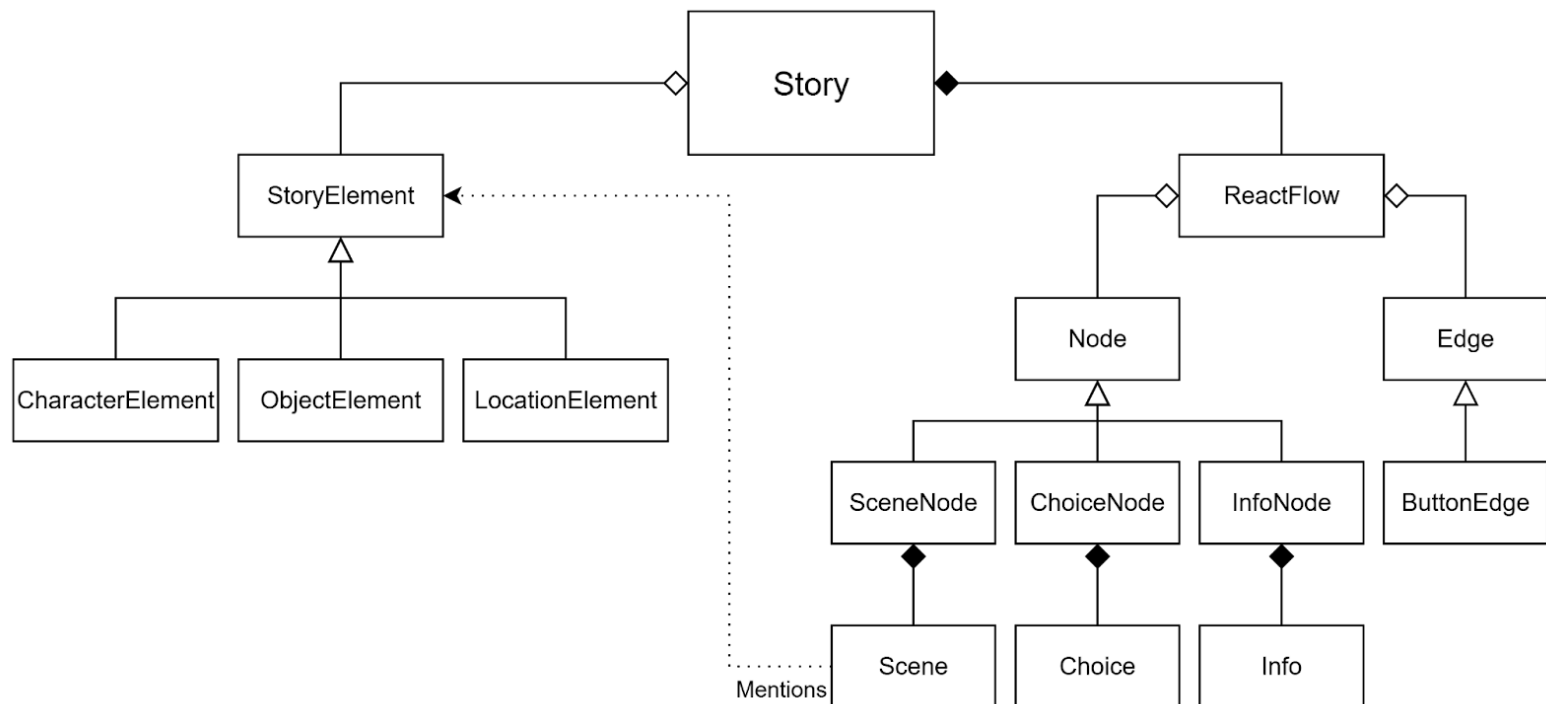


Classi



Il codice si struttura prevalentemente su due rami distinti: ciò che riguarda gli **StoryElement** e ciò che riguarda **ReactFlow**.

StoryElement

Gli **StoryElement**, o Elementi, rappresentano le entità della Storia: Personaggi, Oggetti e Luoghi.

Sono suddivisi virtualmente in due categorie in base a dove risiedono i loro dati: salvati nel DB o salvati localmente.

Gli Elementi del DB provengono dall'Archivio, quindi sono entità che sono realmente esistite e i loro dettagli sono storici.

Gli Elementi locali sono necessari per popolare tutte le scene di una Storia, ma non hanno necessariamente rilevanza storica. Sono quindi salvati assieme alla Storia ed esistono solo al suo interno.

React Flow

È il nome della libreria utilizzata per gestire il Flow Chart della Storia, ma in questo contesto si riferisce a tutte le classi di "backend" che definiscono la struttura e l'ordine della Storia.

Esistono tre tipi di nodi:

- Nonno di Scena - **SceneNode**

- Nodo di Scelta - ChoiceNode
- Nodo di Approfondimento - InfoNode

Il prompt di una Scena può menzionare testualmente un Elemento, quindi il legame tra i due rami esiste, ma è abbastanza lasco (in quanto si basa solo su una corrispondenza tra stringhe).

Segue un'analisi non dettagliata delle classi principali dell'applicazione.

Il codice è stato ampiamente commentato, per dettagli più puntuali rivolgersi ai relativi file.

Story.ts

Story: class

Rappresenta una storia in tutti i suoi dettagli.

Contiene riferimenti a:

- tutti gli Elementi menzionabili al suo interno
- l'oggetto Flow di React Flow

Nota sugli Elementi

La classe ha solo un campo per gli Elementi, nonostante gli elementi salvati nel DB e quelli locali alla Storia siano generalmente gestiti separatamente.

Nel costruttore di Story, gli ID degli Elementi del DB vengono espansi con le informazioni provenienti dal DB, così che abbiano lo stesso formato degli Elementi locali.

Allo stesso modo, durante la serializzazione per il salvataggio, gli Elementi del DB vengono separati, così che vengano salvati solo come ID, mentre quelli locali vengono serializzati integralmente.

StoryElement.ts

StoryElementType: enum

Enumera i tipi di Elementi della Storia:

- `character`
- `object`
- `location`

StoryElement: interface

Tipo che definisce un qualsiasi Elemento appartenente alla Storia.
Molti campi sono condivisi tra tutti i tipi, alcuni esistono solo per certi tipi.

CharacterElement: interface extends StoryElement

Possiede solo i campi base.

ObjectElement: interface extends StoryElement

Oltre ai campi base, possiede anche i campi `material` e `origin`.

LocationElement: interface extends StoryElement

Oltre ai campi base, possiede anche il campo `origin`.

UndoStack.ts

UndoStack<T>: class

Implementa un semplice stack delle modifiche. La serializzazione salva gli ultimi 10 elementi.

Nell'app è utilizzato con `T: {prompt: string, fullText: string}`

Scene.ts

SceneDetails: type

Contiene tutti i vari dettagli narrativi/ambientali di una scena, esclusi i testi.

Scene: class

Rappresenta una singola scena narrativa all'interno della storia.

Racchiude i dettagli della scena e lo storico dei prompt e dei testi generati.

Nota sulla serializzazione

Siccome lo storico richiede una serializzazione dedicata, anche la classe Scene richiede una serializzazione esplicita.

Choice.ts

ChoiceDetails: type

Contiene i dettagli di una singola scelta in una decisione con multiple possibili scelte.

Choice: class

Rappresenta un bivio all'interno della storia.

Racchiude l'interrogativo del bivio e le possibili scelte che si possono compiere.

~~Info.ts~~ → InfoNode.tsx

Siccome il tipo Info è molto semplice, è racchiuso nel componente React del nodo InfoNode.

Info: type

Racchiude il titolo e il contenuto di un box informativo.

@xyflow/ReactFlow

ReactFlowJsonObject: type

Contiene tutti i dati del flow chart della Storia. (Storia \Leftrightarrow Flow)

- `nodes`
- `edges`
- `viewport`

Node: type

Contiene tutti i dati di cui un Nodo necessita per esistere in un flow chart (tranne il componente React che ne definisce l'aspetto grafico, quello è definito in `storyNodeTypes{ }` e relativi file).

data: type

Raccolta di elementi di tipo `unknown`, i contenuti sono lasciati liberi per la singola implementazione.

StoryNode.tsx

Contiene le definizioni di tutti i tipi di Nodo e dei parametri per il componente React di ciascuno.

NodeType: enum

Enumera i tipi di Nodo:

- `scene`
- `choice`
- `info`

EdgeType: enum

Enumera i tipi di Edge:

- `button`

storyNodeTypes: object

Definisce l'associazione tra oggetto Nodo (nodo in memoria) e Nodo componente React (nodo visibile).

Story/Scene/Choice/Info NodeProps : type

Definiscono il tipo delle props per i rispettivi componenti React.

Scene/Choice/Info NodeObject : interface

Definiscono le variazioni della classe Node per ogni tipo di Nodo.

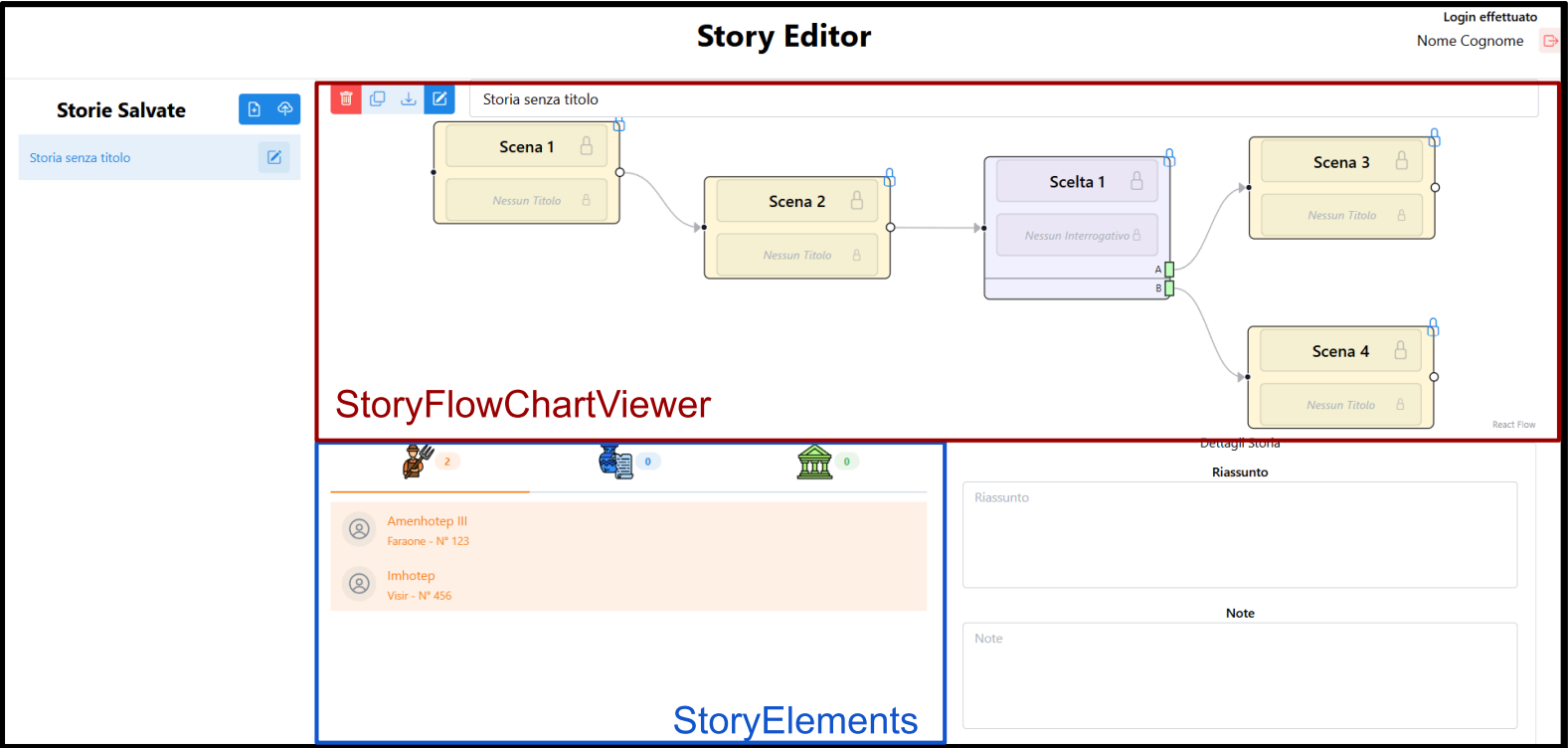
	StoryNode	SceneNode	ChoiceNode	InfoNode
Props	Props comuni a ogni componente	Aggiunge prop scene	Aggiunge prop choice	Aggiunge prop info
Object	/ (Nulla da aggiungere a Node)	data: Scene type: NodeType.Scene	data: Choice type: NodeType.Choice	data: Info type: NodeType.Info

Componenti

App.tsx

Il contenitore più esterno dell'applicazione: contiene Stati, Context e Provider per tutti i livelli sottostanti dell'App.

StoriesDashboard.tsx



StoriesDashboard

Presenta una visualizzazione schematica delle storie salvate e dei relativi dettagli.

StoryEditor.tsx

Il contenitore della singola storia. Tutte le modifiche alla storia avvengono in componenti discendenti di StoryEditor.

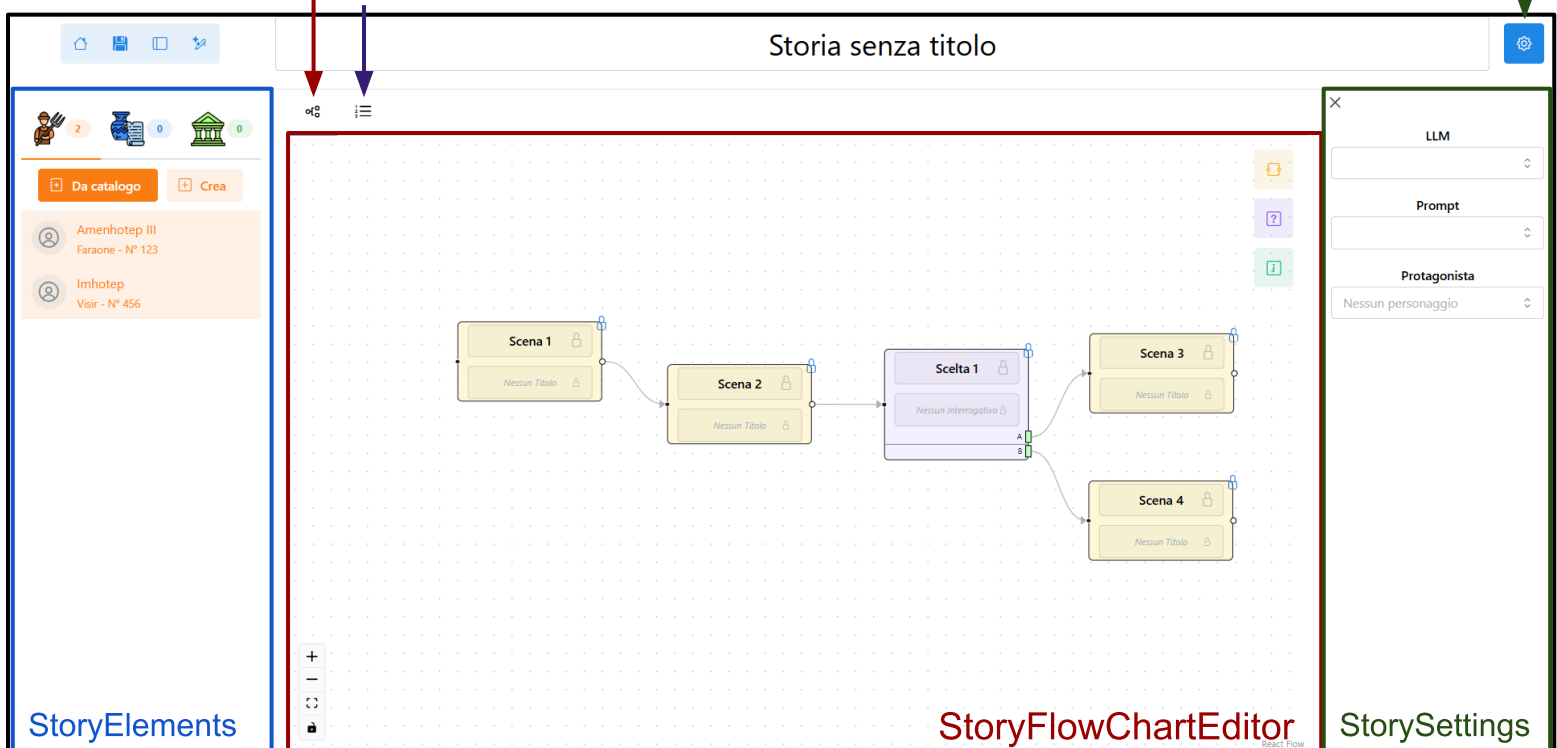
Mentre nella sezione “Navbar” dell’AppShell è sempre montato StoryElements, e nella sezione “Aside” StorySettings, nella sezione “Main” possono essere montati:

- StoryFlowChartEditor (la tab di default)
- StoryTexts
- SceneEditor
- ChoiceEditor
- InfoEditor

StoryFlowChartEditor (Current)

StoryTexts

Toggle StorySettings

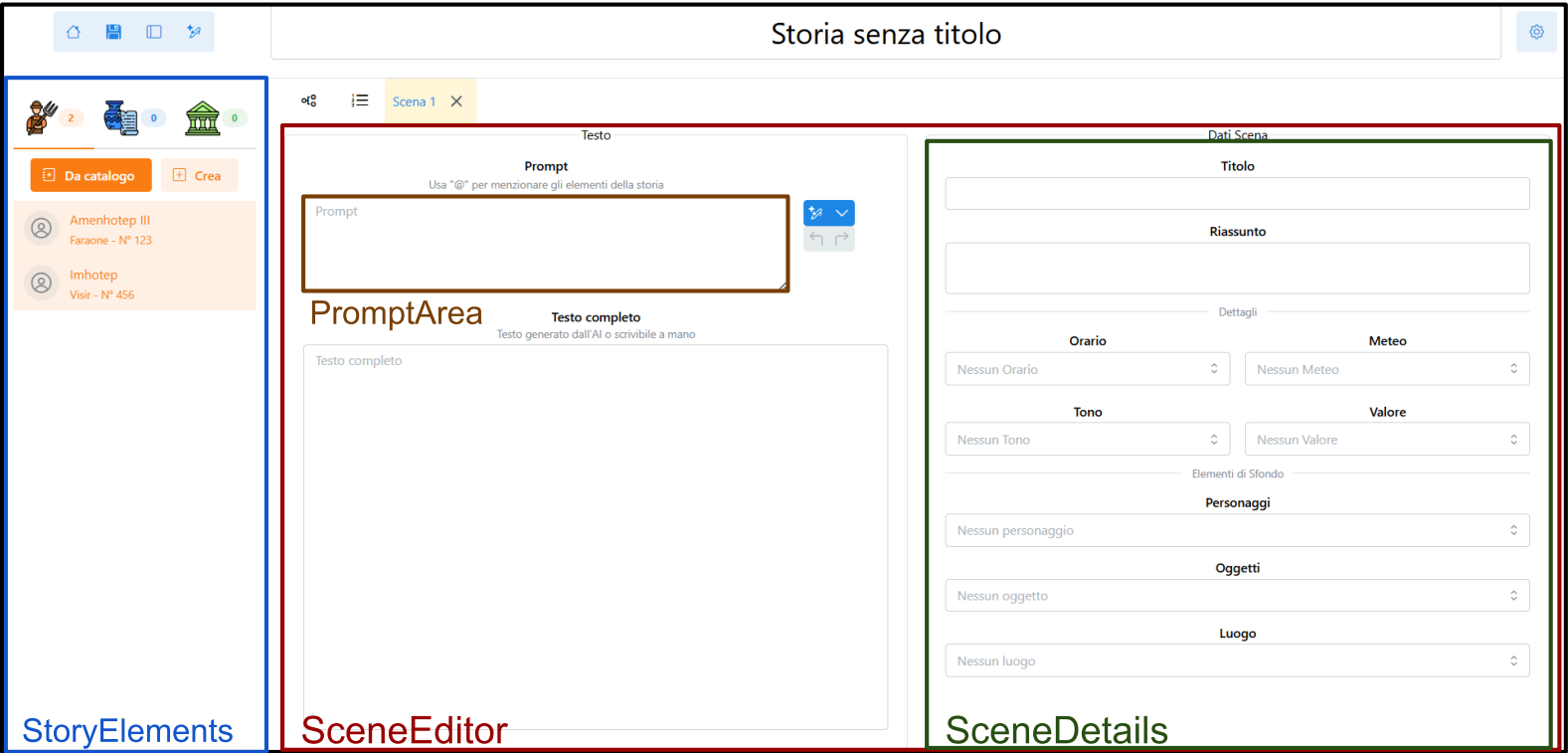


StoryEditor

SceneEditor.tsx

Contiene tutti i dettagli della singola Scena. Inoltre, permette di generare con l’AI il testo completo della Scena corrente, oppure di rigenerarlo anche per tutte le successive.

I contenuti di Prompt e Testo completo sono salvati in un UndoStack di dimensione indefinita. Quando viene salvata la Storia, sul file vengono salvati solo gli ultimi 10 elementi.

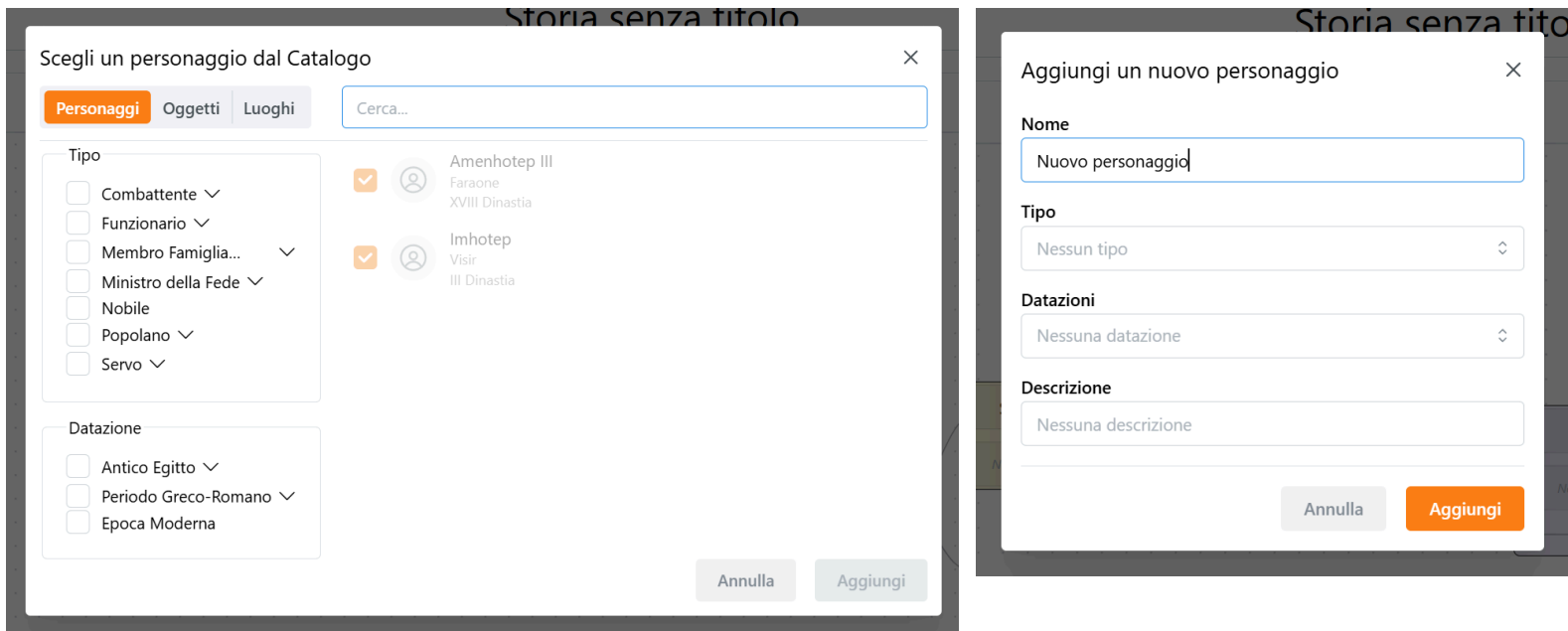


StoryEditor

Uguualmente, quando vengono aperti ChoiceNode e InfoNode, vengono visualizzati ChoiceEditor e InfoEditor.

StoryElements.tsx

Contiene gli Elementi della Storia suddivisi per tipo, e i modal per cercare nel DB e aggiungere un nuovo Elemento.



Ogni tab (Personaggi, Oggetti, Luoghi) apre DBBrowser e ElementModal preimpostati su quella categoria.

In DBBrowser è possibile cambiare categoria senza chiudere e riaprire il modal, tuttavia gli Elementi selezionati verranno deselezionati.