di danno;
- **utilizzo:** viene utilizzata da SidebarFacade per la creazione di Sidebar per l'inserimento di uno scenario di danno.

4.17 DeGeOP::ViewPkg::SidebarPkg::ContentPkg

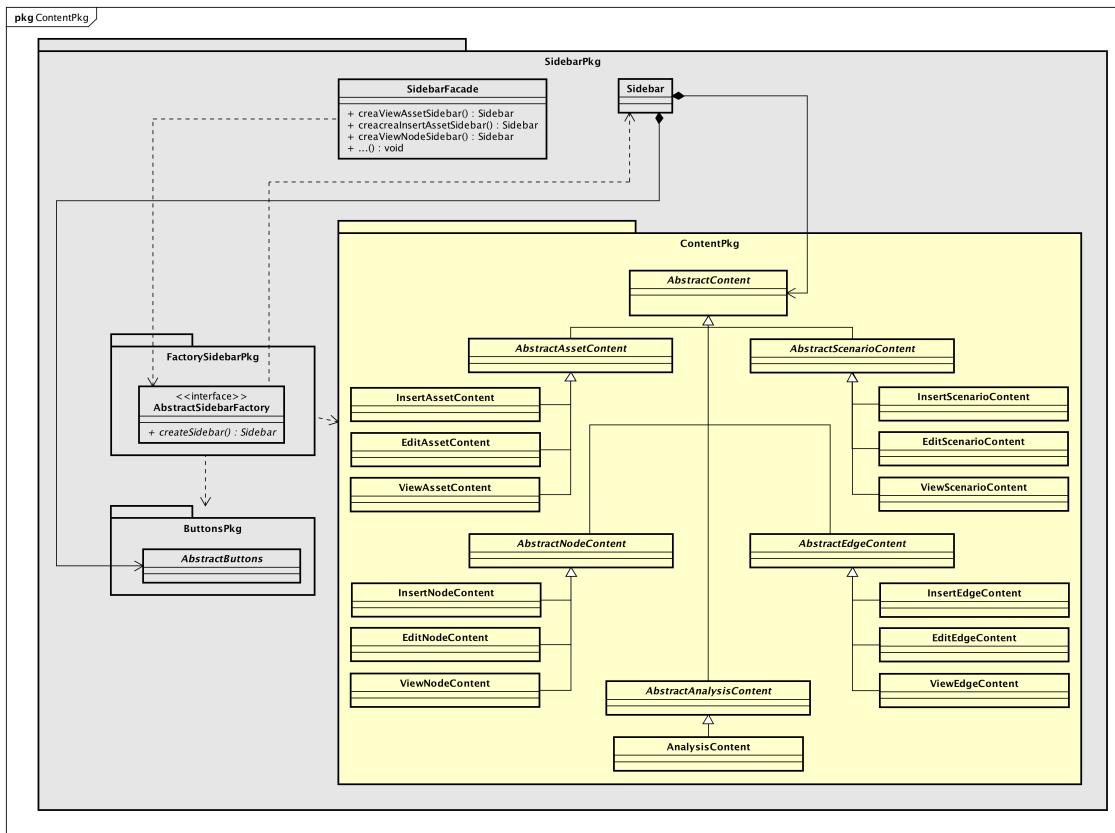


Figura 12: Schema componente DeGeOP::ViewPkg::SidebarPkg::ContentPkg

4.17.1 Informazioni sul package

- descrizione:** racchiude le componenti che sono relative all'area informativa della Sidebar;
- padre:** [SidebarPkg](#);
- interazioni con altri package:**
 - IN [FactorySidebarPkg](#): creazione contenuto della sidebar;
 - OUT [React-color](#): visualizzazione palette colori.
- classi contenute:**
 - `AbstractAnalysisContent`;
 - `AbstractAssetContent`;
 - `AbstractContent`;
 - `AbstractEdgeContent`;
 - `AbstractNodeContent`;
 - `AbstractScenarioContent`;
 - `AnalysisContent`;
 - `EditAssetContent`;

- EditEdgeContent;
- EditNodeContent;
- EditScenarioContent;
- InsertAssetContent;
- InsertEdgeContent;
- InsertNodeContent;
- InsertScenarioContent;
- ViewAssetContent;
- ViewEdgeContent;
- ViewNodeContent;
- ViewScenarioContent.

4.17.2 Classi

4.17.2.1 AbstractAnalysisContent

- **descrizione:** una classe astratta rappresentante il contenuto della sidebar durante le operazioni sulle analisi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractContent e le istanze di contenuti specifiche.

4.17.2.2 AbstractAssetContent

- **descrizione:** una classe astratta rappresentante il contenuto della sidebar durante le operazioni sugli asset;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractContent e le istanze di contenuti specifiche.

4.17.2.3 AbstractContent

- **descrizione:** una classe d'interfaccia rappresentante il contenuto dell'area informativa nella sidebar;
- **utilizzo:** viene riferita da sidebar in quanto è una delle sue componenti.

4.17.2.4 AbstractEdgeContent

- **descrizione:** una classe astratta rappresentante il contenuto della sidebar durante le operazioni sugli archi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractContent e le istanze di contenuti specifiche.

4.17.2.5 AbstractNodeContent

- **descrizione:** una classe astratta rappresentante il contenuto della sidebar durante le operazioni sui nodi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractContent e le istanze di contenuti specifiche.

4.17.2.6 AbstractScenarioContent

- **descrizione:** una classe astratta rappresentante il contenuto della sidebar durante le operazioni sugli scenari;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractContent e le istanze di contenuti specifiche.

4.17.2.7 AnalysisContent

- **descrizione:** rappresenta il contenuto della sidebar relativa all'analisi di danno;
- **utilizzo:** viene creata da AnalysisSidebarFactory .

4.17.2.8 EditAssetContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla modifica di un asset;
- **utilizzo:** viene creata da EditAssetSidebarFactory .

4.17.2.9 EditEdgeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla modifica di un arco;
- **utilizzo:** viene creata da EditEdgeSidebarFactory .

4.17.2.10 EditNodeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla modifica di un nodo;
- **utilizzo:** viene creata da EditNodeSidebarFactory .

4.17.2.11 EditScenarioContent

- **descrizione:** rappresenta il contenuto della Sidebar relativa alla modifica di uno scenario di danno;
- **utilizzo:** viene creata da EditScenarioSidebarFactory .

4.17.2.12 InsertAssetContent

- **descrizione:** rappresenta il contenuto della sidebar relativa all'inserimento di un asset;
- **utilizzo:** viene creata da InsertAssetSidebarFactory .

4.17.2.13 InsertEdgeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa all'inserimento di un arco;
- **utilizzo:** viene creata da InsertEdgeSidebarFactory .

4.17.2.14 InsertNodeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa all'inserimento di un nodo;
- **utilizzo:** viene creata da InsertNodeSidebarFactory .

4.17.2.15 InsertScenarioContent

- **descrizione:** rappresenta il contenuto della Sidebar relativa all'inserimento di uno scenario di danno;
- **utilizzo:** viene creata da InsertScenarioSidebarFactory .

4.17.2.16 ViewAssetContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla visualizzazione di un asset;
- **utilizzo:** viene creata da ViewAssetSidebarFactory .

4.17.2.17 ViewEdgeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla visualizzazione di un arco;
- **utilizzo:** viene creata da ViewEdgeSidebarFactory .

4.17.2.18 ViewNodeContent

- **descrizione:** rappresenta il contenuto della sidebar relativa alla visualizzazione di un nodo;
- **utilizzo:** viene creata da ViewNodeSidebarFactory .

4.17.2.19 ViewScenarioContent

- **descrizione:** rappresenta il contenuto della Sidebar relativa alla visualizzazione di uno scenario di danno;
- **utilizzo:** viene creata da ViewScenarioSidebarFactory .

4.18 DeGeOP::ViewPkg::SidebarPkg::ButtonsPkg

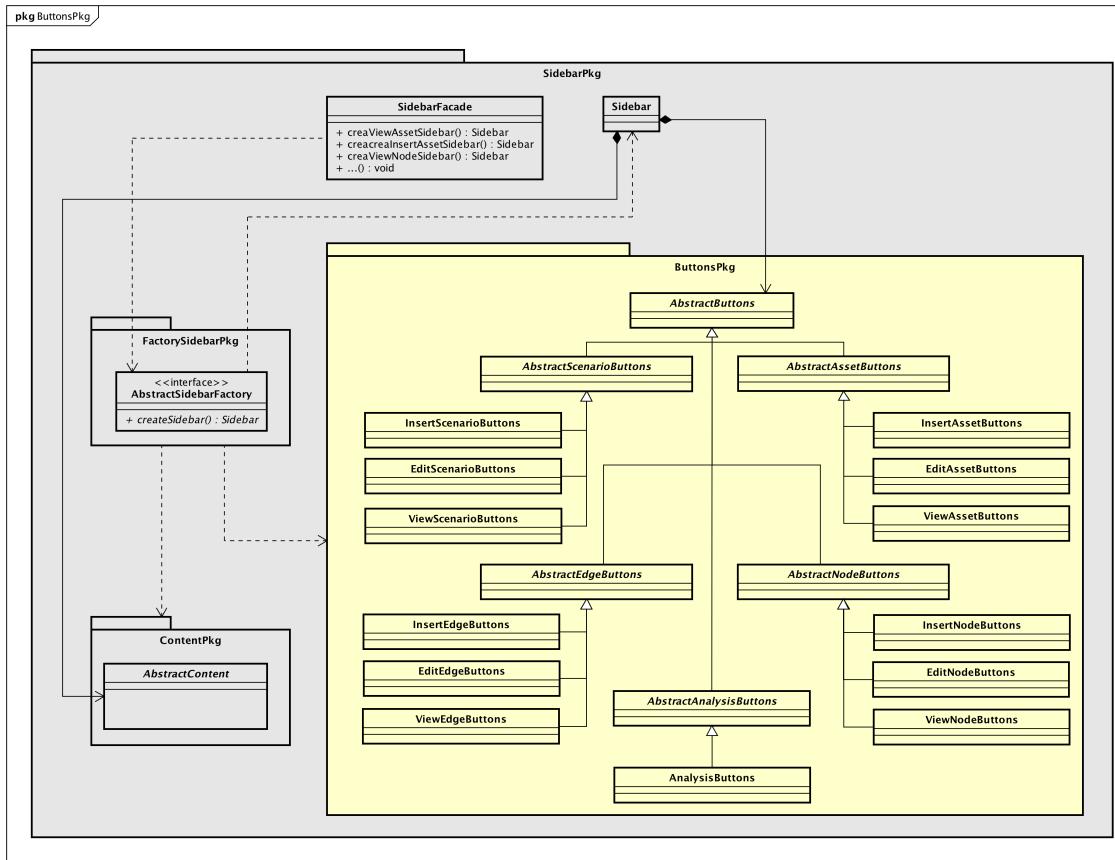


Figura 13: Schema componente DeGeOP::ViewPkg::SidebarPkg::ButtonsPkg

4.18.1 Informazioni sul package

- descrizione:** racchiude le componenti che sono relative all'area con i bottoni della Sidebar;
- padre:** [SidebarPkg](#);
- classi contenute:**
 - AbstractAnalysisButtons;
 - AbstractAssetButtons;
 - AbstractButtons;
 - AbstractEdgeButtons;
 - AbstractNodeButtons;
 - AbstractScenarioButtons;
 - AnalysisButtons;
 - EditAssetButtons;
 - EditEdgeButtons;
 - EditNodeButtons;

- EditScenarioButtons;
- InsertAssetButtons;
- InsertEdgeButtons;
- InsertNodeButtons;
- InsertScenarioButtons;
- ViewAssetButtons;
- ViewEdgeButtons;
- ViewNodeButtons;
- ViewScenarioButtons.

4.18.2 Classi

4.18.2.1 AbstractAnalysisButtons

- **descrizione:** una classe astratta rappresentante i bottoni della sidebar durante le operazioni sulle analisi;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractButtons e le istanze di bottoni specifiche.

4.18.2.2 AbstractAssetButtons

- **descrizione:** una classe astratta rappresentante i bottoni della sidebar durante le operazioni sugli asset;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractButtons e le istanze di bottoni specifiche.

4.18.2.3 AbstractButtons

- **descrizione:** una classe d'interfaccia rappresentante i bottoni inseriti nella sidebar;
- **utilizzo:** viene riferita da sidebar in quanto è una delle sue componenti.

4.18.2.4 AbstractEdgeButtons

- **descrizione:** una classe astratta rappresentante i bottoni della sidebar durante le operazioni sugli archi ;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractButtons e le istanze di bottoni specifiche.

4.18.2.5 AbstractNodeButtons

- **descrizione:** una classe astratta rappresentante i bottoni della sidebar durante le operazioni sui nodi ;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractButtons e le istanze di bottoni specifiche.

4.18.2.6 AbstractScenarioButtons

- **descrizione:** una classe astratta rappresentante i bottoni della sidebar durante le operazioni sugli scenari;
- **utilizzo:** viene usata come interfaccia di specializzazione fra abstractButtons e le istanze di bottoni specifiche.

4.18.2.7 AnalysisButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi all'analisi di danno;
- **utilizzo:** viene creata da AnalysisFactory.

4.18.2.8 EditAssetButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla modifica di un asset;
- **utilizzo:** viene creata da EditAssetFactory.

4.18.2.9 EditEdgeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla modifica di un arco;
- **utilizzo:** viene creata da EditEdgeFactory.

4.18.2.10 EditNodeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla modifica di un nodo;
- **utilizzo:** viene creata da EditNodeFactory.

4.18.2.11 EditScenarioButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla modifica di uno scenario di danno;
- **utilizzo:** viene creata da EditScenarioFactory.

4.18.2.12 InsertAssetButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi all'inserimento di un asset;
- **utilizzo:** viene creata da InsertAssetFactory.

4.18.2.13 InsertEdgeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi all'inserimento di un arco;
- **utilizzo:** viene creata da InsertEdgeFactory.

4.18.2.14 InsertNodeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi all'inserimento di un nodo;
- **utilizzo:** viene creata da InsertNodeFactory.

4.18.2.15 InsertScenarioButtons

- **descrizione:** rappresenta i bottoni della sidebar relativa all'inserimento di uno scenario di danno;
- **utilizzo:** viene creata da InsertScenarioFactory.

4.18.2.16 ViewAssetButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla visualizzazione di un asset;
- **utilizzo:** viene creata da ViewAssetFactory.

4.18.2.17 ViewEdgeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla visualizzazione di un arco;
- **utilizzo:** viene creata da ViewEdgeFactory.

4.18.2.18 ViewNodeButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla visualizzazione di un nodo;
- **utilizzo:** viene creata da ViewNodeFactory.

4.18.2.19 ViewScenarioButtons

- **descrizione:** rappresenta i bottoni della sidebar relativi alla visualizzazione di uno scenario di danno;
- **utilizzo:** viene creata da ViewScenarioFactory.

5 Stime di fattibilità e bisogno di risorse

L'architettura definita precedentemente ha raggiunto un livello di dettaglio sufficiente a fornire una stima sulla fattibilità e di bisogno di risorse. Durante la progettazione iniziale del prodotto e la scelta delle tecnologie da utilizzare, si sono riscontrati diversi potenziali limiti legati agli stessi. Il [gruppoG](#) si è impegnato a coprire le parti carenti come descritto nei paragrafi sottostanti, al fine di rendere le tecnologie completamente adeguate per la realizzazione del prodotto.

5.1 JavaScript

L'utilizzo ottimale di [librerieG](#) come [ReactG](#) e Redux prevedono l'uso di costrutti e sintassi presenti solo dalla versione ES6 di [JavaScriptG](#) in poi. Tale versione non è ancora totalmente definita e quindi molti browser odierni non supportano nativamente alcune feature. Specificatamente ES6 ha inserito la sintassi relativa al costrutto delle classi (class, construct, ecc.) e alcuni operatori utili all'implementazione dei [reducerG](#) in Redux (per esempio lo spread operator). Questa situazione ha portato ad una serie di conseguenze, fra le quali:

- utilizzo di Babel, una [componenteG](#) JavaScript che agisce come compilatore (o meglio, come un refactor di codice) trasformando codice, che utilizza le feature definite in ES6 o superiore, in codice completamente compatibile alla versione ES5, che quindi è supportato nativamente dai browser moderni;
- bassa presenza di codice e librerie scritte in ES6. Per esempio la libreria OpenLayer e la sua documentazione sono scritte in codice ES5. Ciò si ripercuote sul prodotto da definire in due modi:
 - necessità di convertire il codice ES5 in codice ES6 per mantenere uniforme il codice prodotto;
 - necessità di importare librerie secondarie che operano tale conversione liberando il team dalla scrittura di ulteriore codice.

Il [packageG](#) Babel è una componente popolare e molto usata quindi il rischio che non funzioni correttamente è molto basso, ma comunque da non trascurare.

5.2 React

La libreria React è relativamente giovane, ma risulta molto usata. Risulta essere soprattutto una libreria stabile, anche a causa del numero basso di [issueG](#) correttamente aperti sulla relativa pagina [GitHubG](#).

Il requisito di funzionamento del prodotto su dispositivi mobile, nello specifico su tablet, implica che il prodotto stesso dovrebbe risultare di piccole dimensioni e computazionalmente leggero, per favorire un uso veloce e fluido anche su questa famiglia di dispositivi. React risulta una libreria adatta allo scopo in quanto leggera e pienamente supportata su dispositivi mobili.

6 Attività

6.1 Introduzione

Questa sezione descriverà le operazioni che l'utente può svolgere all'interno di *DeGeOP*. Per ogni operazione viene fornita:

- una descrizione;
- un diagramma di attività;
- uno o più eventuali mockup.

I mockup hanno lo scopo di chiarificare la struttura della single-page *DeGeOP*. Il *gruppoG* si riserva di poter modificare alcuni elementi grafici quando il prodotto sarà progettato nel dettaglio.

Le operazioni più complesse nei diagrammi di attività presentano uno sfondo di colore azzurro e sono ulteriormente descritte da sotto-diagrammi.

Di seguito vengono mostrati due mockup che evidenziano come verrà organizzata l'interfaccia grafica nel progetto *DeGeOP*.

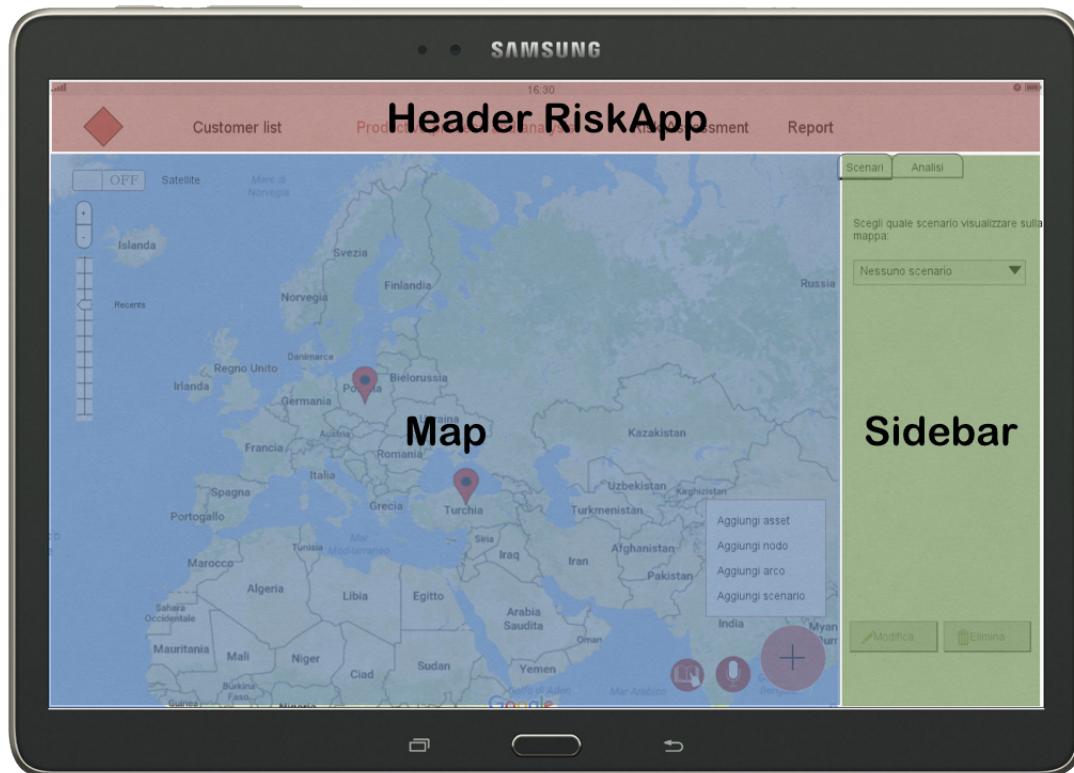


Figura 14: Mockup interfaccia grafica ad alto livello



Figura 15: Mockup interfaccia grafica basso livello

6.2 Visualizzazione di default

DeGeOP sarà sviluppato come una single-page e tutte le operazioni che l'utente può svolgere sono attuabili a partire dalla visualizzazione di default (si veda il relativo mockup). L'utente, dopo aver acceduto a *DeGeOP*, potrà:

- interagire ripetutamente con la mappa:
 - aumentando/diminuendo il livello di zoom;
 - spostandosi sulla mappa;
 - attivando/disattivando la vista satellitare.
- svolgere ripetutamente una o più tra le seguenti operazioni:
 - selezionare un *asset_G*, *nodo_G*, *arco_G*, scenario;
 - aggiungere asset, nodi, archi, scenari;
 - gestire le analisi;
 - avviare il tutorial o l'assistente vocale.

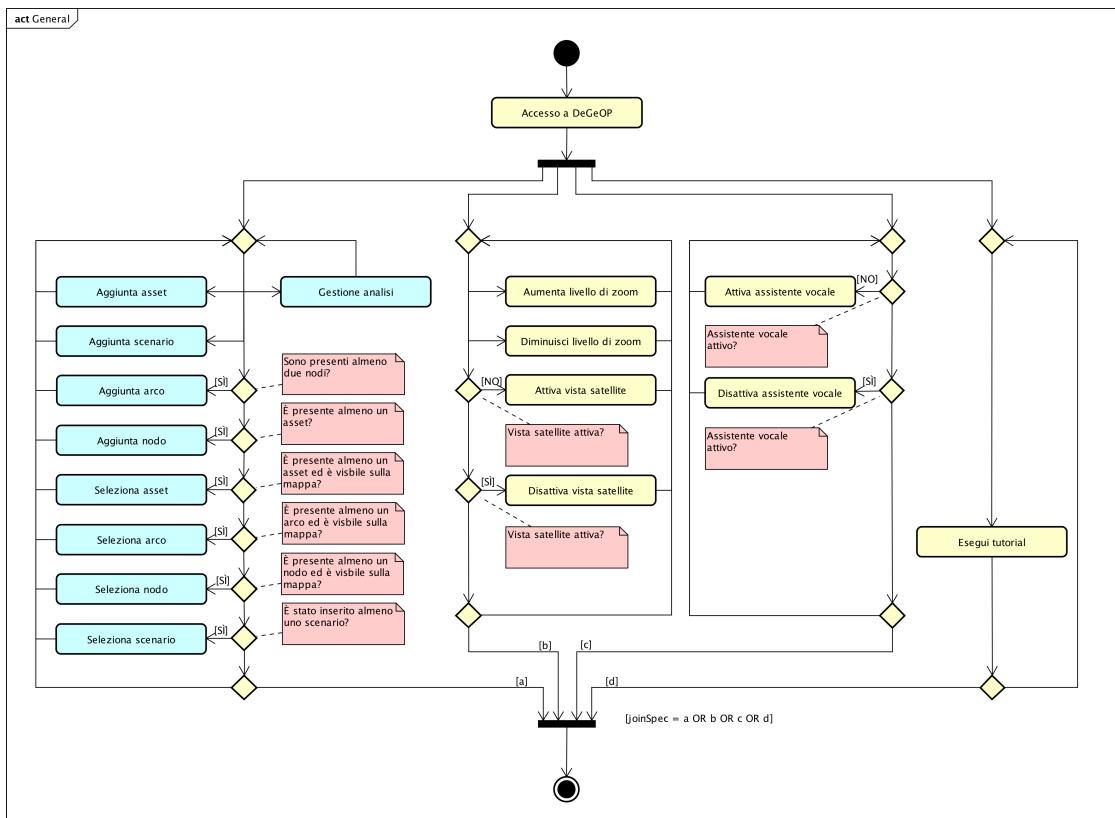


Figura 16: Diagramma delle attività generali

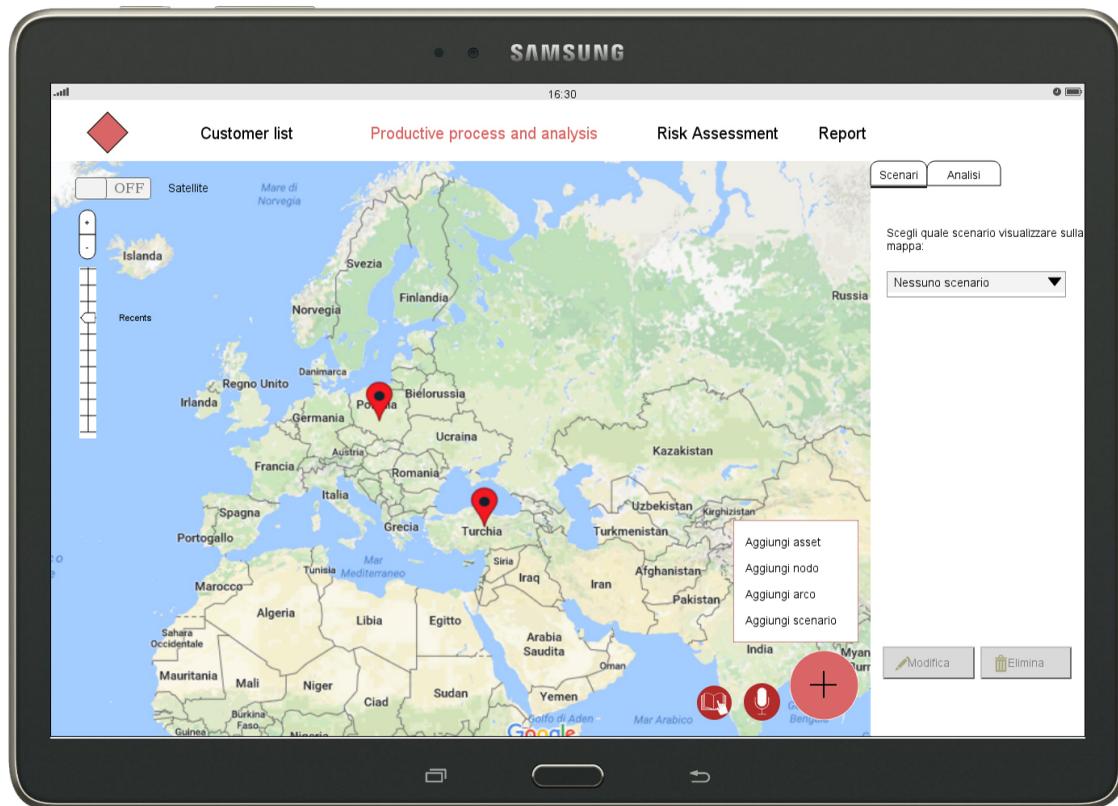


Figura 17: Mockup per la visualizzazione di default

6.3 Aggiunta asset

Per aggiungere un asset l'utente dovrà disegnare il perimetro dell'asset sulla mappa, compilarne i dati e confermare l'inserimento. In caso di dati non corretti, l'inserimento potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare l'inserimento. Il sistema richiede una conferma per portare a termine tale operazione.

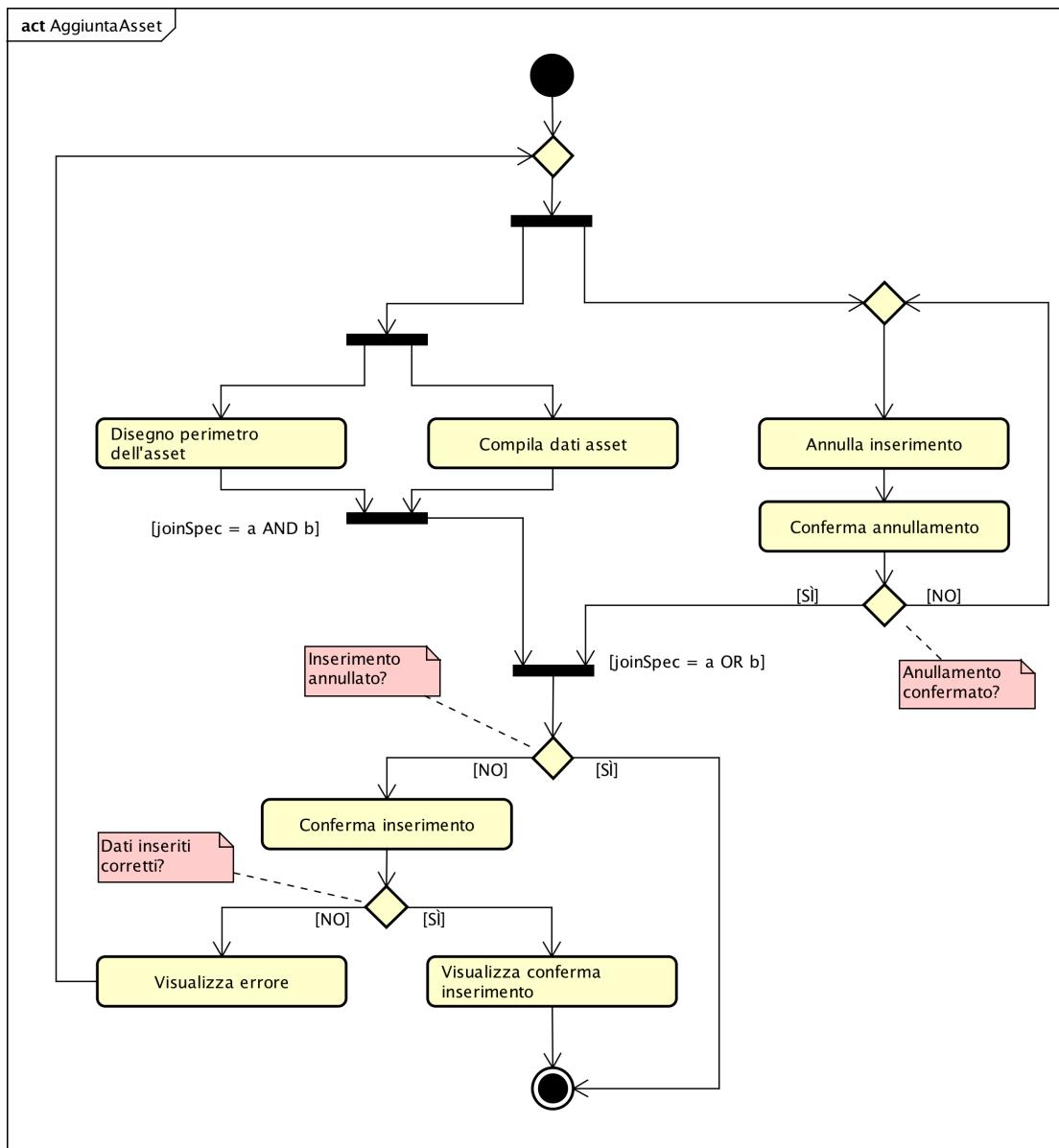


Figura 18: Diagramma di attività per l'aggiunta di un asset

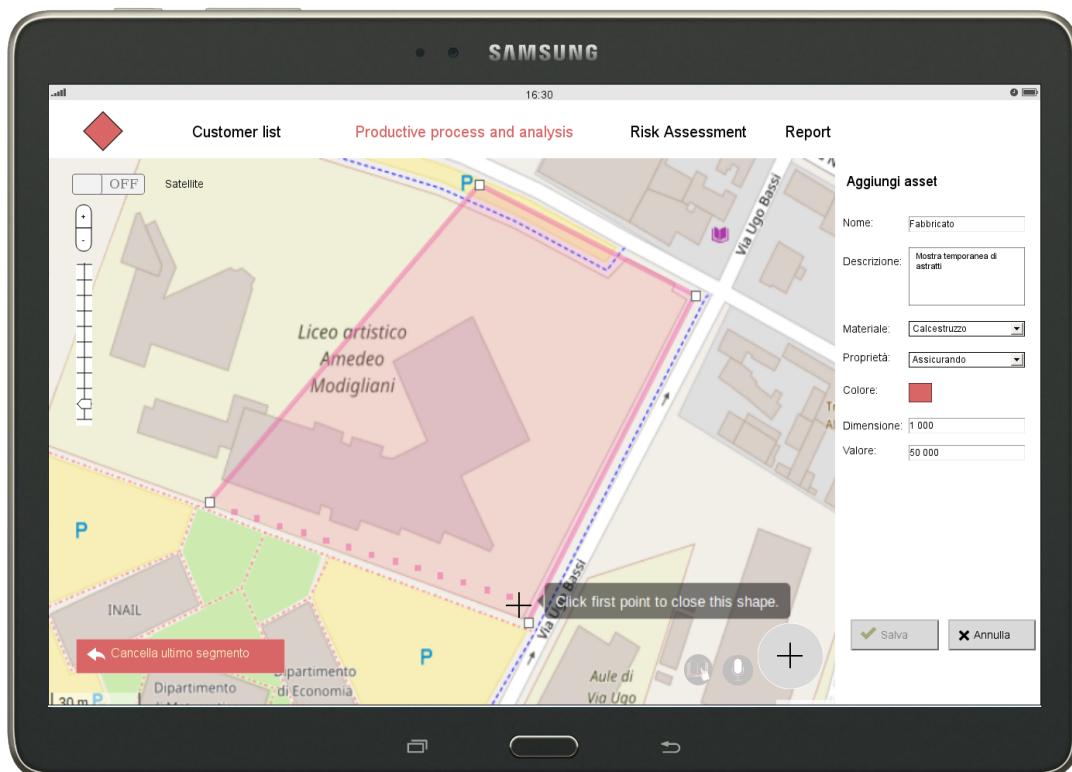


Figura 19: Mockup per l'aggiunta dell'asset

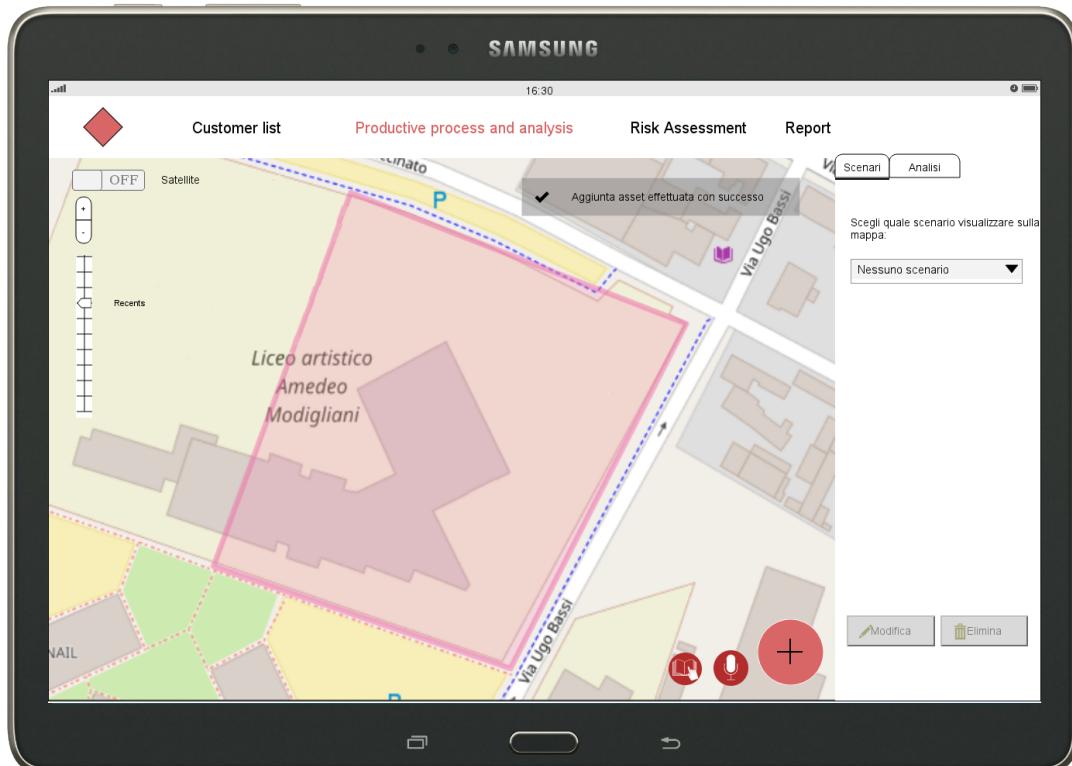


Figura 20: Mockup per il successo dell'operazione di aggiunta asset

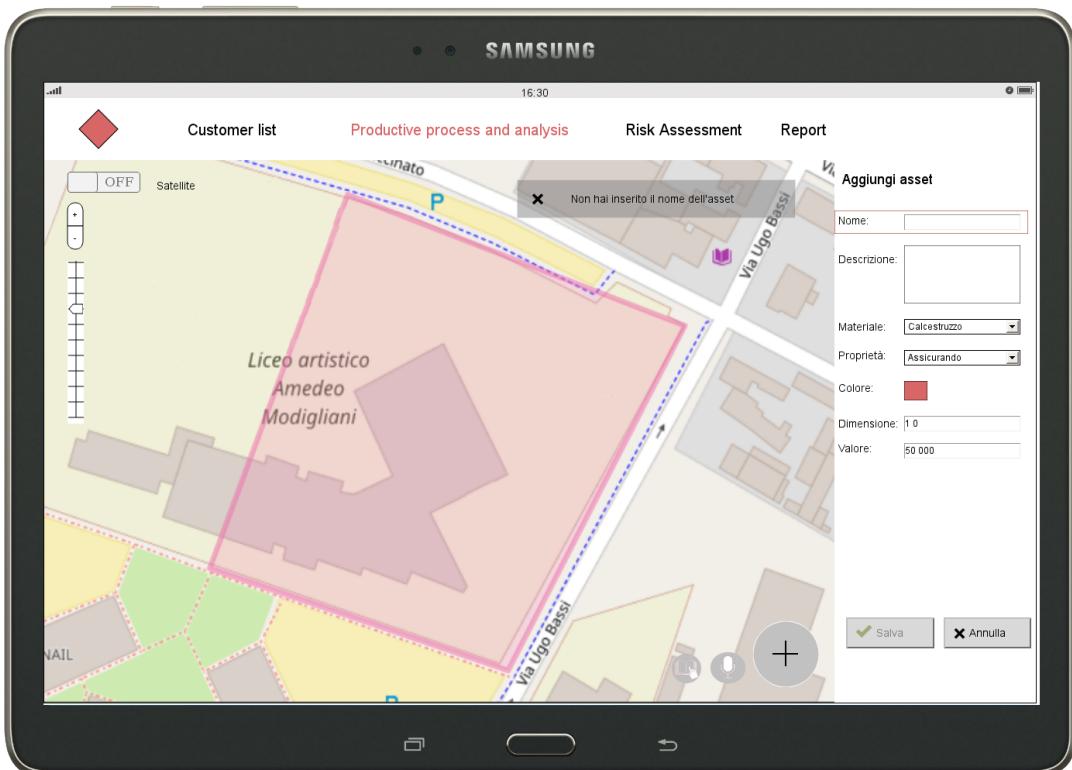


Figura 21: Mockup per l'errore durante l'operazione di aggiunta asset

6.4 Aggiunta nodo

Per aggiungere un nodo, l'utente dovrà posizionare il nodo all'interno dell'asset di appartenenza sulla mappa, compilare i dati e confermare l'inserimento. In caso di dati non corretti, l'inserimento potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare l'inserimento. Il sistema richiede una conferma per portare a termine tale operazione.

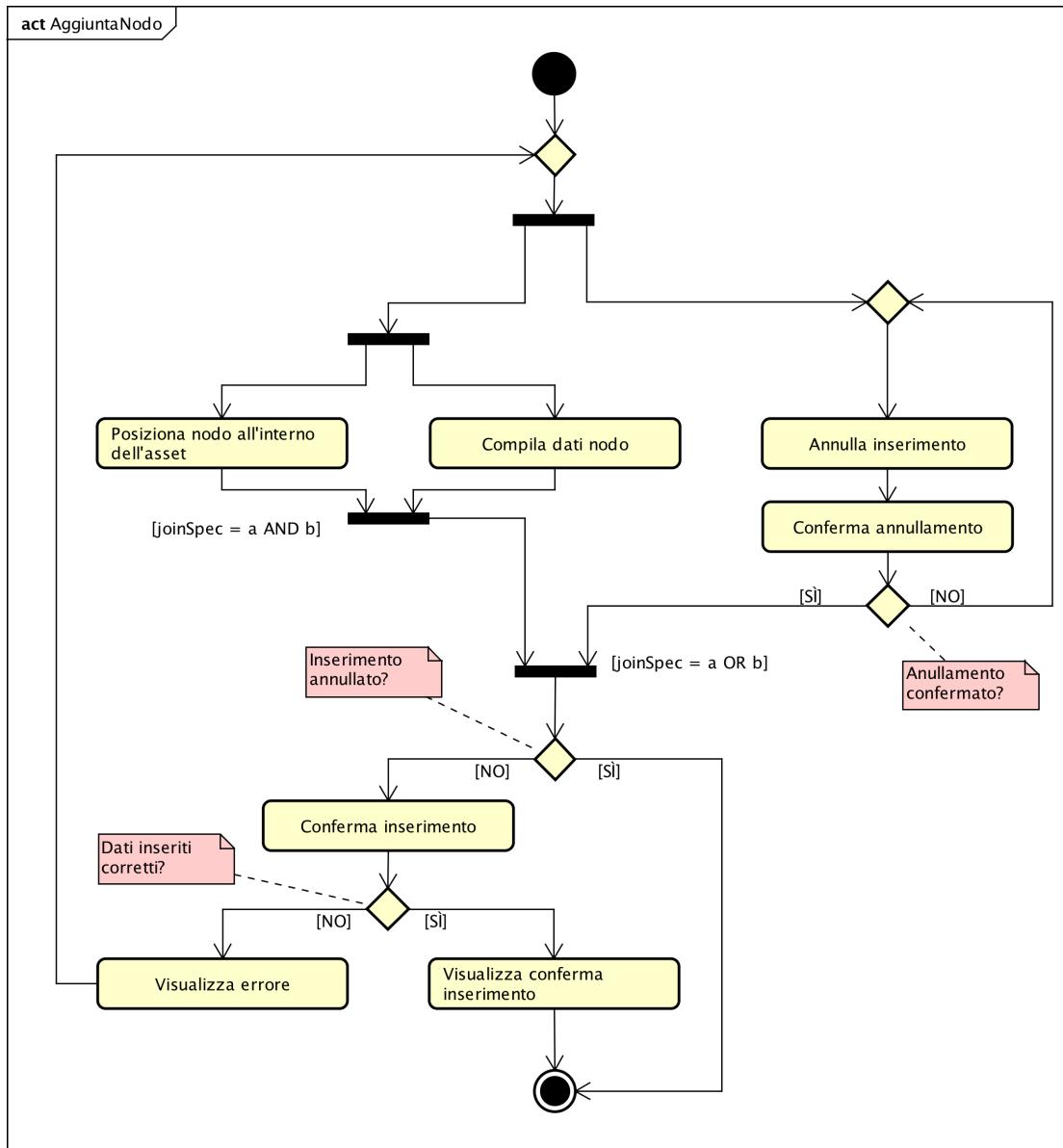


Figura 22: Diagramma di attività per l'aggiunta di un nodo

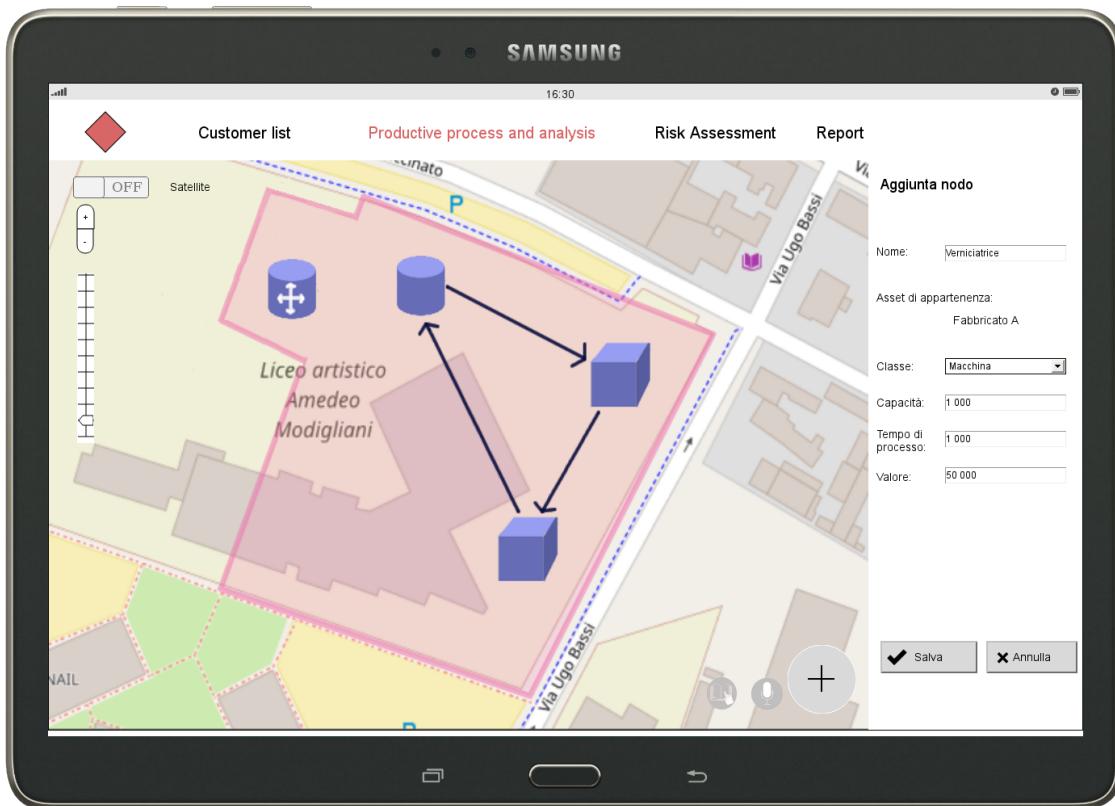


Figura 23: Mockup per l'aggiunta di un nodo

6.5 Aggiunta arco

Per aggiungere un arco, l'utente dovrà disegnare l'arco sulla mappa, compilarne i dati e confermare l'inserimento. In caso di dati non corretti, l'inserimento potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare l'inserimento. Il sistema richiede una conferma per portare a termine tale operazione.

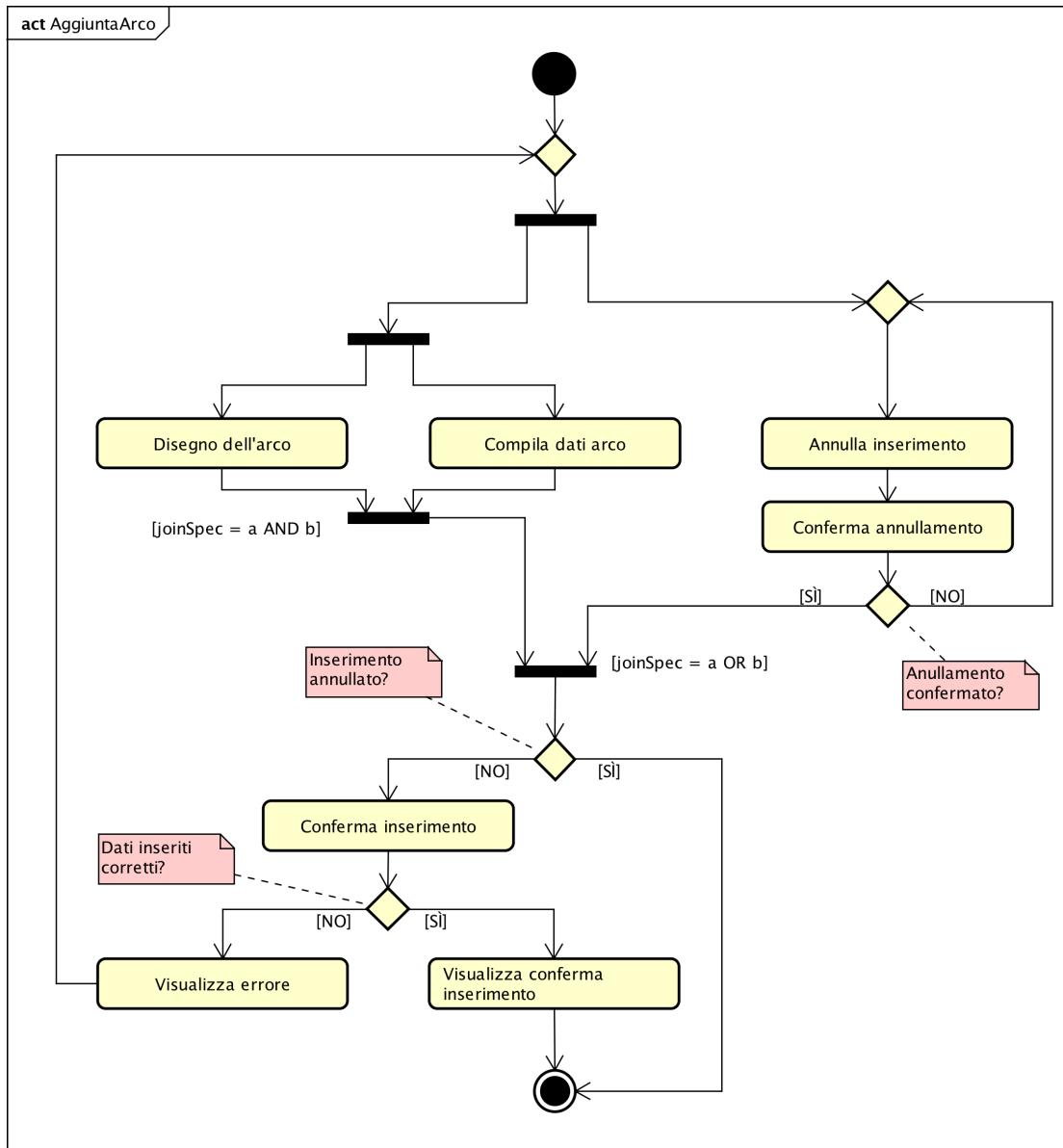


Figura 24: Diagramma di attività per l'aggiunta di un arco



Figura 25: Mockup per l'aggiunta di un arco

6.6 Aggiunta scenario

Per aggiungere uno scenario l'utente dovrà disegnare il perimetro dello scenario sulla mappa, compilare i dati e confermare l'inserimento. In caso di dati non corretti, l'inserimento potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare l'inserimento. Il sistema richiede una conferma per portare a termine tale operazione.

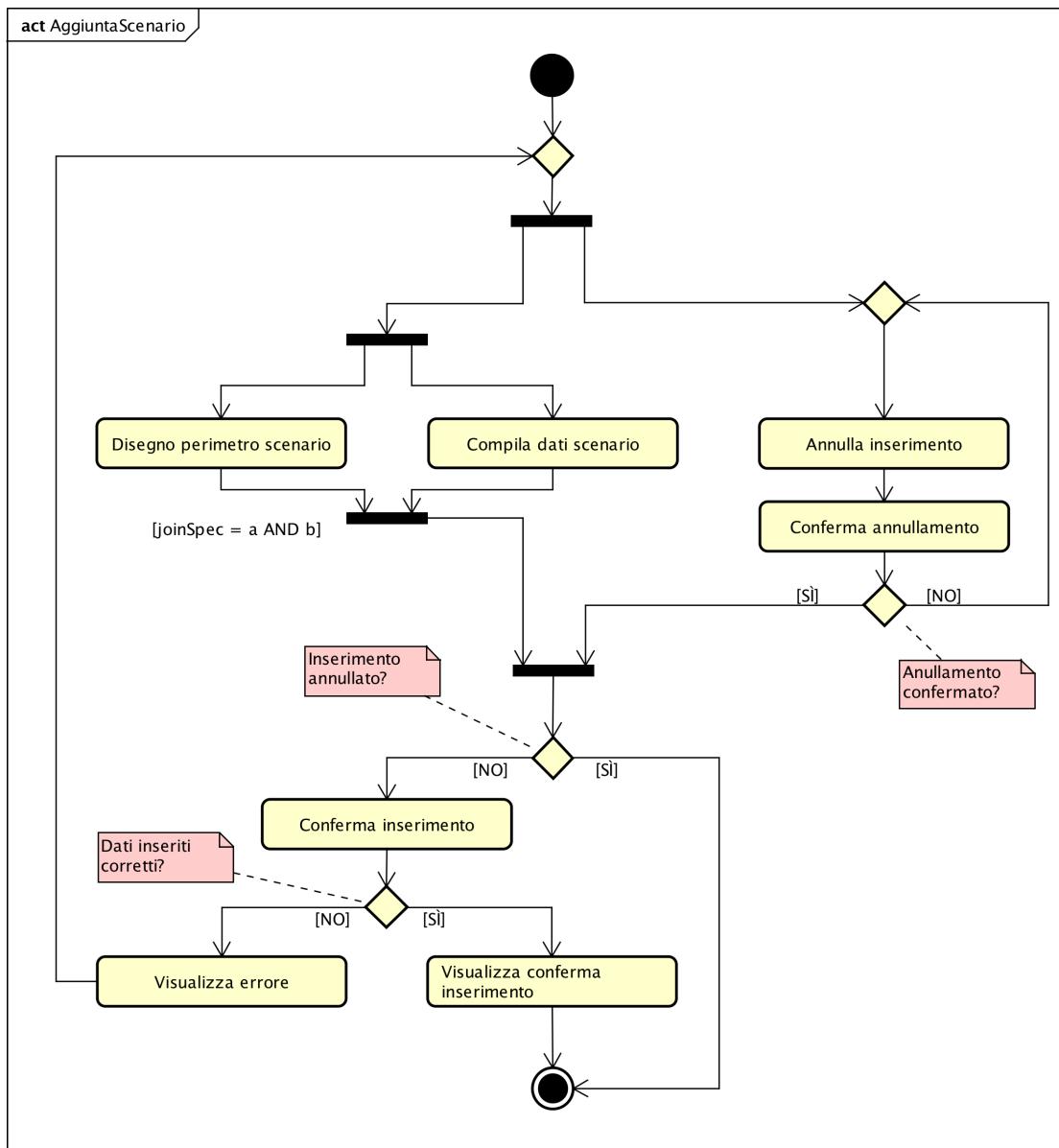


Figura 26: Diagramma di attività per l'aggiunta di uno scenario



Figura 27: Mockup per l'aggiunta di uno scenario

6.7 Gestione Analisi

Dopo essere entrati nella sezione di analisi, l'utente può effettuare ripetutamente una tra le seguenti operazioni:

- è possibile aggiungere scenari su cui non è ancora stata calcolata alla lista di quelli da analizzare. In seguito è possibile avviare l'analisi;
- è possibile eliminare scenari dalla lista degli scenari su cui è stata calcolata l'analisi. In questo caso l'utente è invitato a confermare l'eliminazione o ad annullarla;
- l'utente non può effettuare alcuna operazione all'interno della sezione di analisi, se non sono presenti scenari.

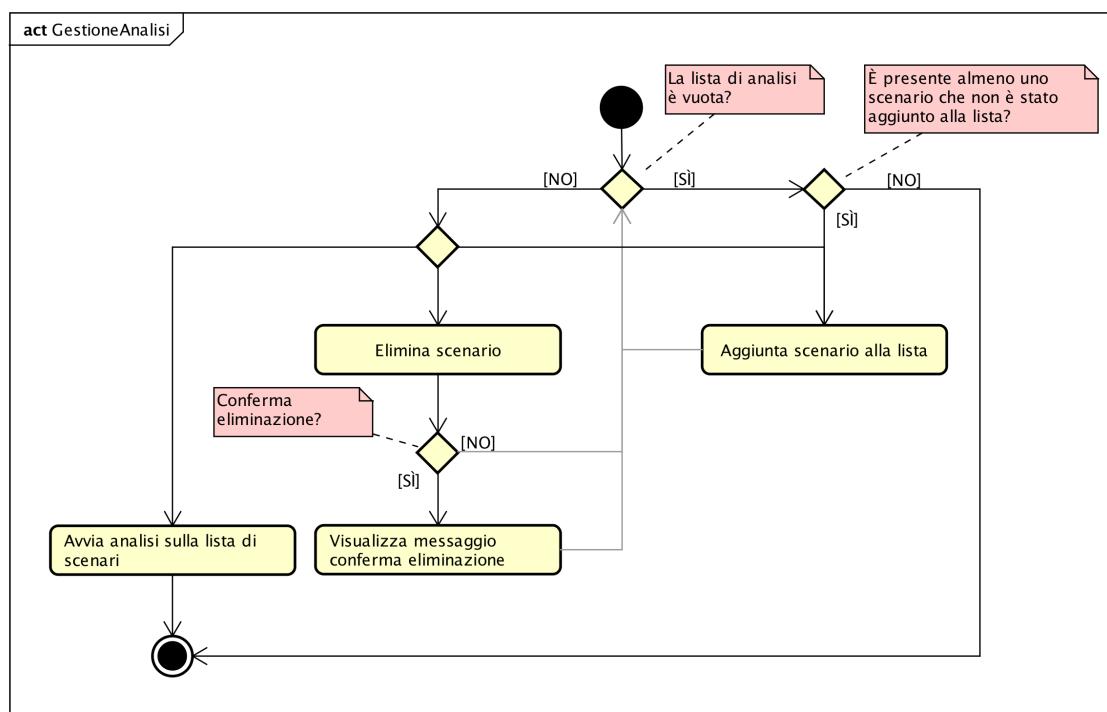


Figura 28: Diagramma di attività per la gestione analisi

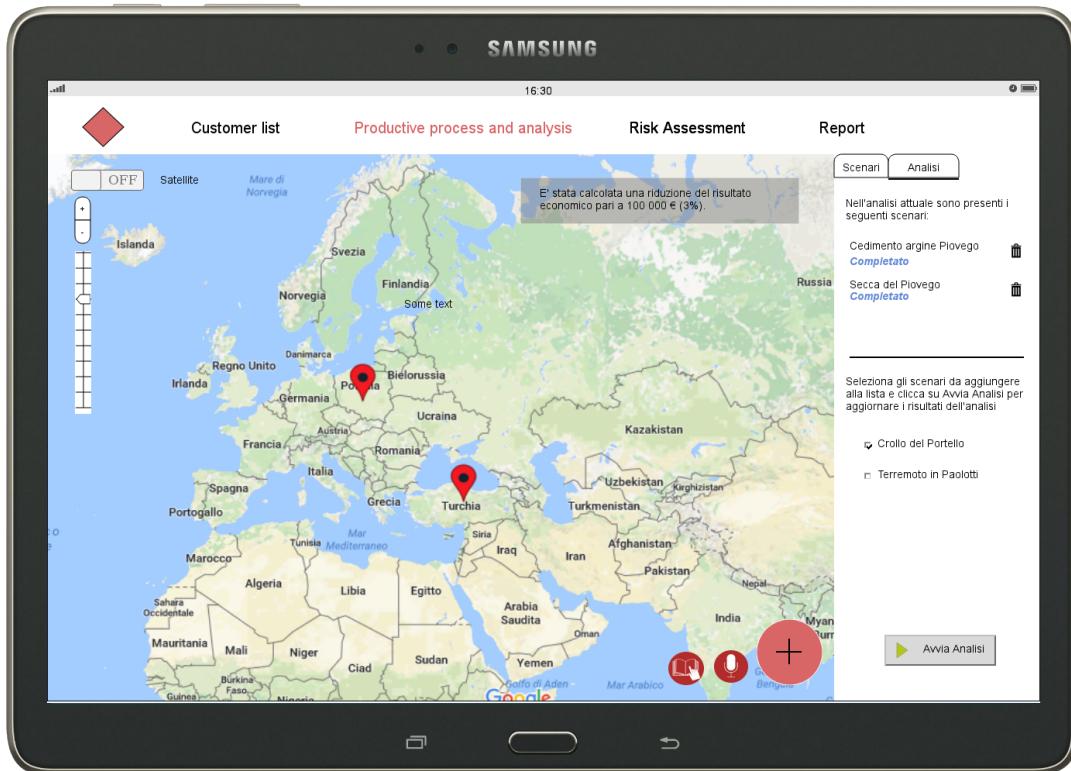


Figura 29: Mockup per l'avvio dell'analisi

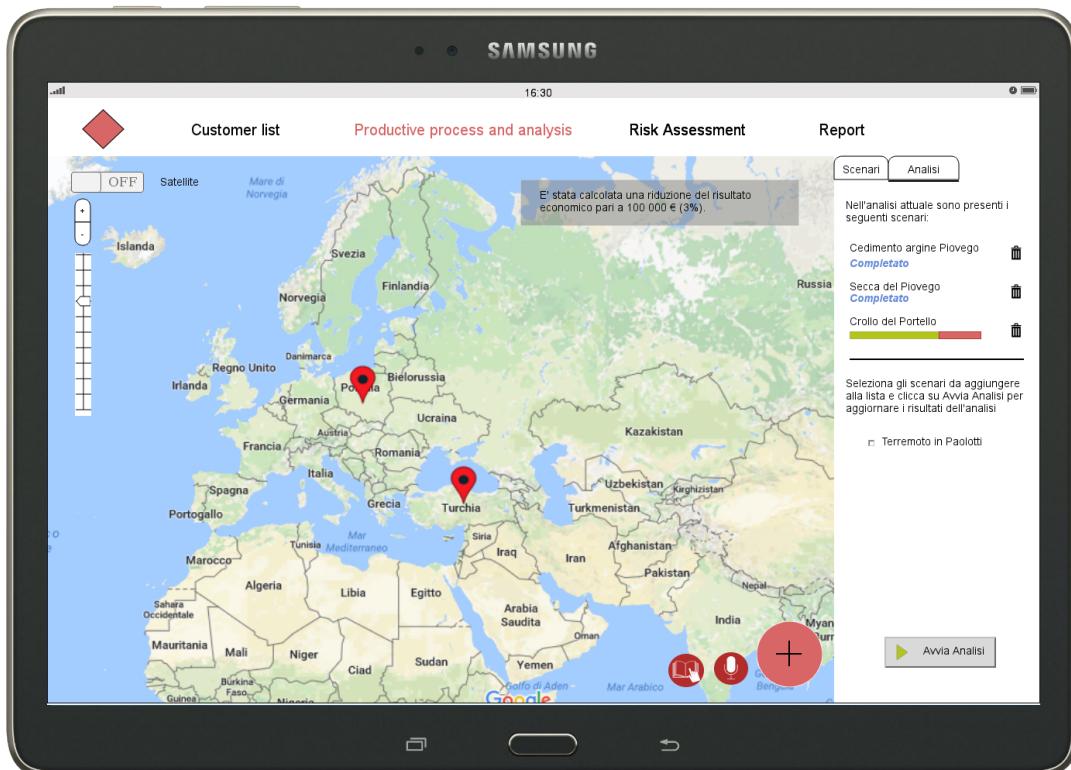


Figura 30: Mockup per l'analisi in corso

6.8 Selezione asset

Una volta selezionato l'asset l'utente potrà:

- centrarlo automaticamente sulla mappa (utile nel caso in cui nel frattempo l'utente si sia spostato sulla mappa), eventualmente più volte, senza annullarne la selezione;
- annullarne la selezione;
- modificarlo;
- eliminarlo.

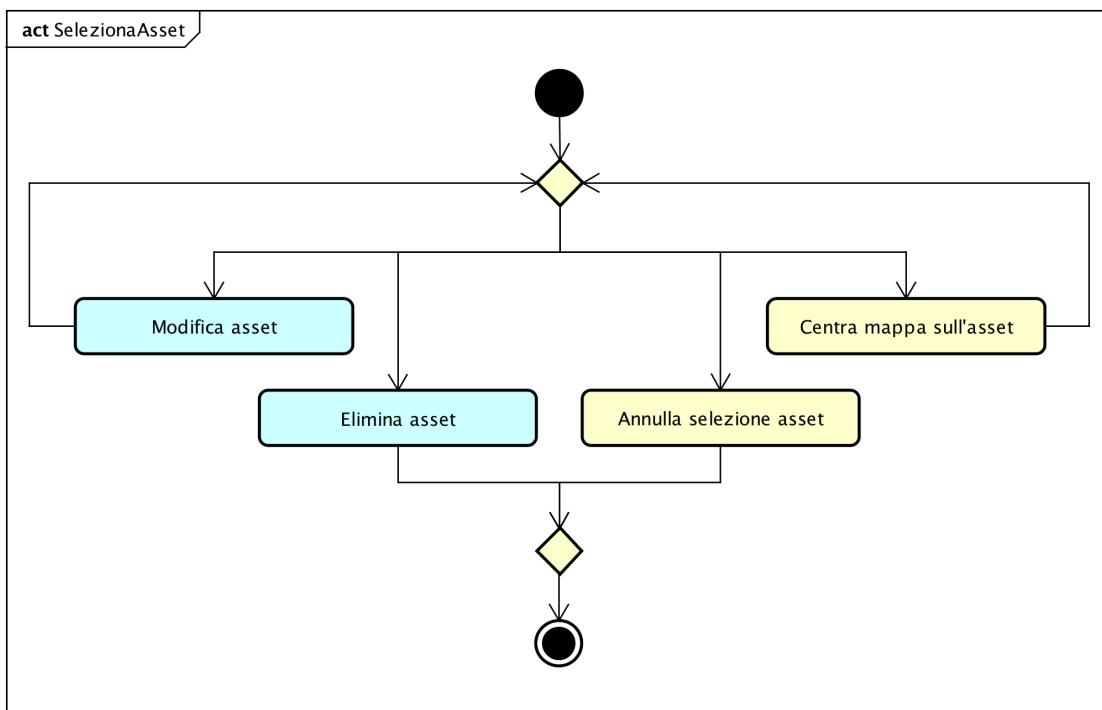


Figura 31: Diagramma di attività per la selezione di un asset

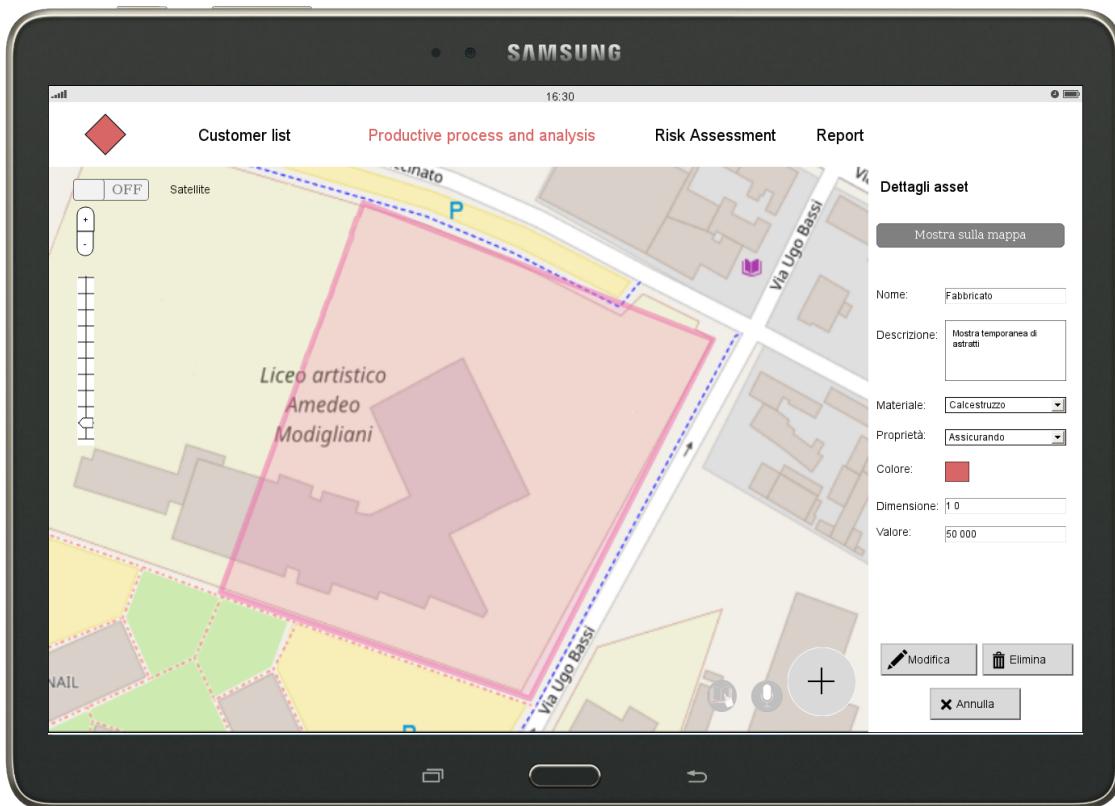


Figura 32: Mockup per la selezione di un asset

6.9 Selezione nodo

Una volta selezionato il nodo l'utente potrà:

- centrarlo automaticamente sulla mappa (utile nel caso in cui nel frattempo l'utente si sia spostato sulla mappa), eventualmente più volte, senza annullarne la selezione;
- annullarne la selezione;
- modificarlo;
- eliminarlo.

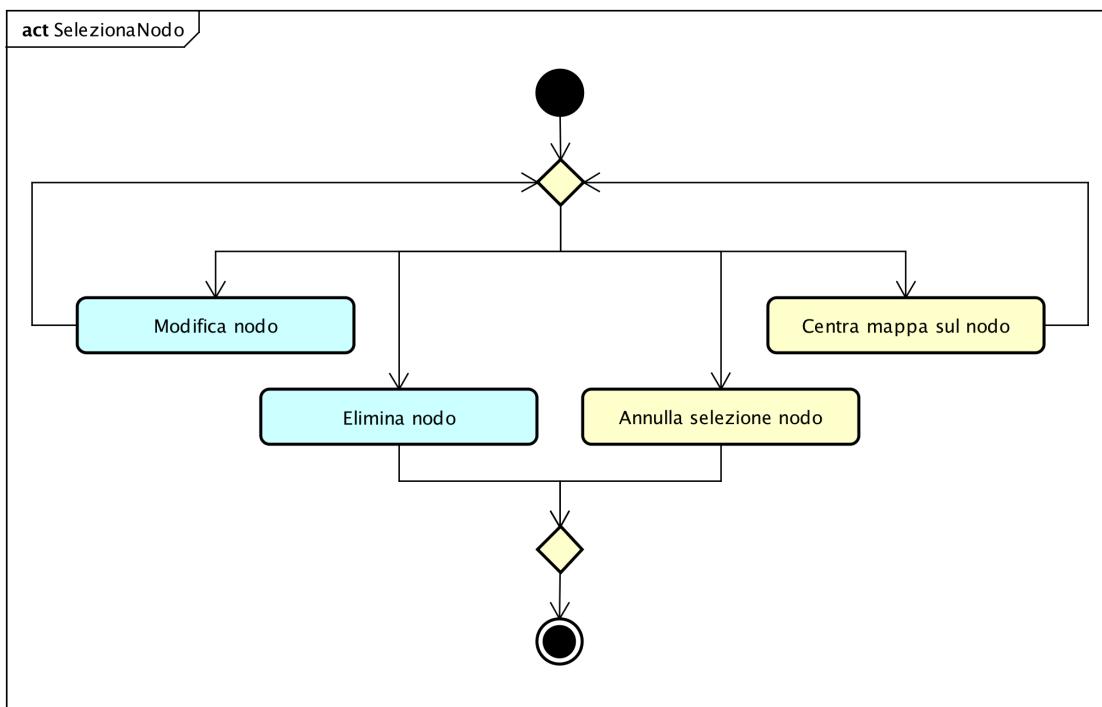


Figura 33: Diagramma di attività per la selezione di un nodo

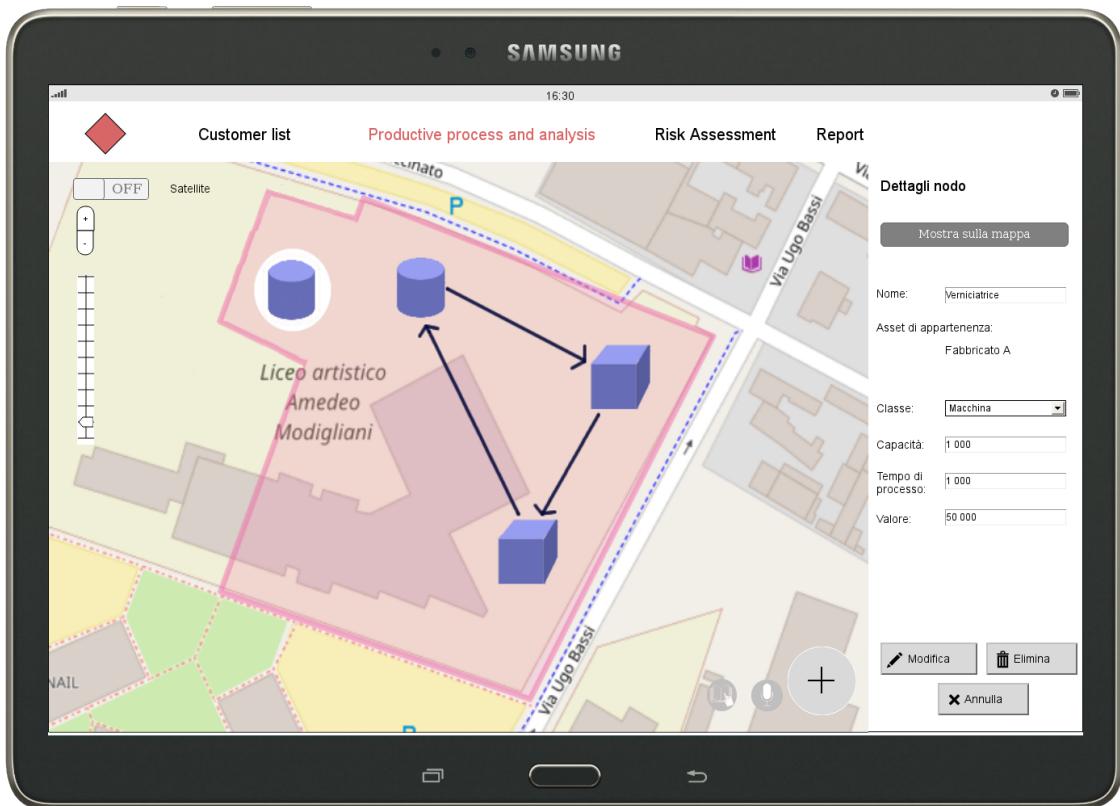


Figura 34: Mockup per la selezione di un nodo

6.10 Selezione arco

Una volta selezionato il nodo l'utente potrà:

- centrarlo automaticamente sulla mappa (utile nel caso in cui nel frattempo l'utente si sia spostato sulla mappa), eventualmente più volte, senza annullarne la selezione;
- annullarne la selezione;
- modificarlo;
- eliminarlo.

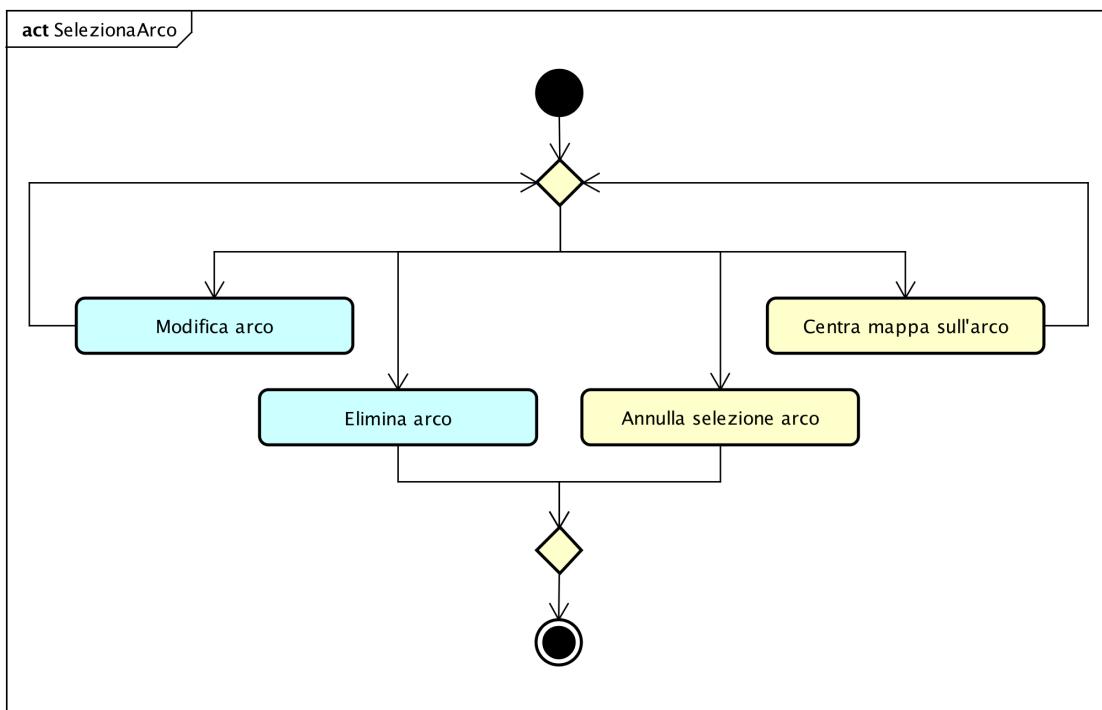


Figura 35: Diagramma di attività per la selezione di un arco

6.11 Selezione scenario

Una volta selezionato lo scenario l'utente potrà:

- annullarne la selezione;
- selezionare un altro scenario;
- modificarlo;
- eliminarlo.

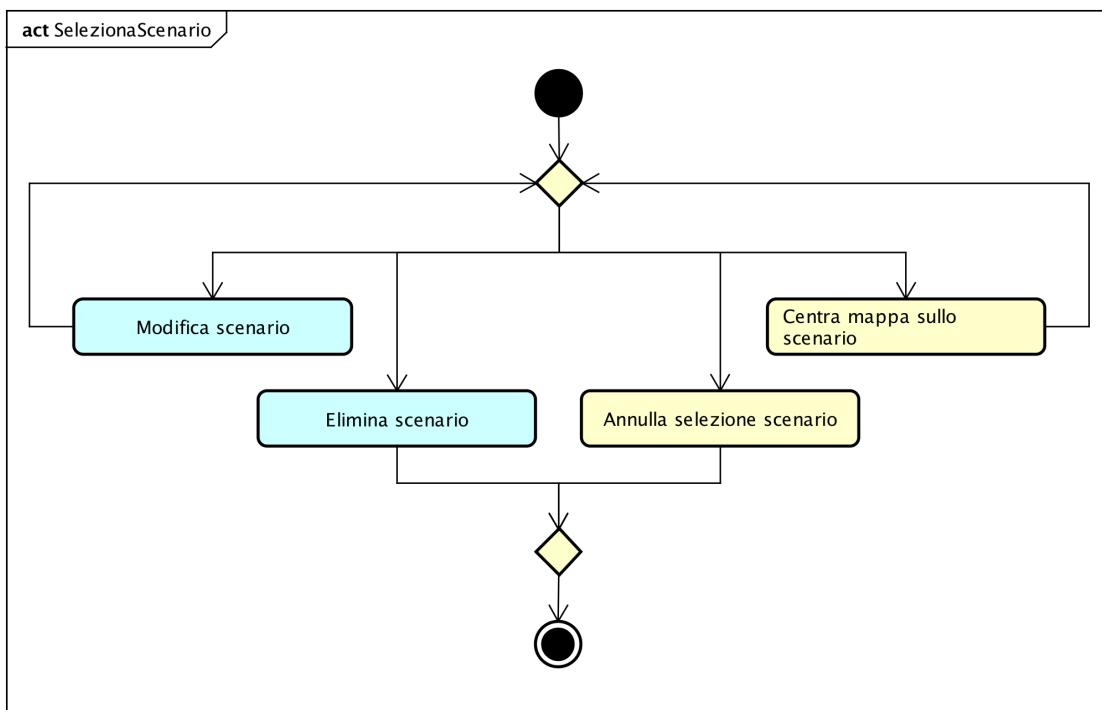


Figura 36: Diagramma di attività per la selezione di uno scenario



Figura 37: Mockup per la selezione di uno scenario

6.12 Modifica asset

Per quanto riguarda la modifica dell'asset, sarà possibile ridisegnarne il perimetro dell'asset sulla mappa e/o modificarne i dati precedentemente compilati. Infine si dovrà confermare la modifica. In caso di dati non corretti, la modifica potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare la modifica. Il sistema richiede una conferma per portare a termine tale operazione.

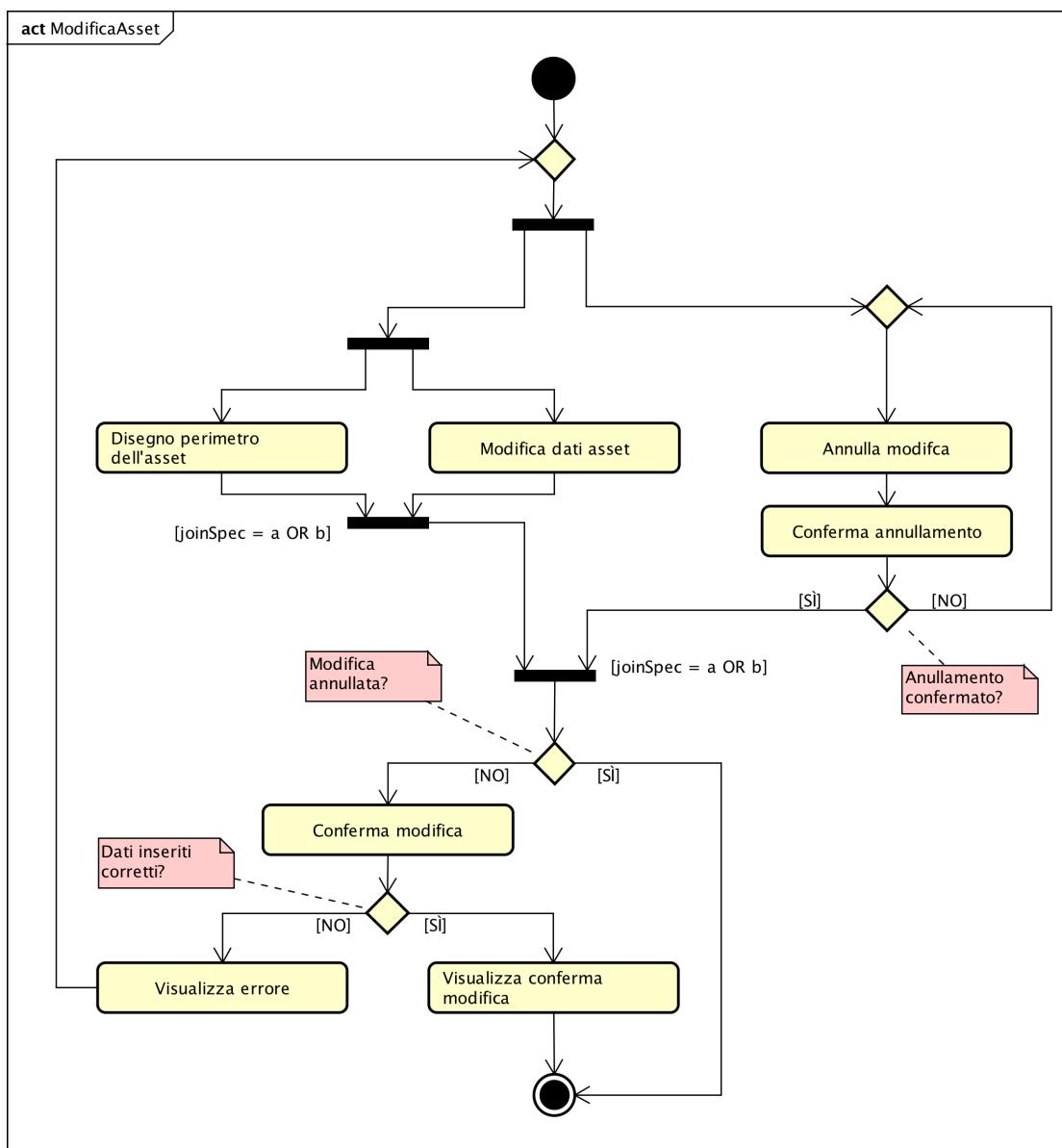


Figura 38: Diagramma di attività per la modifica di un asset

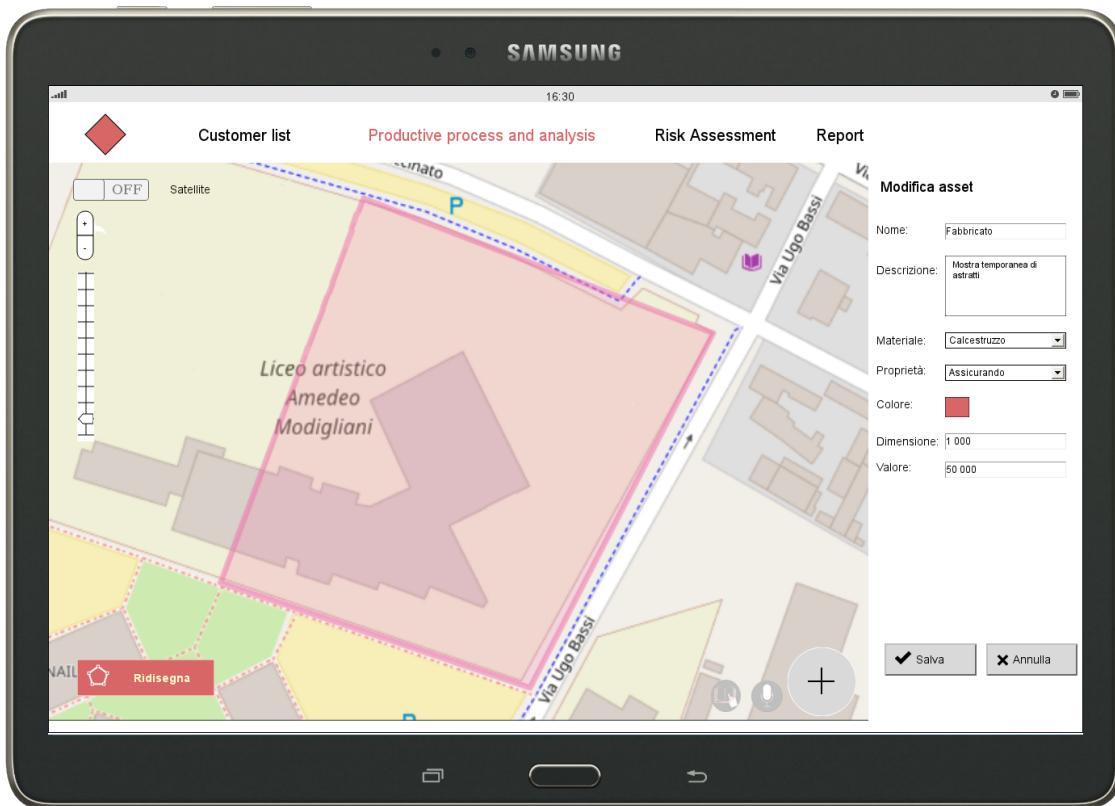


Figura 39: Mockup per la modifica di un asset

6.13 Modifica nodo

Per quanto riguarda la modifica del nodo, sarà possibile riposizionare il nodo sulla mappa e/o modificarne i dati precedentemente compilati. Infine si dovrà confermare la modifica. In caso di dati non corretti, la modifica potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare la modifica. Il sistema richiede una conferma per portare a termine tale operazione.

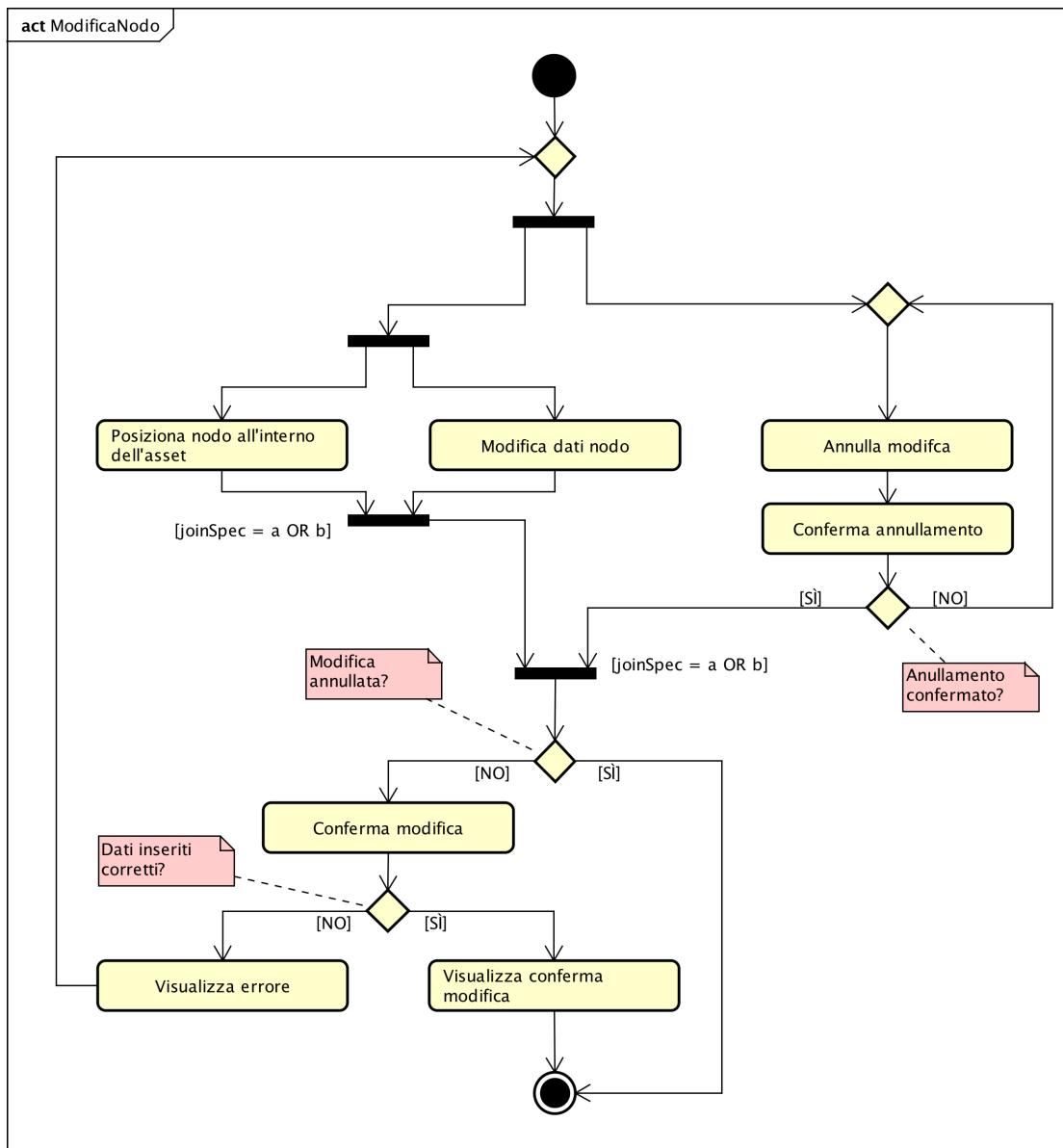


Figura 40: Diagramma di attività per la modifica di un nodo

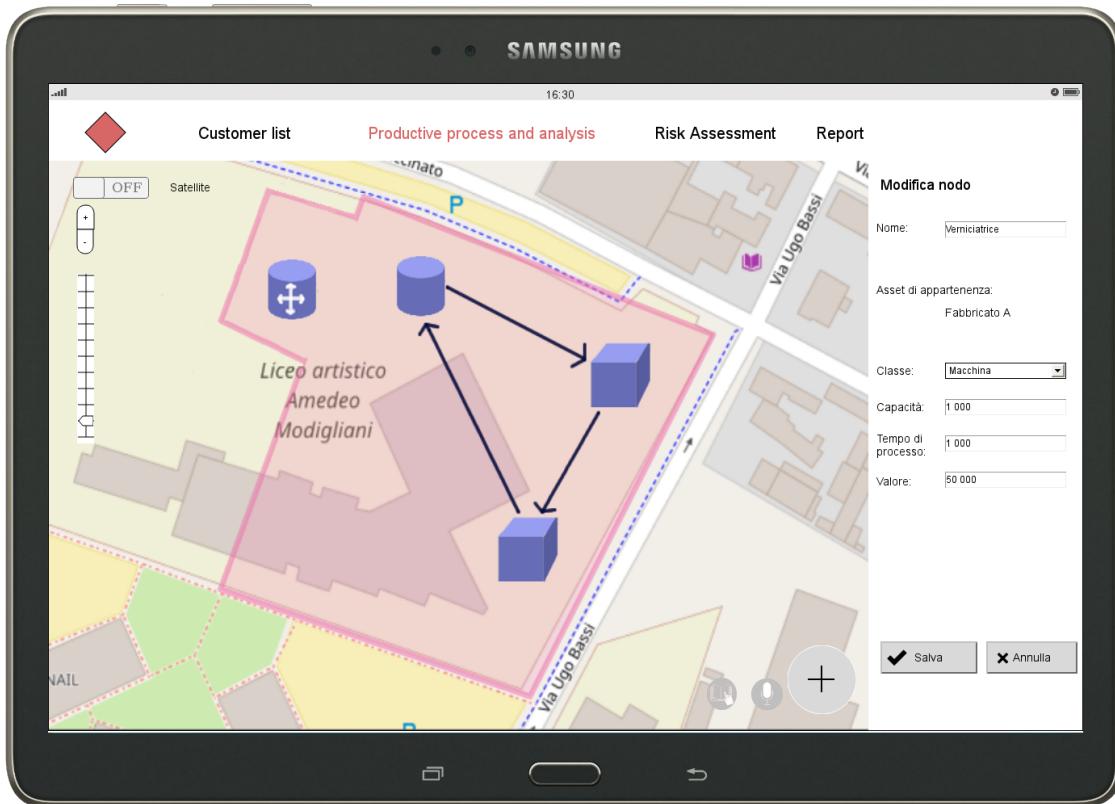


Figura 41: Mockup per la modifica di un nodo

6.14 Modifica arco

Per quanto riguarda la modifica dell'arco, sarà possibile ridisegnare l'arco sulla mappa e/o modificare i dati precedentemente compilati. Infine si dovrà confermare la modifica. In caso di dati non corretti, la modifica potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare la modifica. Il sistema richiede una conferma per portare a termine tale operazione.

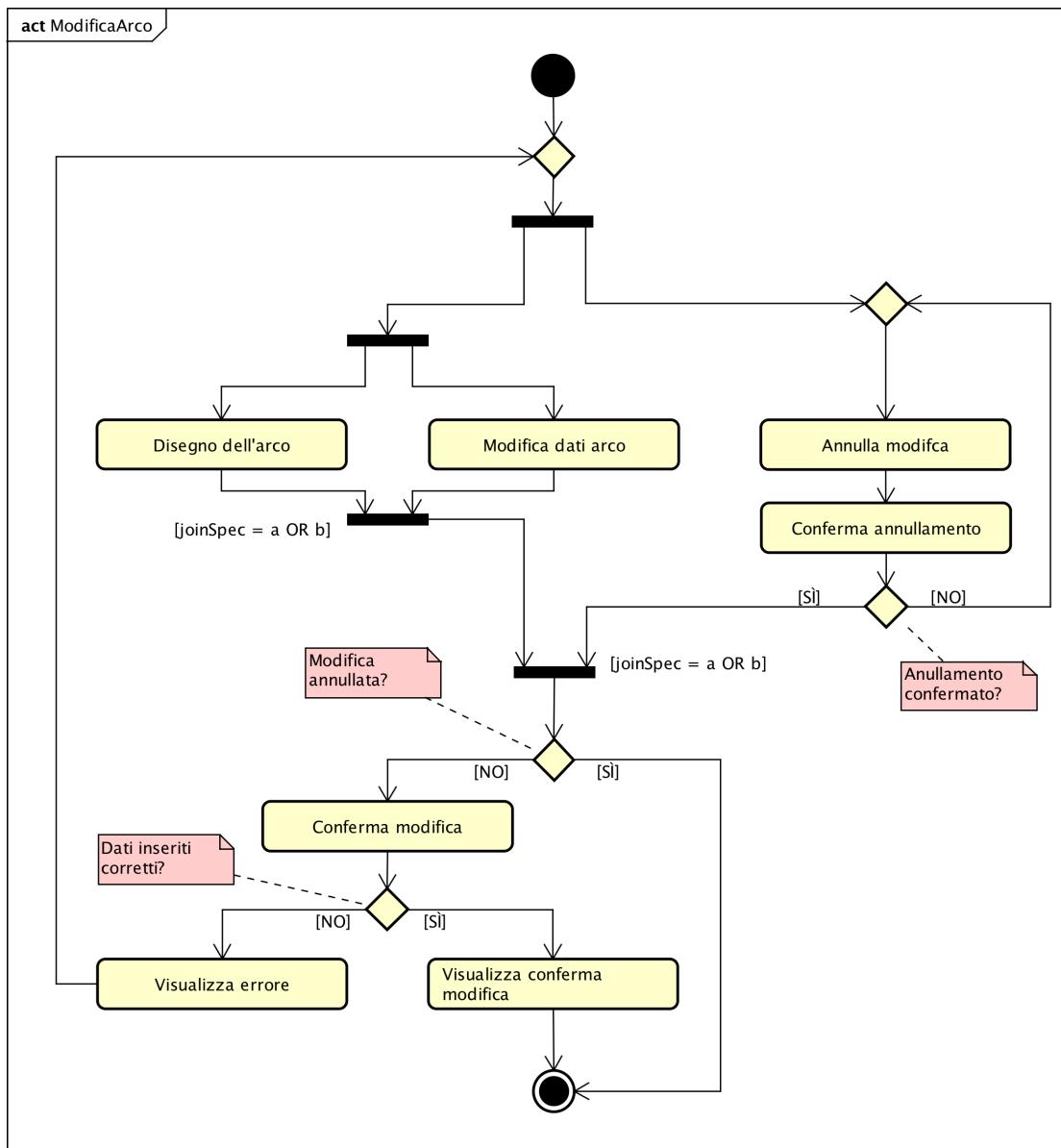


Figura 42: Diagramma di attività per la modifica di un arco

6.15 Modifica scenario

Per quanto riguarda la modifica dello scenario, sarà possibile ridisegnarne il perimetro dell'asset sulla mappa e/o modificarne i dati precedentemente compilati. Infine si dovrà confermare la modifica. In caso di dati non corretti, la modifica potrebbe non andare a buon fine: verrà visualizzato un messaggio di errore e l'utente sarà tenuto a correggere eventuali errori o incompletezze. In ogni momento l'utente può annullare la modifica. Il sistema richiede una conferma per portare a termine tale operazione.

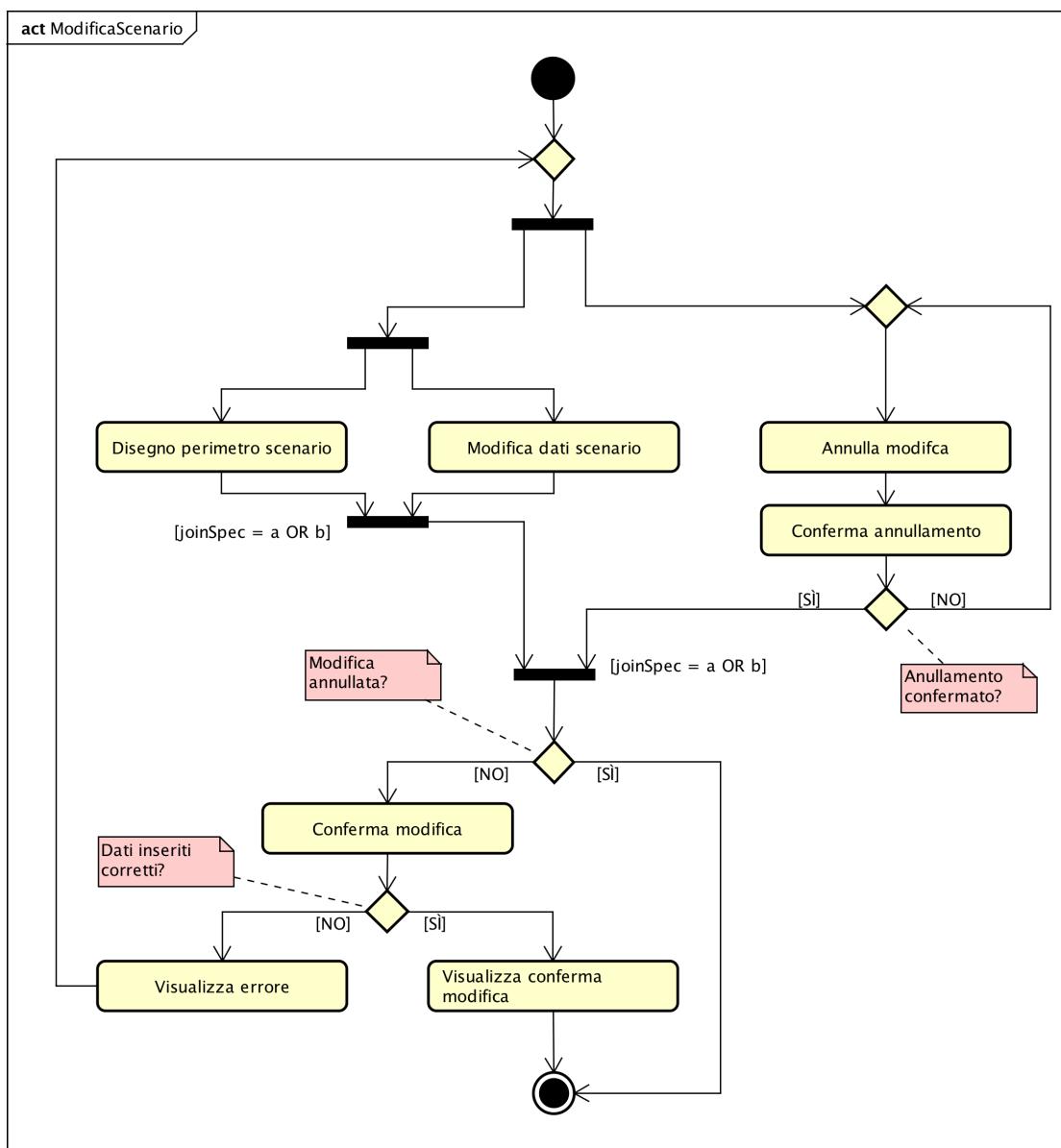


Figura 43: Diagramma di attività per la modifica di uno scenario

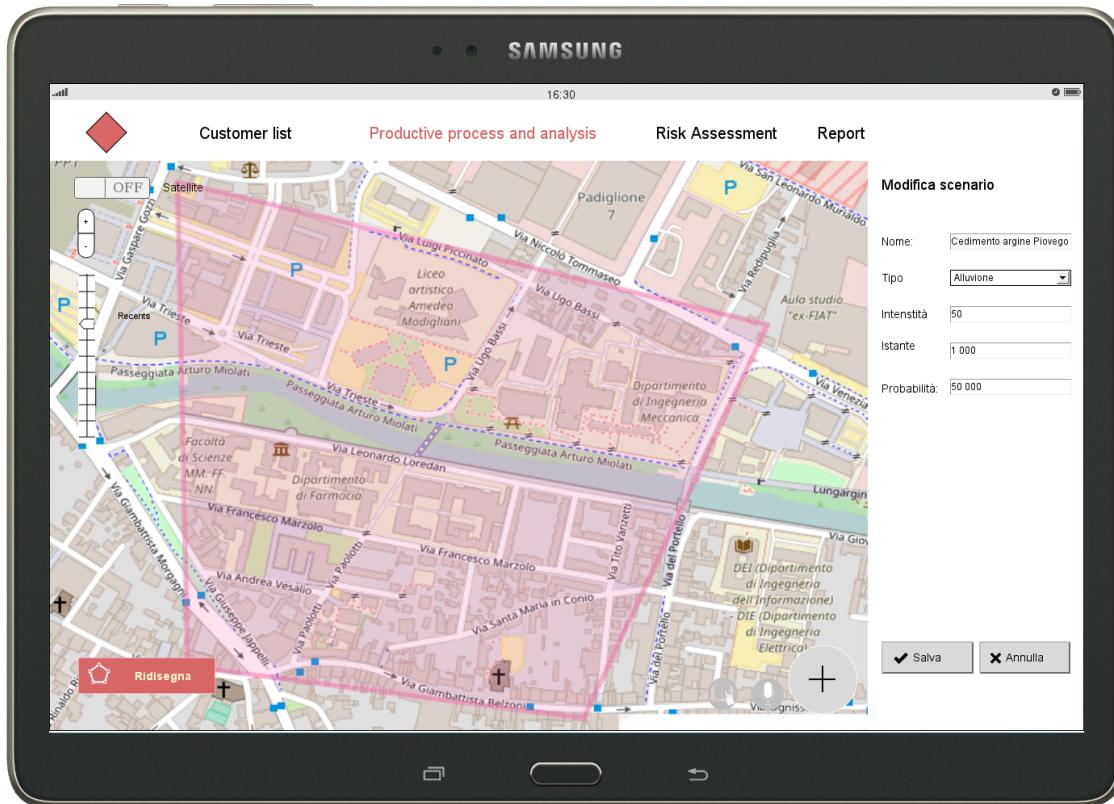


Figura 44: Mockup per la modifica di uno scenario

6.16 Eliminazione asset

Una volta scelto di eliminare l'asset, l'utente è invitato a confermare l'eliminazione o ad annullarla.

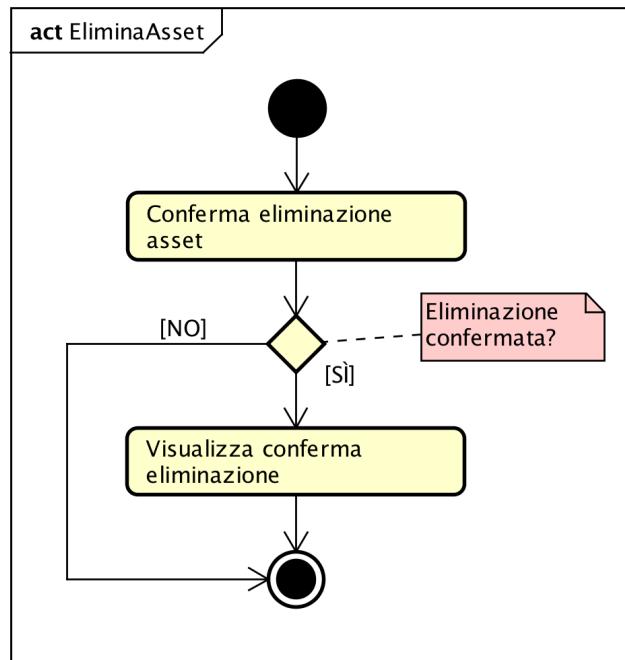


Figura 45: Diagramma di attività per l'eliminazione di un asset

6.17 Eliminazione nodo

Una volta scelto di eliminare il nodo, l'utente è invitato a confermare l'eliminazione o ad annullarla.

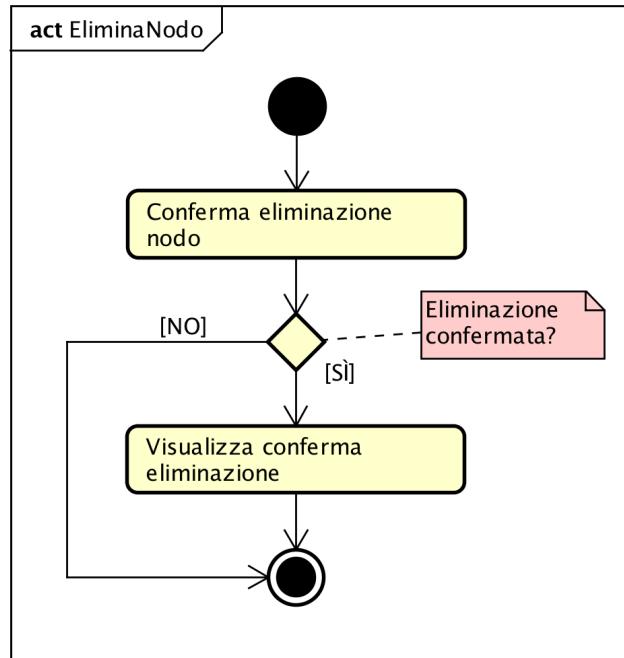


Figura 46: Diagramma di attività per l'eliminazione di un nodo

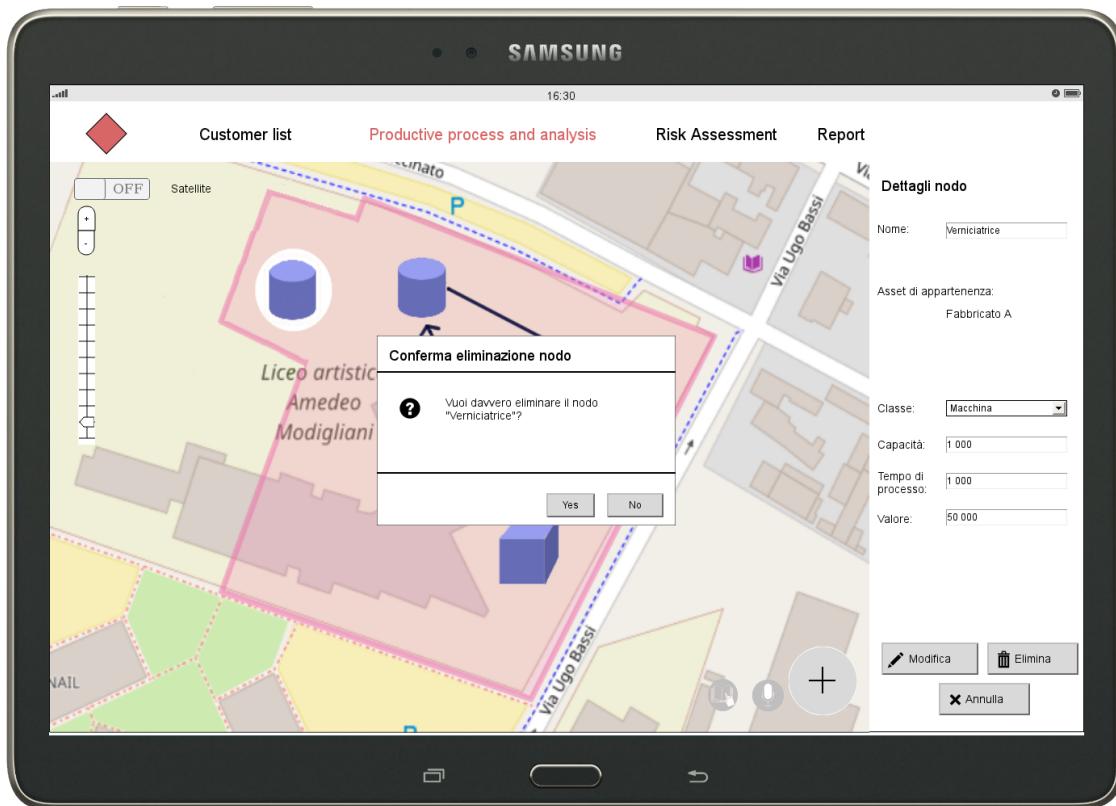


Figura 47: Mockup per l'eliminazione di un nodo

6.18 Eliminazione arco

Una volta scelto di eliminare l'arco, l'utente è invitato a confermare l'eliminazione o ad annullarla.

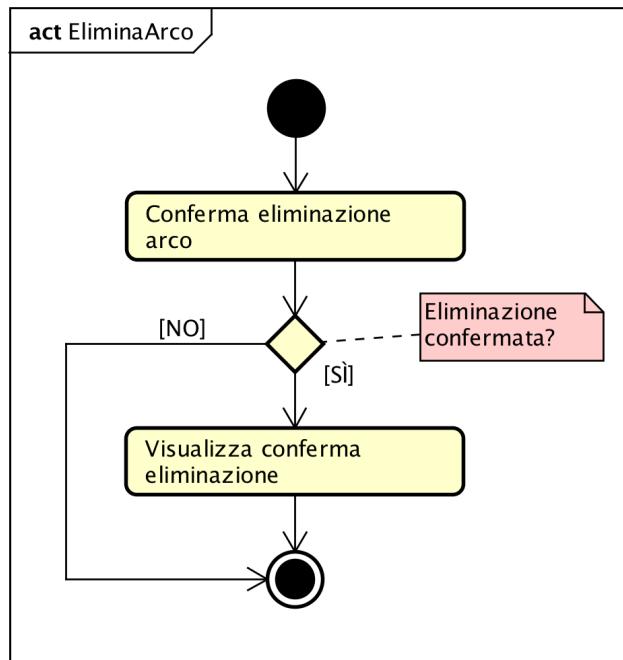


Figura 48: Diagramma di attività per l'eliminazione di un arco

6.19 Eliminazione scenario

Una volta scelto di eliminare lo scenario, l'utente è invitato a confermare l'eliminazione o ad annullarla.

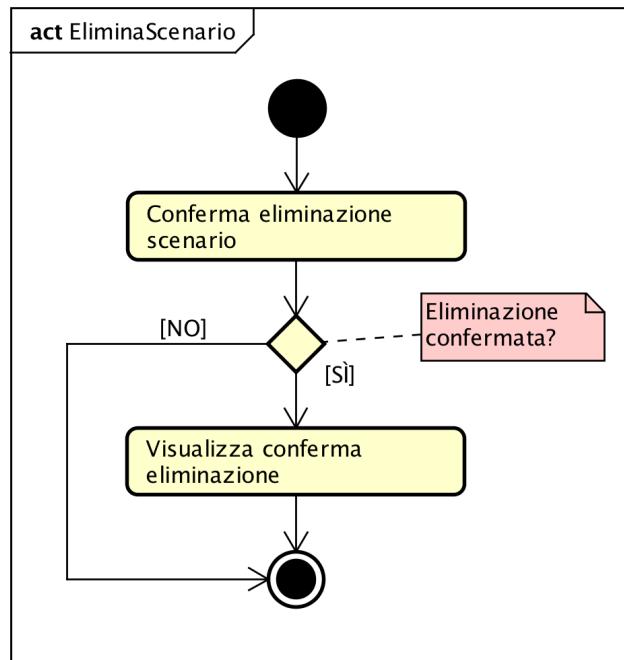


Figura 49: Diagramma di attività per l'eliminazione di uno scenario

6.20 Errore di connessione

In caso di errori di connessione compare un messaggio di errore. In questo caso l'utente può chiudere la visualizzazione del messaggio di errore o ricaricare la pagina.

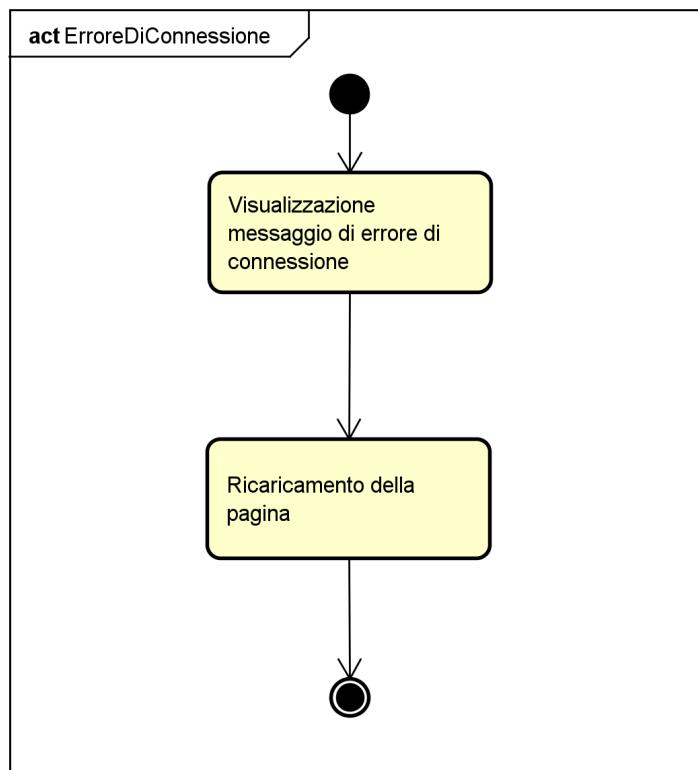


Figura 50: Diagramma di attività per l'eliminazione di un nodo

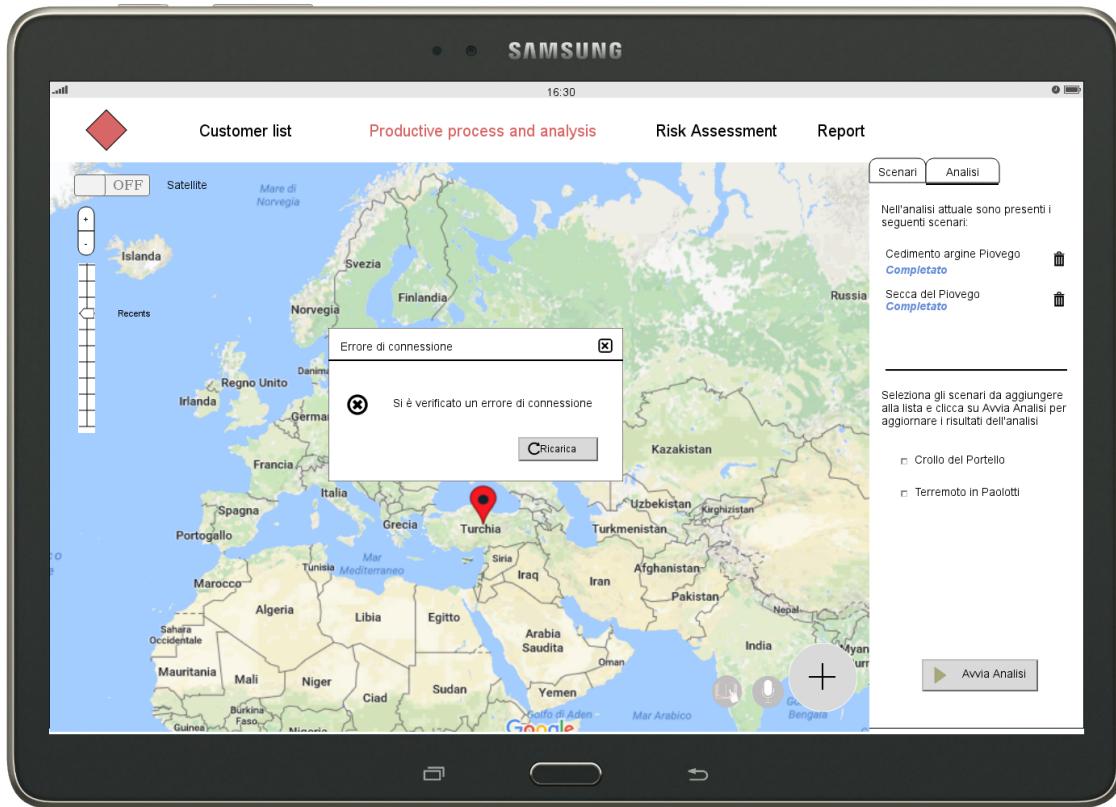


Figura 51: Mockup per gli errori di connessione

7 Tracciamento

7.1 Tracciamento classi-requisiti

Classe	Requisito
AbstractAnalysisButtons	RFF48.1
Analysis	RFF21 RFF22 RFF47
AnalysisAction	RFF21
AnalysisActionCreator	RFF21
AnalysisButtons	RFF22 RFF45 RFF46
AnalysisContent	RFF22 RFF45 RFF46
AnalysisReducer	RFF21
AnalysisSidebarFactory	RFF45 RFF46
Asset	ROF1 ROF1.3 ROF1.4 ROF1.5 ROF2 ROF4 ROF4.4 ROF5
AssetAction	ROF1 ROF1.5 ROF4 ROF4.4 ROF5
AssetActionCreator	ROF1 ROF1.5 ROF4 ROF4.4 ROF5 ROF5.1 ROF6.10

Classe	Requisito
AssetReducer	ROF1 ROF1.5 ROF4 ROF4.4 ROF5 ROF5.1 ROF6.10
ButtonWrapper	RFF16 RFF19.1.9 ROF1 ROF1.3 ROF1.4 ROF10 ROF11 ROF14 ROF15 ROF4 ROF4.3 ROF5 ROF6 ROF9 ROF9.2 ROF9.9
CallManager	RFF16 ROF1 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF5 ROF6 ROF9

Classe	Requisito
ConcreteDeGeOPViewBuilder	RFF16 RFF19 RFF21 ROF1 ROF1.3 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF4.3 ROF5 ROF6 ROF9 ROF9.2 ROF9.9
ConcretePolygon	ROF1.3 ROF1.4
ConcretePolygonFactory	RFF16 RFF19 ROF1 ROF4
Coordinate	ROF1.3 ROF1.4
Customer	RFF16 RFF20 RFF47 RFF48 RFF48.1 ROF1 ROF1.3 ROF1.4 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF5 ROF6 ROF9

Classe	Requisito
DataFromServer	RFF16 RFF21 RFF22 ROF1 ROF11 ROF14 ROF4 ROF6
DataToServer	RFF16 RFF19 RFF20 RFF21 ROF1 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF5 ROF6 ROF9

Classe	Requisito
DeGeOPView	RFF16
	RFF19.4
	RFF21
	RFF24.4
	RFF25
	RFF25.1
	RFF26
	RFF26.1
	RFF47
	RFF48
	ROF1
	ROF1.3
	ROF1.4
	ROF1.5
	ROF10
	ROF11
	ROF14
	ROF15
	ROF24
	ROF24.1
	ROF24.2
	ROF24.3
	ROF39
	ROF4
	ROF4.1.1
	ROF4.1.2
	ROF4.1.3
	ROF4.2
	ROF4.3
	ROF4.5
	ROF40
	ROF41
	ROF5
	ROF6
	ROF6.11
	ROF9
	ROF9.2
	ROF9.9

Classe	Requisito
Director	RFF16 RFF19 RFF21 ROF1 ROF1.3 ROF1.4 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF4.3 ROF5 ROF6 ROF9 ROF9.2 ROF9.9
Edge	ROF1.3 ROF1.4 ROF10 ROF11 ROF12 ROF14 ROF15 ROF5
EdgeAction	ROF11 ROF14 ROF15
EdgeActionCreator	RFF14.5 ROF11 ROF14 ROF14.7 ROF15 ROF15.1
EdgeReducer	RFF14.5 ROF11 ROF11.5 ROF14 ROF14.1 ROF14.2 ROF15 ROF15.1

Classe	Requisito
EditAssetButtons	ROF4 ROF4.2.1 ROF4.4
EditAssetContent	ROF4 ROF4.2.1 ROF4.2.2 ROF4.2.2.1 ROF4.2.3 ROF4.2.3.6 ROF4.2.4 ROF4.2.4.4 ROF4.2.5 ROF4.2.5.2 ROF4.2.6 ROF4.2.6.1 ROF4.2.7 ROF4.2.7.1 ROF4.4
EditAssetSidebarFactory	ROF4 ROF4.2 ROF4.2.1
EditEdgeButtons	RFF14.5 ROF14 ROF14.3 ROF14.7
EditEdgeContent	RFF14.4 RFF14.5 RFF14.5.1 RFF14.5.1.1 RFF14.5.2 RFF14.5.2.1 ROF14
EditEdgeSidebarFactory	ROF14
EditNodeButtons	ROF9 ROF9.1.1 ROF9.1.3 ROF9.10 ROF9.2 ROF9.9

Classe	Requisito
EditNodeContent	ROF1.2.3.3 ROF14.1 ROF14.2 ROF9 ROF9.1.1 ROF9.1.3 ROF9.10 ROF9.11 ROF9.12 ROF9.12.1 ROF9.12.1.1 ROF9.3 ROF9.4 ROF9.5 ROF9.5.2 ROF9.5.2.1 ROF9.5.3 ROF9.5.3.1 ROF9.5.4 ROF9.5.4.1 ROF9.6 ROF9.6.2 ROF9.6.2.1 ROF9.7 ROF9.7.2 ROF9.7.2.1 ROF9.8
EditNodeSidebarFactory	ROF9
EditScenarioButtons	RFF19 RFF19.1.11 RFF19.2 RFF19.3

Classe	Requisito
EditScenarioContent	RFF19 RFF19.1 RFF19.1.1 RFF19.1.1.1 RFF19.1.2 RFF19.1.2.1 RFF19.1.3 RFF19.1.3.11 RFF19.1.4 RFF19.1.4.1 RFF19.1.5 RFF19.1.5.1 RFF19.1.6 RFF19.1.6.1 RFF19.4
EditScenarioSidebarFactory	RFF19
ExitNode	ROF1.3 ROF1.4 ROF10 ROF5
InsertAssetButtons	ROF6.10

Classe	Requisito
InsertAssetContent	ROF1.2 ROF1.2.1 ROF1.2.1.1 ROF1.2.2 ROF1.2.2.1 ROF1.2.3 ROF1.2.3.1 ROF1.2.3.2 ROF1.2.3.3 ROF1.2.3.4 ROF1.2.3.5 ROF1.2.3.6 ROF1.2.4 ROF1.2.4.1 ROF1.2.4.2 ROF1.2.4.3 ROF1.2.4.4 ROF1.2.5 ROF1.2.5.1 ROF1.2.5.2 ROF1.2.6 ROF1.2.6.1 ROF1.2.7 ROF1.2.7.1 ROF1.6
InsertAssetSidebarFactory	ROF1 ROF1.5
InsertEdgeButtons	ROF11.5
InsertEdgeContent	RFF11.2 RFF11.3 RFF11.3.1 RFF11.3.1.1 RFF11.3.2 RFF11.6 RFF11.7 ROF1.6 ROF11.3.2.1 ROF11.4
InsertEdgeSidebarFactory	ROF11
InsertNodeButtons	ROF6.1 ROF6.10 ROF6.9 ROF9.1

Classe	Requisito
InsertNodeContent	ROF6.1 ROF6.12 ROF6.12.1 ROF6.12.1.1 ROF6.3 ROF6.3.1 ROF6.3.2 ROF6.3.3 ROF6.3.4 ROF6.3.5 ROF6.3.6 ROF6.4 ROF6.5 ROF6.5.2 ROF6.5.2.1 ROF6.5.3 ROF6.5.3.1 ROF6.5.4 ROF6.5.4.1 ROF6.6 ROF6.6.2 ROF6.6.2.1 ROF6.7 ROF6.7.2 ROF6.7.2.1 ROF6.8 ROF9.1
InsertNodeSidebarFactory	ROF6
InsertScenarioButtons	RFF16.1.11 RFF16.2 RFF16.3

Classe	Requisito
InsertScenarioContent	RFF16.1 RFF16.1.1 RFF16.1.1.1 RFF16.1.2 RFF16.1.2.1 RFF16.1.3 RFF16.1.3.1 RFF16.1.3.10 RFF16.1.3.11 RFF16.1.3.2 RFF16.1.3.3 RFF16.1.3.4 RFF16.1.3.5 RFF16.1.3.6 RFF16.1.3.7 RFF16.1.3.8 RFF16.1.3.9 RFF16.1.4 RFF16.1.4.1 RFF16.1.5 RFF16.1.5.1 RFF16.1.6 RFF16.1.6.1 ROF10.2 ROF11.4
InsertScenarioSidebarFactory	RFF16
MachineNode	ROF1.3 ROF1.4 ROF10 ROF5

Classe	Requisito
MapWrapper	RFF16
	RFF16.1.10
	RFF16.1.7
	RFF16.1.7.1
	RFF16.1.8
	RFF16.1.8.1
	RFF16.1.9
	RFF16.2
	RFF16.3
	RFF19.1.10
	RFF19.1.7
	RFF19.1.7.1
	RFF19.1.8
	RFF19.1.8.1
	RFF19.1.9
	RFF24.4
	ROF1
	ROF1.1
	ROF1.1.1
	ROF1.1.2
	ROF1.1.3
	ROF1.3
	ROF1.5
	ROF10
	ROF11
	ROF11.1
	ROF11.1.1
	ROF11.1.2
	ROF14
	ROF15
	ROF24
	ROF24.1
	ROF24.2
	ROF24.3
	ROF4
	ROF4.1
	ROF4.3
	ROF42
	ROF43
	ROF44
	ROF5
	ROF6
	ROF6.1
	ROF6.2
	ROF9
	ROF9.1
	ROF9.1.1
	ROF9.1.3
	ROF9.2
	ROF9.9

Classe	Requisito
MessageWrapper	RFF14.8 RFF16 RFF16.4 RFF19.4 RFF20 RFF22 RFF23 RFF47 RFF48 ROF1 ROF1.3 ROF1.5 ROF1.6 ROF10 ROF11 ROF14 ROF15 ROF15.1 ROF15.2 ROF20.1 ROF4 ROF4.3 ROF4.5 ROF5 ROF6 ROF6.11 ROF9 ROF9.11 ROF9.2 ROF9.9
Node	ROF1.3 ROF1.4 ROF10 ROF6 ROF7 ROF9 ROF9.10
NodeAction	ROF10 ROF6 ROF9 ROF9.10
NodeActionCreator	ROF10 ROF10.1 ROF6 ROF9 ROF9.10

Classe	Requisito
NodeReducer	ROF10 ROF10.1 ROF6 ROF9 ROF9.10
Polygon	RFF16 RFF19 ROF1 ROF1.5 ROF10 ROF15 ROF4 ROF5
PolygonFactory	RFF19
PolygonOperationWrapper	RFF16 RFF16.1.10 RFF16.1.7.1 RFF16.1.8 RFF16.1.8.1 RFF16.1.9 RFF19.1.7 RFF19.1.8 ROF1 ROF1.1 ROF1.2 ROF1.5 ROF10 ROF15 ROF20.2 ROF4 ROF4.1 ROF5
Process	ROF1.3 ROF1.4 ROF10 ROF15 ROF5
QueueNode	ROF1.3 ROF1.4 ROF10 ROF15 ROF5

Classe	Requisito
Reducer	RFF14.5 RFF16 ROF1 ROF1.5 ROF10 ROF10.1 ROF11 ROF11.5 ROF14 ROF14.1 ROF14.2 ROF15 ROF15.1 ROF4 ROF5 ROF5.1 ROF6 ROF6.10 ROF9
ResourceNode	ROF1.3 ROF1.4 ROF10 ROF15 ROF5
Scenario	RFF19 RFF19.1 RFF19.3 RFF20
ScenarioAction	RFF16 RFF19 RFF19.1 RFF19.3
ScenarioActionCreator	RFF16 RFF19 RFF19.1 RFF19.3
ScenarioReducer	RFF16 RFF19 RFF19.1 RFF19.3

Classe	Requisito
Sidebar	RFF16 ROF1 ROF1.3 ROF1.4 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF4.3 ROF5 ROF6 ROF9 ROF9.2 ROF9.9
SidebarFacade	RFF16 ROF1 ROF1.3 ROF1.4 ROF1.5 ROF10 ROF11 ROF14 ROF15 ROF4 ROF4.3 ROF5 ROF6 ROF9 ROF9.2 ROF9.9
SourceNode	ROF1.3 ROF1.4 ROF10 ROF15 ROF5

Classe	Requisito
StoreDeGeOP	RFF20 RFF25 RFF25.1 RFF26 RFF26.1 RFF47 RFF48 RFF48.1 ROF1.3 ROF1.4 ROF10 ROF10.1 ROF11.5 ROF14.1 ROF14.2 ROF15 ROF5 ROF5.1 ROF6.10
TransportEdge	ROF1.3 ROF1.4 ROF10 ROF15 ROF5
ViewAssetButtons	ROF2 ROF3 ROF5.1 ROF5.2 ROF6.10
ViewAssetContent	ROF2 ROF3 ROF42 ROF5.2 ROF6.10
ViewAssetSidebarFactory	ROF5
ViewEdgeButtons	ROF12 ROF13 ROF14.6
ViewEdgeContent	ROF12 ROF13 ROF44
ViewEdgeSidebarFactory	ROF15

Classe	Requisito
ViewNodeButtons	ROF10.1 ROF7 ROF8
ViewNodeContent	ROF10.2 ROF43 ROF7 ROF8
ViewNodeSidebarFactory	ROF10
ViewScenarioButtons	RFF18 RFF20 ROF1.3 ROF1.4 ROF10 ROF15 ROF4.3 ROF5 ROF6.10 ROF6.9 ROF9.2 ROF9.9
ViewScenarioContent	RFF17 RFF18 RFF19.1.11 RFF19.2 RFF20 ROF1.3 ROF1.4 ROF10 ROF15 ROF4.3 ROF5 ROF6.10 ROF6.9 ROF9.2 ROF9.9
ViewScenarioSidebarFactory	RFF20 ROF1.3 ROF1.4 ROF10 ROF15 ROF4.3 ROF5 ROF9.2 ROF9.9

Classe	Requisito
--------	-----------

Tabella 2: Tracciamento classe-requisiti

7.2 Tracciamento requisiti-classi

Requisito	Classe
RFF11.2	InsertEdgeContent
RFF11.3	InsertEdgeContent
RFF11.3.1	InsertEdgeContent
RFF11.3.1.1	InsertEdgeContent
RFF11.3.2	InsertEdgeContent
RFF11.6	InsertEdgeContent
RFF11.7	InsertEdgeContent
RFF14.4	EditEdgeContent
RFF14.5	EdgeActionCreator EdgeReducer EditEdgeButtons EditEdgeContent Reducer
RFF14.5.1	EditEdgeContent
RFF14.5.1.1	EditEdgeContent
RFF14.5.2	EditEdgeContent
RFF14.5.2.1	EditEdgeContent
RFF14.8	MessageWrapper
RFF16	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder ConcretePolygonFactory Customer DataFromServer DataToServer DeGeOPView Director InsertScenarioSidebarFactory MapWrapper MessageWrapper Polygon PolygonOperationWrapper Reducer ScenarioAction ScenarioActionCreator ScenarioReducer Sidebar SidebarFacade

Requisito	Classe
RFF16.1	InsertScenarioContent
RFF16.1.1	InsertScenarioContent
RFF16.1.1.1	InsertScenarioContent
RFF16.1.10	MapWrapper PolygonOperationWrapper
RFF16.1.11	InsertScenarioButtons
RFF16.1.2	InsertScenarioContent
RFF16.1.2.1	InsertScenarioContent
RFF16.1.3	InsertScenarioContent
RFF16.1.3.1	InsertScenarioContent
RFF16.1.3.10	InsertScenarioContent
RFF16.1.3.11	InsertScenarioContent
RFF16.1.3.2	InsertScenarioContent
RFF16.1.3.3	InsertScenarioContent
RFF16.1.3.4	InsertScenarioContent
RFF16.1.3.5	InsertScenarioContent
RFF16.1.3.6	InsertScenarioContent
RFF16.1.3.7	InsertScenarioContent
RFF16.1.3.8	InsertScenarioContent
RFF16.1.3.9	InsertScenarioContent
RFF16.1.4	InsertScenarioContent
RFF16.1.4.1	InsertScenarioContent
RFF16.1.5	InsertScenarioContent
RFF16.1.5.1	InsertScenarioContent
RFF16.1.6	InsertScenarioContent
RFF16.1.6.1	InsertScenarioContent
RFF16.1.7	MapWrapper
RFF16.1.7.1	MapWrapper PolygonOperationWrapper
RFF16.1.8	MapWrapper PolygonOperationWrapper
RFF16.1.8.1	MapWrapper PolygonOperationWrapper

Requisito	Classe
RFF16.1.9	MapWrapper PolygonOperationWrapper
RFF16.2	InsertScenarioButtons MapWrapper
RFF16.3	InsertScenarioButtons MapWrapper
RFF16.4	MessageWrapper
RFF17	ViewScenarioContent
RFF18	ViewScenarioButtons ViewScenarioContent
RFF19	ConcreteDeGeOPViewBuilder ConcretePolygonFactory DataToServer Director EditScenarioButtons EditScenarioContent EditScenarioSidebarFactory Polygon PolygonFactory Scenario ScenarioAction ScenarioActionCreator ScenarioReducer
RFF19.1	EditScenarioContent Scenario ScenarioAction ScenarioActionCreator ScenarioReducer
RFF19.1.1	EditScenarioContent
RFF19.1.1.1	EditScenarioContent
RFF19.1.10	MapWrapper
RFF19.1.11	EditScenarioButtons ViewScenarioContent
RFF19.1.2	EditScenarioContent
RFF19.1.2.1	EditScenarioContent
RFF19.1.3	EditScenarioContent
RFF19.1.3.11	EditScenarioContent
RFF19.1.4	EditScenarioContent
RFF19.1.4.1	EditScenarioContent

Requisito	Classe
RFF19.1.5	EditScenarioContent
RFF19.1.5.1	EditScenarioContent
RFF19.1.6	EditScenarioContent
RFF19.1.6.1	EditScenarioContent
RFF19.1.7	MapWrapper PolygonOperationWrapper
RFF19.1.7.1	MapWrapper
RFF19.1.8	MapWrapper PolygonOperationWrapper
RFF19.1.8.1	MapWrapper
RFF19.1.9	ButtonWrapper MapWrapper
RFF19.2	EditScenarioButtons ViewScenarioContent
RFF19.3	EditScenarioButtons Scenario ScenarioAction ScenarioActionCreator ScenarioReducer
RFF19.4	DeGeOPView EditScenarioContent MessageWrapper
RFF20	Customer DataToServer MessageWrapper Scenario StoreDeGeOP ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
RFF21	Analysis AnalysisAction AnalysisActionCreator AnalysisReducer ConcreteDeGeOPViewBuilder DataFromServer DataToServer DeGeOPView Director

Requisito	Classe
RFF22	Analysis AnalysisButtons AnalysisContent DataFromServer MessageWrapper
RFF23	MessageWrapper
RFF24.4	DeGeOPView MapWrapper
RFF25	DeGeOPView StoreDeGeOP
RFF25.1	DeGeOPView StoreDeGeOP
RFF26	DeGeOPView StoreDeGeOP
RFF26.1	DeGeOPView StoreDeGeOP
RFF45	AnalysisButtons AnalysisContent AnalysisSidebarFactory
RFF46	AnalysisButtons AnalysisContent AnalysisSidebarFactory
RFF47	Analysis Customer DeGeOPView MessageWrapper StoreDeGeOP
RFF48	Customer DeGeOPView MessageWrapper StoreDeGeOP
RFF48.1	AbstractAnalysisButtons Customer StoreDeGeOP

Requisito	Classe
ROF1	Asset AssetAction AssetActionCreator AssetReducer ButtonWrapper CallManager ConcreteDeGeOPViewBuilder ConcretePolygonFactory Customer DataFromServer DataToServer DeGeOPView Director InsertAssetSidebarFactory MapWrapper MessageWrapper Polygon PolygonOperationWrapper Reducer Sidebar SidebarFacade
ROF1.1	MapWrapper PolygonOperationWrapper
ROF1.1.1	MapWrapper
ROF1.1.2	MapWrapper PolygonOperationWrapper
ROF1.1.3	MapWrapper
ROF1.2	InsertAssetContent
ROF1.2.1	InsertAssetContent
ROF1.2.1.1	InsertAssetContent
ROF1.2.2	InsertAssetContent
ROF1.2.2.1	InsertAssetContent
ROF1.2.3	InsertAssetContent
ROF1.2.3.1	InsertAssetContent
ROF1.2.3.2	InsertAssetContent
ROF1.2.3.3	EditNodeContent InsertAssetContent
ROF1.2.3.4	InsertAssetContent
ROF1.2.3.5	InsertAssetContent

Requisito	Classe
ROF1.2.3.6	InsertAssetContent
ROF1.2.4	InsertAssetContent
ROF1.2.4.1	InsertAssetContent
ROF1.2.4.2	InsertAssetContent
ROF1.2.4.3	InsertAssetContent
ROF1.2.4.4	InsertAssetContent
ROF1.2.5	InsertAssetContent
ROF1.2.5.1	InsertAssetContent
ROF1.2.5.2	InsertAssetContent
ROF1.2.6	InsertAssetContent
ROF1.2.6.1	InsertAssetContent
ROF1.2.7	InsertAssetContent
ROF1.2.7.1	InsertAssetContent
ROF1.3	Asset ButtonWrapper ConcreteDeGeOPViewBuilder ConcretePolygon Coordinate Customer DeGeOPView Director Edge ExitNode MachineNode MapWrapper MessageWrapper Node Process QueueNode ResourceNode Sidebar SidebarFacade SourceNode StoreDeGeOP TransportEdge ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory

Requisito	Classe
ROF1.4	Asset ButtonWrapper ConcretePolygon Coordinate Customer DeGeOPView Director Edge ExitNode MachineNode Node Process QueueNode ResourceNode Sidebar SidebarFacade SourceNode StoreDeGeOP TransportEdge ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF1.5	Asset AssetAction AssetActionCreator AssetReducer CallManager ConcreteDeGeOPViewBuilder Customer DataToServer DeGeOPView Director InsertAssetSidebarFactory MapWrapper MessageWrapper Polygon PolygonOperationWrapper Reducer Sidebar SidebarFacade
ROF1.6	InsertAssetContent InsertEdgeContent MessageWrapper

Requisito	Classe
ROF10	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataToServer DeGeOPView Director Edge ExitNode MachineNode MapWrapper MessageWrapper Node NodeAction NodeActionCreator NodeReducer Polygon PolygonOperationWrapper Process QueueNode Reducer ResourceNode Sidebar SidebarFacade SourceNode StoreDeGeOP TransportEdge ViewNodeSidebarFactory ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF10.1	NodeActionCreator NodeReducer Reducer StoreDeGeOP ViewNodeButtons
ROF10.2	InsertScenarioContent ViewNodeContent

Requisito	Classe
ROF11	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataFromServer DataToServer DeGeOPView Director Edge EdgeAction EdgeActionCreator EdgeReducer InsertEdgeSidebarFactory MapWrapper MessageWrapper Reducer Sidebar SidebarFacade
ROF11.1	MapWrapper
ROF11.1.1	MapWrapper
ROF11.1.2	MapWrapper
ROF11.3.2.1	InsertEdgeContent
ROF11.4	InsertEdgeContent InsertScenarioContent
ROF11.5	EdgeReducer InsertEdgeButtons Reducer StoreDeGeOP
ROF12	Edge ViewEdgeButtons ViewEdgeContent
ROF13	ViewEdgeButtons ViewEdgeContent

Requisito	Classe
ROF14	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataFromServer DataToServer DeGeOPView Director Edge EdgeAction EdgeActionCreator EdgeReducer EditEdgeButtons EditEdgeContent EditEdgeSidebarFactory MapWrapper MessageWrapper Reducer Sidebar SidebarFacade
ROF14.1	EdgeReducer EditNodeContent Reducer StoreDeGeOP
ROF14.2	EdgeReducer EditNodeContent Reducer StoreDeGeOP
ROF14.3	EditEdgeButtons
ROF14.6	ViewEdgeButtons
ROF14.7	EdgeActionCreator EditEdgeButtons

Requisito	Classe
ROF15	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataToServer DeGeOPView Director Edge EdgeAction EdgeActionCreator EdgeReducer MapWrapper MessageWrapper Polygon PolygonOperationWrapper Process QueueNode Reducer ResourceNode Sidebar SidebarFacade SourceNode StoreDeGeOP TransportEdge ViewEdgeSidebarFactory ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF15.1	EdgeActionCreator EdgeReducer MessageWrapper Reducer
ROF15.2	MessageWrapper
ROF2	Asset ViewAssetButtons ViewAssetContent
ROF20.1	MessageWrapper
ROF20.2	PolygonOperationWrapper
ROF24	DeGeOPView MapWrapper
ROF24.1	DeGeOPView MapWrapper

Requisito	Classe
ROF24.2	DeGeOPView MapWrapper
ROF24.3	DeGeOPView MapWrapper
ROF3	ViewAssetButtons ViewAssetContent
ROF39	DeGeOPView
ROF4	Asset AssetAction AssetActionCreator AssetReducer ButtonWrapper CallManager ConcreteDeGeOPViewBuilder ConcretePolygonFactory Customer DataFromServer DataToServer DeGeOPView Director EditAssetButtons EditAssetContent EditAssetSidebarFactory MapWrapper MessageWrapper Polygon PolygonOperationWrapper Reducer Sidebar SidebarFacade
ROF4.1	MapWrapper PolygonOperationWrapper
ROF4.1.1	DeGeOPView
ROF4.1.2	DeGeOPView
ROF4.1.3	DeGeOPView
ROF4.2	DeGeOPView EditAssetSidebarFactory
ROF4.2.1	EditAssetButtons EditAssetContent EditAssetSidebarFactory
ROF4.2.2	EditAssetContent

Requisito	Classe
ROF4.2.2.1	EditAssetContent
ROF4.2.3	EditAssetContent
ROF4.2.3.6	EditAssetContent
ROF4.2.4	EditAssetContent
ROF4.2.4.4	EditAssetContent
ROF4.2.5	EditAssetContent
ROF4.2.5.2	EditAssetContent
ROF4.2.6	EditAssetContent
ROF4.2.6.1	EditAssetContent
ROF4.2.7	EditAssetContent
ROF4.2.7.1	EditAssetContent
ROF4.3	ButtonWrapper ConcreteDeGeOPViewBuilder DeGeOPView Director MapWrapper MessageWrapper Sidebar SidebarFacade ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF4.4	Asset AssetAction AssetActionCreator AssetReducer EditAssetButtons EditAssetContent
ROF4.5	DeGeOPView MessageWrapper
ROF40	DeGeOPView
ROF41	DeGeOPView
ROF42	MapWrapper ViewAssetContent
ROF43	MapWrapper ViewNodeContent
ROF44	MapWrapper ViewEdgeContent

Requisito	Classe
ROF5	Asset AssetAction AssetActionCreator AssetReducer ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataToServer DeGeOPView Director Edge ExitNode MachineNode MapWrapper MessageWrapper Polygon PolygonOperationWrapper Process QueueNode Reducer ResourceNode Sidebar SidebarFacade SourceNode StoreDeGeOP TransportEdge ViewAssetSidebarFactory ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF5.1	AssetActionCreator AssetReducer Reducer StoreDeGeOP ViewAssetButtons
ROF5.2	ViewAssetButtons ViewAssetContent

Requisito	Classe
ROF6	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataFromServer DataToServer DeGeOPView Director InsertNodeSidebarFactory MapWrapper MessageWrapper Node NodeAction NodeActionCreator NodeReducer Reducer Sidebar SidebarFacade
ROF6.1	InsertNodeButtons InsertNodeContent MapWrapper
ROF6.10	AssetActionCreator AssetReducer InsertAssetButtons InsertNodeButtons Reducer StoreDeGeOP ViewAssetButtons ViewAssetContent ViewScenarioButtons ViewScenarioContent
ROF6.11	DeGeOPView MessageWrapper
ROF6.12	InsertNodeContent
ROF6.12.1	InsertNodeContent
ROF6.12.1.1	InsertNodeContent
ROF6.2	MapWrapper
ROF6.3	InsertNodeContent
ROF6.3.1	InsertNodeContent
ROF6.3.2	InsertNodeContent
ROF6.3.3	InsertNodeContent

Requisito	Classe
ROF6.3.4	InsertNodeContent
ROF6.3.5	InsertNodeContent
ROF6.3.6	InsertNodeContent
ROF6.4	InsertNodeContent
ROF6.5	InsertNodeContent
ROF6.5.2	InsertNodeContent
ROF6.5.2.1	InsertNodeContent
ROF6.5.3	InsertNodeContent
ROF6.5.3.1	InsertNodeContent
ROF6.5.4	InsertNodeContent
ROF6.5.4.1	InsertNodeContent
ROF6.6	InsertNodeContent
ROF6.6.2	InsertNodeContent
ROF6.6.2.1	InsertNodeContent
ROF6.7	InsertNodeContent
ROF6.7.2	InsertNodeContent
ROF6.7.2.1	InsertNodeContent
ROF6.8	InsertNodeContent
ROF6.9	InsertNodeButtons ViewScenarioButtons ViewScenarioContent
ROF7	Node ViewNodeButtons ViewNodeContent
ROF8	ViewNodeButtons ViewNodeContent

Requisito	Classe
ROF9	ButtonWrapper CallManager ConcreteDeGeOPViewBuilder Customer DataToServer DeGeOPView Director EditNodeButtons EditNodeContent EditNodeSidebarFactory MapWrapper MessageWrapper Node NodeAction NodeActionCreator NodeReducer Reducer Sidebar SidebarFacade
ROF9.1	InsertNodeButtons InsertNodeContent MapWrapper
ROF9.1.1	EditNodeButtons EditNodeContent MapWrapper
ROF9.1.3	EditNodeButtons EditNodeContent MapWrapper
ROF9.10	EditNodeButtons EditNodeContent Node NodeAction NodeActionCreator NodeReducer
ROF9.11	EditNodeContent MessageWrapper
ROF9.12	EditNodeContent
ROF9.12.1	EditNodeContent
ROF9.12.1.1	EditNodeContent

Requisito	Classe
ROF9.2	ButtonWrapper ConcreteDeGeOPViewBuilder DeGeOPView Director EditNodeButtons MapWrapper MessageWrapper Sidebar SidebarFacade ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory
ROF9.3	EditNodeContent
ROF9.4	EditNodeContent
ROF9.5	EditNodeContent
ROF9.5.2	EditNodeContent
ROF9.5.2.1	EditNodeContent
ROF9.5.3	EditNodeContent
ROF9.5.3.1	EditNodeContent
ROF9.5.4	EditNodeContent
ROF9.5.4.1	EditNodeContent
ROF9.6	EditNodeContent
ROF9.6.2	EditNodeContent
ROF9.6.2.1	EditNodeContent
ROF9.7	EditNodeContent
ROF9.7.2	EditNodeContent
ROF9.7.2.1	EditNodeContent
ROF9.8	EditNodeContent

Requisito	Classe
ROF9.9	ButtonWrapper ConcreteDeGeOPViewBuilder DeGeOPView Director EditNodeButtons MapWrapper MessageWrapper Sidebar SidebarFacade ViewScenarioButtons ViewScenarioContent ViewScenarioSidebarFactory

Tabella 3: Tracciamento requisito-classi

A Descrizione design pattern

A.1 Introduzione

I *design pattern* semplificano l'attività di progettazione, favorendo il riutilizzo del codice e rendendo l'architettura più manutenibile. I design pattern vengono definiti come soluzioni progettuali generali a problemi ricorrenti. Si tratta di una descrizione o dei modelli logici da applicare per la risoluzione di problemi che possono presentarsi durante la $fase_G$ di progettazione. Esistono diversi design pattern e vengono suddivisi in base al problema da risolvere:

- **pattern creazionali:** nascondono i costruttori delle classi e espongono dei metodi al loro posto. In questo modo si possono utilizzare oggetti senza sapere come sono implementati;
- **pattern comportamentali:** forniscono soluzioni alle più comuni tipologie di interazione tra gli oggetti;
- **pattern architetturali:** operano ad un livello più elevato rispetto ad altri design pattern, ed esprimono schemi di base per impostare l'organizzazione strutturale di un sistema software. In questi schemi si descrivono sottosistemi predefiniti, i ruoli che essi assumono e le relazioni reciproche;
- **pattern strutturali:** consentono di riutilizzare degli oggetti esistenti fornendo agli utilizzatori un'interfaccia più adatta alle loro esigenze.

Sono stati utilizzati i seguenti design pattern:

- Abstract Factory (creazionale);
- Builder (creazionale);
- Factory Method (creazionale);
- Facade (strutturale);
- Redux (architetturale).

Di seguito verranno mostrati i design pattern utilizzati e come vengono contestualizzati nel progetto *DeGeOP*. I diagrammi delle classi mostrati in queste comparazioni sono parziali e a scopo illustrativo. Per la loro visualizzazione completa si rimanda alla sezione 4.

A.2 Pattern Creazionali

A.2.1 Abstract Factory Method

A.2.1.1 Descrizione

Lo scopo dell'Abstract Factory Method è fornire un'interfaccia per creare famiglie di oggetti correlati o dipendenti senza specificare quali siano le loro classi concrete. In questo modo si permette che un sistema sia indipendente dall'implementazione degli oggetti concreti e che il client, attraverso l'interfaccia, utilizzi diverse famiglie di prodotti.

Questo pattern è utile quando:

- si vuole un sistema indipendente da come gli oggetti vengono creati, composti e rappresentati;
- si vuole permettere la configurazione del sistema come scelta tra diverse famiglie di prodotti;
- si vuole fornire una *libreria_G* di classi mostrando solo le interfacce e nascondendo le implementazioni.

A.2.1.2 Contestualizzazione

Il pattern Abstract Factory Method viene utilizzato nel *package_G* ViewPkg per la creazione dell'oggetto di tipo Sidebar. In particolare è stato introdotto un livello in più di astrazione rispetto al design pattern comunemente conosciuto, per legare tra loro le tipologie di Sidebar creabili. *AbstractSidebarFactory* e le sue cinque sottoclassi astratte:

- *AbstractAssetSidebarFactory*;
- *AbstractNodeSidebarFactory*;
- *AbstractEdgeSidebarFactory*;
- *AbstractScenarioSidebarFactory*;
- *AbstractAnalysisSidebarFactory*.

dichiarano soltanto un'interfaccia per la creazione di sidebar, mentre la responsabilità della creazione effettiva delle istanze è delle sottoclassi concrete. È stato quindi definito un metodo di factory per ogni Sidebar. Le factory concrete specificano poi i propri prodotti ridefinendo per ciascuno di essi il metodo factory.

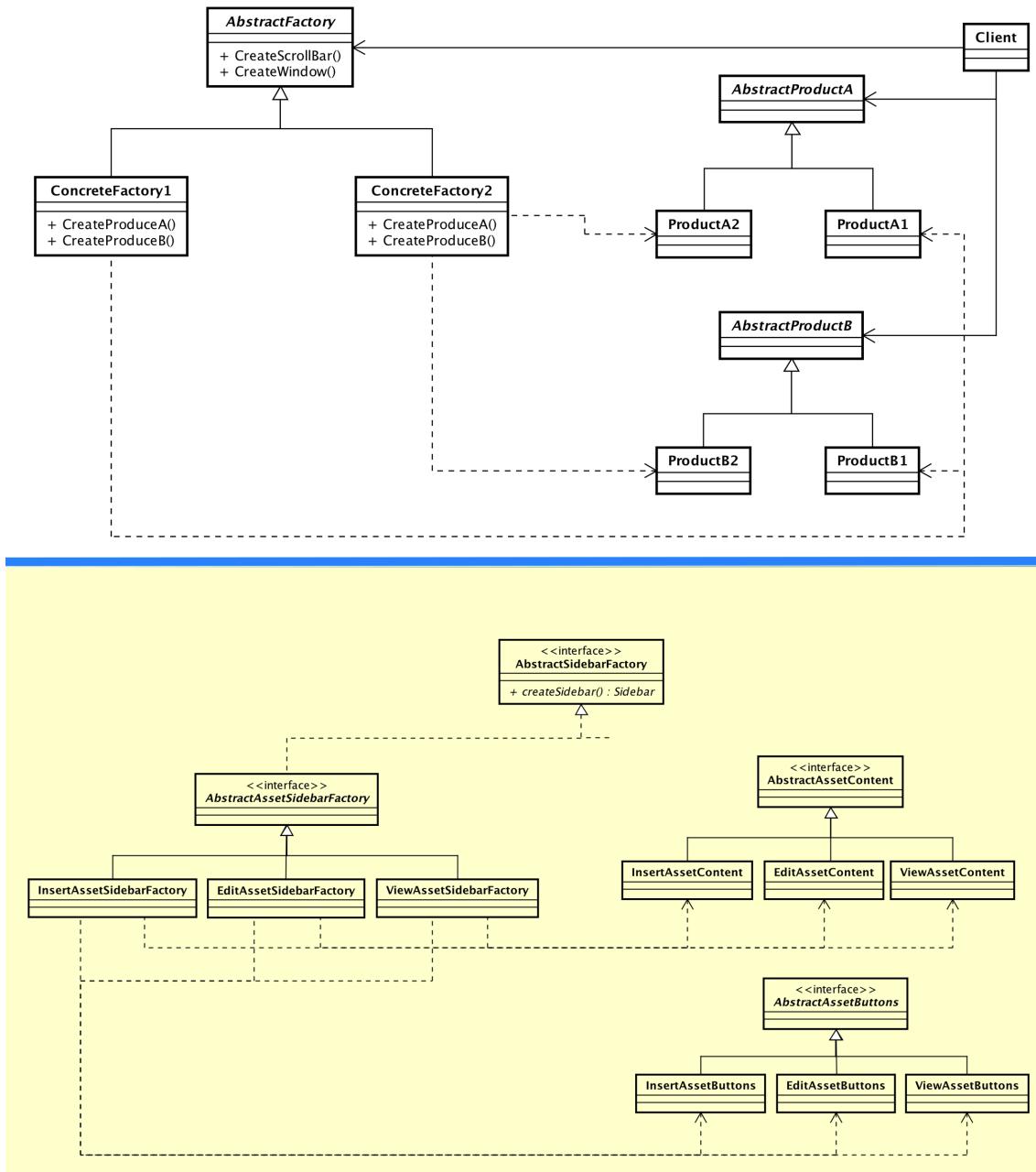


Figura 52: Abstract Factory ed estratto della sua contestualizzazione in DeGeOP

A.2.2 Builder

A.2.2.1 Descrizione

Il Builder pattern, permette di separare la costruzione di un oggetto complesso dalla sua rappresentazione sicché il processo di costruzione stesso possa creare diverse rappresentazioni. L'algoritmo per la creazione di un oggetto complesso è indipendente da come le varie parti vengono assemblate e così la separata classe builder si focalizza sulla corretta costruzione di un'istanza mentre la classe originale si concentra sul funzionamento degli oggetti. Ciò è particolarmente utile quando si vuole assicurare che un oggetto sia valido prima di istanziarlo, e che la logica di controllo non appaia nei costruttori degli oggetti. Un builder permette anche di costruire un oggetto passo-passo, cosa che si può verificare quando si fa il parsing di un testo o si ottengono i parametri da un'interfaccia interattiva.

A.2.2.2 Contestualizzazione

Questo pattern viene utilizzato nel package ViewPkg per la creazione di un oggetto di tipo DeGeOPView.

E' stata creata l'interfaccia DeGeOPViewBuilder che definisce le operazioni che agiscono sull'elemento DeGeOPView che il Director chiederà di creare. Il ConcreteDeGeOPViewBuilder sovrascriverà tali operazioni, incapsulando il modo di costruire e rappresentare l'oggetto DeGeOPView.

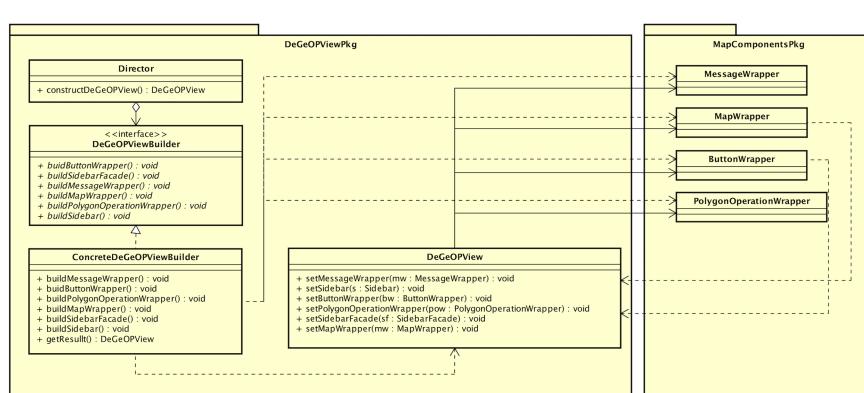
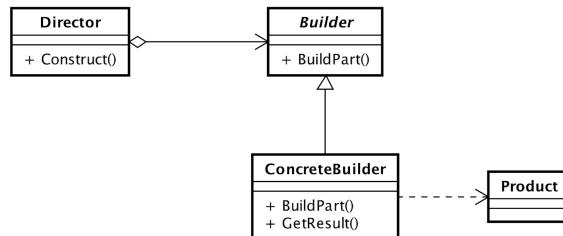


Figura 53: Builder e la sua contestualizzazione in DeGeOP

A.2.3 Factory Method

A.2.3.1 Descrizione

Questo pattern permette di creare oggetti fornendo un'interfaccia per creare un oggetto, ma lascia che le sottoclassi decidano quale oggetto istanziare. Il pattern factory può essere utilizzato quando:

- La creazione di un oggetto preclude il suo riuso senza una significativa duplicazione di codice.
- La creazione di un oggetto richiede l'accesso ad informazioni o risorse che non dovrebbero essere contenute nella classe di composizione.
- La gestione del *ciclo di vita* degli oggetti gestiti deve essere centralizzata in modo da assicurare un comportamento coerente all'interno dell'applicazione.

A.2.3.2 Contestualizzazione

Questo pattern viene utilizzato nel package `StorePkg::PolygonPkg`. Sono state create le classi `ConcretePolygon`, che implementa l'interfaccia `Polygon`, e `ConcretePolygonFactory`, che implementa l'interfaccia `PolygonFactory`. `PolygonFactory` espone il metodo `CreatePolygon`, che viene sovrascritto dalla sua sottoclasse `ConcretePolygonFactory` consentendo quindi ai suoi utilizzatori, `Scenario` e `Asset`, la creazione dell'oggetto di tipo `Polygon`.

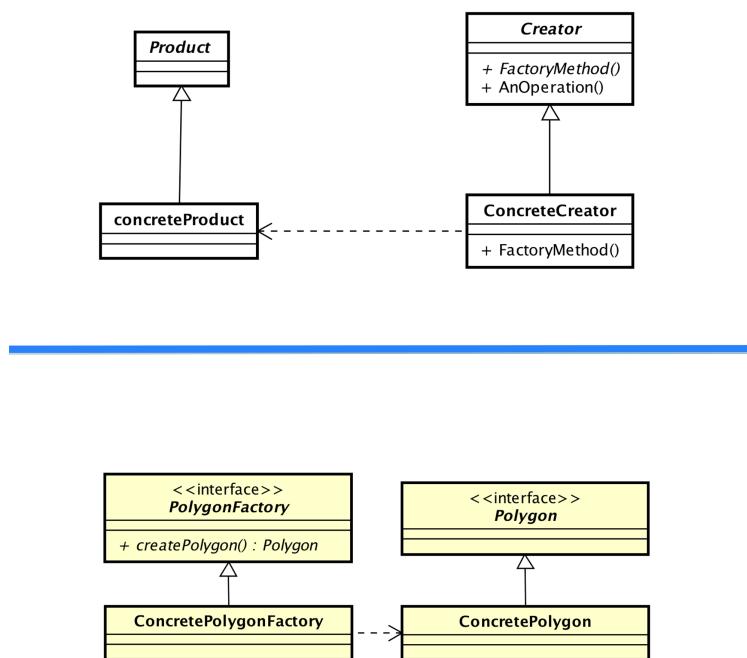


Figura 54: Factory Method e la sua contestualizzazione in DeGeOP

A.2.4 Singleton

A.2.4.1 Descrizione

Il singleton è un design pattern creazionale che ha lo scopo di garantire la creazione di una e una sola istanza di una determinata classe e di fornire un punto di accesso globale a tale istanza.

Solitamente questo pattern viene implementato mettendo privato il costruttore della classe interessata e gestendo gli accessi all'oggetto con chiamate statiche per ottenere l'unico riferimento (anch'esso statico) alla classe. La creazione dell'oggetto può avvenire per un metodo di inizializzazione o la prima volta che si tenta di accedere all'istanza di classe.

Il Singleton pattern viene comunemente usato nell'implementazione di abstract Factory e Builder e a volte nell'implementazione di Facade od oggetti stato.

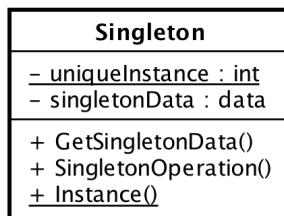


Figura 55: Diagramma del Singleton

A.2.4.2 Contestualizzazione

Molte classi dell'applicazione sono singleton per costruzione. Viene riportato l'elenco delle classi singleton (il percorso è da considerarsi relativo a partire da DeGeOP::).

- StorePkg::AnalysisPkg::AnalysisContainer;
- StorePkg::CustomerContentsPkg::Customer;
- StorePkg::PolygonPkg::PolygonFactory;
- StorePkg::PolygonPkg::ConcretePolygonFactory;
- ReducerPkg::Reducer;
- ReducerPkg::AssetReducer;
- ReducerPkg::NodeReducer;
- ReducerPkg::EdgeReducer;
- ReducerPkg::ScenarioReducer;
- ReducerPkg::AnalysisReducer;

- ActionCreatorPkg::AssetActionCreator;
- ActionCreatorPkg::NodoActionCreator;
- ActionCreatorPkg::EdgeActionCreator;
- ActionCreatorPkg::ScenarioActionCreator;
- ActionCreatorPkg::AnalysisActionCreator;
- CallManagerPkg::CallManager;
- CallManagerPkg::DataFromServer;
- CallManagerPkg::DataToServer;
- View::Director;
- View::DeGeOPViewBuilder;
- View::ConcreteDeGeOPView;
- View::MapPkg::DeGeOPView;
- View::MapPkg::MessageWrapper;
- View::MapPkg::MapWrapper;
- View::MapPkg::ButtonWrapper;
- View::MapPkg::PolygonOperationWrapper;
- tutte le classi contenute in View::SidebarPkg.

A.3 Pattern Strutturali

A.3.1 Facade

A.3.1.1 Descrizione

Questo design pattern ha lo scopo di fornire un'interfaccia unificata per un insieme di interfacce presenti in un sottosistema. Facade definisce un'interfaccia di livello più alto che rende il sottosistema più semplice da utilizzare.

A.3.1.2 Contestualizzazione

Questo pattern viene utilizzato nel package ViewPkg. E' stata creata la classe SidebarFacade, in modo che il client DeGeOPView possa creare le varie tipologie di Sidebar senza la necessità di conoscere le implementazioni dell'interfaccia factory.

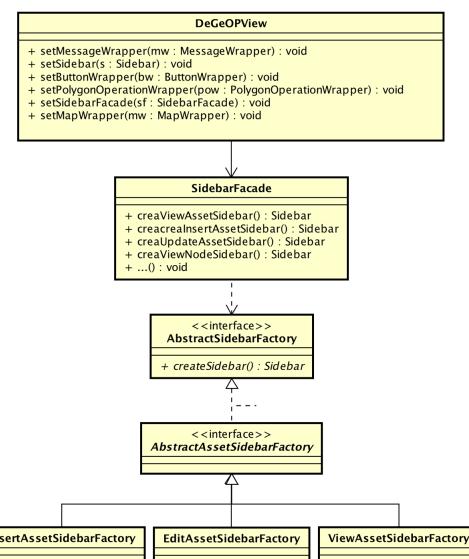
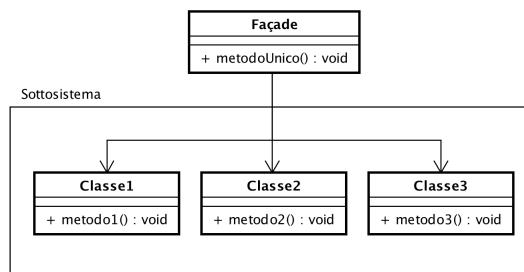


Figura 56: Facade ed estratto della sua contestualizzazione in DeGeOP

A.4 Pattern Architetturali

A.4.1 Redux

A.4.1.1 Descrizione

Il pattern proposto dalla libreria Redux comprende 3 componenti:

- **Store**: è un unico oggetto globale read-only che contiene i dati da gestire, memorizzando l'intero stato del programma;
- **Actions**: definiscono le azioni che si occupano di aggiornare lo Store: l'unico modo di cambiare lo Store è emettere un'azione;
- **Reducer**: funzioni pure che, ricevuto un input uno stato e un'azione, restituiscono un nuovo stato modificato in base all'azione.

Dato uno Store e una funzione Reducer, lo stato del programma viene aggiornato in modo deterministico con i dati delle azioni. Il vero punto di forza di Redux è la gestione del flusso di dati in modo unidirezionale. Questo significa che tutti i dati dell'applicazione seguono lo stesso flusso, rendendo la logica dell'applicazione più predicable e facile da capire e implementare.

A.4.1.2 Contestualizzazione

Questo design pattern viene utilizzato per il design dell'architettura ad alto livello, come illustrato nella sezione [Descrizione architettura](#) e [Tecnologie utilizzate](#). L'intera applicazione nel suo insieme implementa questo design pattern.

Le classi [Action](#) all'interno del package ActionPkg e [Reducer](#) all'interno di ReducerPkg implementano rispettivamente le componenti Action e Reducer di questo design pattern.

L'unico modo per cambiare i dati contenuti all'interno del package StorePkg è emettere delle Actions.

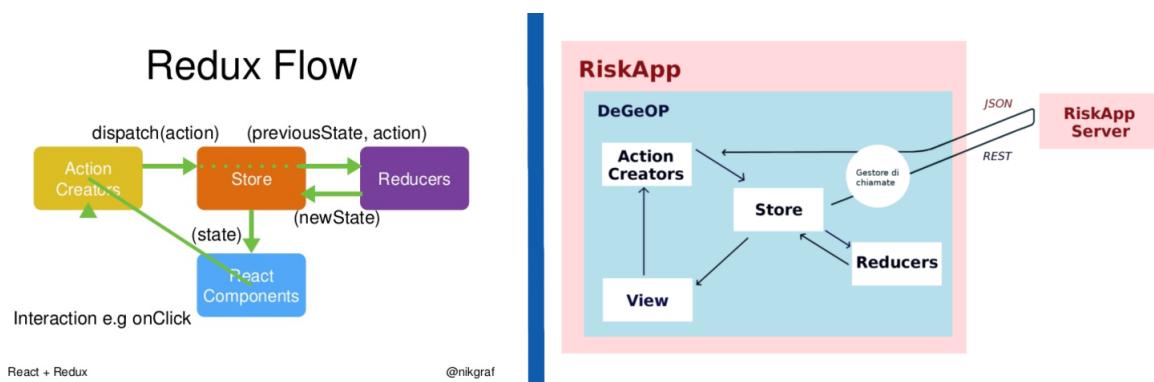


Figura 57: Design pattern proposto da Redux e la sua contestualizzazione in DeGeOP