

SiVoDiM

Sintesi Vocale per Dispositivi Mobili



Specifica Tecnica

Versione	3.0.0
Redattori	Francesco Bizzaro Enrico Chiara Gino Zaidan Riccardo Rizzo
Verificatori	Alberto Andriolo
Responsabili	Gino Zaidan
Uso	Esterno
Lista di distribuzione	Stark Labs Prof. Tullio Vardanega, Prof. Riccardo Cardin

Documento contenente l'architettura logica che il gruppo Stark Labs ha steso per la realizzazione del progetto SiVoDiM

Registro delle modifiche

Versione	Data	Attività	Autori
3.0.0	06/07/2016	Accettazione	Gino Zaidan
2.1.0	06/07/2016	Revisione delle modifiche apportate	Alberto Andriolo
2.0.3	06/07/2016	Rimossa classe ShareAlgorithm in sezione 4	Gino Zaidan
2.0.2	06/07/2016	Rimosso design pattern Strategy per ShareAlgorithm in sezione 6	Gino Zaidan
2.0.1	06/07/2016	Rimosso il design pattern Adapter in sezione 6 (pattern ereditato da Android Studio)	Riccardo Rizzo
2.0.0	04/06/2016	Accettazione	Federico Rossetto
1.1.0	04/06/2016	Revisione delle modifiche apportate	Alberto Andriolo
1.0.7	09/06/2016	Contestualizzato l'uso dei design pattern	Enrico Chiara
1.0.6	09/06/2016	Corretto diagramma di sequenza per MVP	Enrico Chiara
1.0.5	01/06/2016	Inseriti diagrammi delle classi per i specifici design pattern utilizzati	Enrico Chiara
1.0.4	01/06/2016	Ampliate le descrizioni delle relazioni fra le classi: Package Libraries	Federico Rossetto
1.0.3	30/06/2016	Ampliate le descrizioni delle relazioni fra le classi: Package Drama	Francesco Bizzaro
1.0.2	27/06/2016	Ampliate le descrizioni delle relazioni fra le classi: Package Drama	Riccardo Rizzo
1.0.1	27/06/2016	Corretti i diagrammi di attività	Gino Zaidan
1.0.0	16/05/2016	Accettazione	Riccardo Rizzo
0.2.0	16/05/2016	Verifica	Alberto Andriolo
0.1.1	13/05/2016	Correzione errori segnalati	Enrico Chiara
0.1.0	13/05/2016	Verifica generale documento	Alberto Andriolo
0.0.14	12/05/2016	Stesura Appendice A	Enrico Chiara
0.0.13	12/05/2016	Stesura Sezione 8 Tracciamento	Gino Zaidan

0.0.12	12/05/2016	Stesura Sezione 7 Stime di fattibilità e di bisogno di risorse	Riccardo Rizzo
0.0.11	12/05/2016	Stesura Sezione 6 Design pattern	Riccardo Rizzo
0.0.10	11/05/2016	Stesura Sezione 5 Diagrammi di attività	Riccardo Rizzo
0.0.9	10/05/2016	Realizzazione dei diagrammi di attività	Francesco Bizzaro
0.0.8	09/05/2016	Stesura Sezione 4 Componenti e Classi	Alberto Andriolo, Francesco Bizzaro, Gino Zaidan
0.0.7	04/05/2016	Realizzazione dei diagrammi delle classi	Francesco Bizzaro, Federico Rossetto
0.0.6	05/05/2016	Stesura Sezione 3 Descrizione architettonica	Riccardo Rizzo
0.0.5	04/05/2016	Stesura delle sezioni: Sezione 2.5 Volley , Sezione 2.6 TextToSpeech	Gino Zaidan
0.0.4	04/05/2016	Stesura delle sezioni: Sezione 2.2 Android Studio , Sezione 2.4 FFmpeg	Riccardo Rizzo
0.0.3	04/05/2016	Stesura delle sezioni: Sezione 2.1 Android , Sezione 2.3 Java	Enrico Chiara
0.0.2	29/04/2016	Stesura Sezione 1 Introduzione	Riccardo Rizzo
0.0.1	29/04/2016	Stesura struttura documento	Riccardo Rizzo

Indice

1 Introduzione	1
1.1 Scopo del documento	1
1.2 Scopo del progetto	1
1.3 Glossario	1
1.4 Riferimenti	1
1.4.1 Normativi	1
1.4.2 Informativi	1
2 Tecnologie utilizzate	3
2.1 Android	3
2.1.1 Descrizione	3
2.1.2 Vantaggi	3
2.1.3 Svantaggi	3
2.2 Android Studio	3
2.2.1 Vantaggi	4
2.2.2 Svantaggi	4
2.3 Java	4
2.3.1 Vantaggi	4
2.3.2 Svantaggi	4
2.4 FFmpeg	5
2.4.1 Vantaggi	5
2.4.2 Svantaggi	5
2.5 Volley	5
2.5.1 Vantaggi	5
2.5.2 Svantaggi	5
2.6 TextToSpeech	6
2.6.1 Vantaggi	6
2.6.2 Svantaggi	6
2.7 TextToSpeechService	6
2.7.1 Vantaggi	6
2.7.2 Svantaggi	6
3 Descrizione architettura	7
3.1 Metodo e formalismo di specifica	7
3.2 Architettura generale	7
3.2.1 Model	8
3.2.2 View	8
3.2.3 Presenter	8
4 Componenti e Classi	9
4.1 Sivodim	9
4.1.1 Informazioni sul package	9
4.1.2 Descrizione	9
4.1.3 Package contenuti	9
4.2 Sivodim::Drama	10
4.2.1 Informazioni sul package	10
4.2.2 Descrizione	10

4.2.3	Package contenuti	10
4.2.4	Interazioni con altre componenti	10
4.3	Sivodim::Libraries	11
4.3.1	Informazioni sul package	11
4.3.2	Descrizione	11
4.3.3	Package contenuti	11
4.3.4	Interazioni con altre componenti	11
4.4	FFMPEG	11
4.4.1	Informazioni sul package	11
4.4.2	Descrizione	11
4.4.3	Interazioni con altre componenti	12
4.5	Volley	12
4.5.1	Informazioni sul package	12
4.5.2	Descrizione	12
4.5.3	Interazioni con altre componenti	12
4.5.4	Classi	12
4.6	tts	12
4.6.1	Informazioni sul package	12
4.6.2	Descrizione	12
4.6.3	Interazioni con altre componenti	13
4.6.4	Classi	13
4.7	Sivodim::Libraries::Model	13
4.7.1	Informazioni sul package	13
4.7.2	Descrizione	14
4.7.3	Package contenuti	14
4.7.4	Interazioni con altre componenti	14
4.8	Sivodim::Libraries::Model::Integration	14
4.8.1	Informazioni sul package	14
4.8.2	Descrizione	14
4.8.3	Interazioni con altre componenti	15
4.8.4	Classi	15
4.9	Sivodim::Libraries::Model::EngineManager	15
4.9.1	Informazioni sul package	15
4.9.2	Descrizione	15
4.9.3	Interazioni con altre componenti	16
4.9.4	Classi	16
4.10	Sivodim::Libraries::Model::Mivoq	17
4.10.1	Informazioni sul package	17
4.10.2	Descrizione	17
4.10.3	Interazioni con altre componenti	17
4.10.4	Classi	18
4.11	Sivodim::Libraries::Model::Voice	20
4.11.1	Informazioni sul package	20
4.11.2	Descrizione	20
4.11.3	Interazioni con altre componenti	20
4.11.4	Classi	20
4.12	Sivodim::Libraries::Presenter	22
4.12.1	Informazioni sul package	22
4.12.2	Descrizione	22

4.12.3	Interazioni con altre componenti	22
4.12.4	Classi	23
4.13	Sivodim::Libraries::View	25
4.13.1	Informazioni sul package	25
4.13.2	Descrizione	26
4.13.3	Interazioni con altre componenti	26
4.13.4	Classi	26
4.14	Sivodim::Drama::Model	29
4.14.1	Informazioni sul package	29
4.14.2	Descrizione	29
4.14.3	Package contenuti	29
4.14.4	Interazioni con altre componenti	29
4.15	Sivodim::Drama::Presenter	30
4.15.1	Informazioni sul package	30
4.15.2	Descrizione	30
4.15.3	Interazioni con altre componenti	30
4.15.4	Classi	30
4.16	Sivodim::Drama::View	37
4.16.1	Informazioni sul package	37
4.16.2	Descrizione	37
4.16.3	Interazioni con altre componenti	37
4.16.4	Classi	37
4.17	Sivodim::Drama::Model::Utilities	44
4.17.1	Informazioni sul package	44
4.17.2	Descrizione	44
4.17.3	Interazioni con altre componenti	45
4.17.4	Classi	45
4.18	Sivodim::Drama::Model::Chapter	48
4.18.1	Informazioni sul package	48
4.18.2	Descrizione	48
4.18.3	Interazioni con altre componenti	48
4.18.4	Classi	48
4.19	Sivodim::Drama::Model::Character	51
4.19.1	Informazioni sul package	51
4.19.2	Descrizione	51
4.19.3	Interazioni con altre componenti	51
4.19.4	Classi	52
4.20	Sivodim::Drama::Model::Screenplay	54
4.20.1	Informazioni sul package	54
4.20.2	Descrizione	54
4.20.3	Interazioni con altre componenti	54
4.20.4	Classi	54
5	Diagrammi di attività	58
5.1	Applicazione per gli sceneggiati	59
5.1.1	Attività di alto livello	59
5.1.2	Accesso al Drawer	60
5.1.3	Creazione sceneggiato	62
5.1.4	Gestione sceneggiato	63

5.1.5	Creazione capitolo	64
5.1.6	Condivisione sceneggiato	66
5.1.7	Esportazione sceneggiato	68
5.1.8	Gestione personaggio	70
5.1.9	Gestione capitolo	71
5.1.10	Creazione battuta	73
5.1.11	Gestione battuta	75
5.1.12	Cambio sfondo	77
5.1.13	Cambio soundtrack	79
5.1.14	Creazione personaggio	81
5.2	Applicazione di configurazione	83
5.2.1	Attività di alto livello	83
5.2.2	Creazione voce	85
5.2.3	Modifica voce	87
6	Design pattern	89
6.1	Design pattern architetturali	89
6.1.1	Model-view-presenter	89
6.2	Design pattern creazionali	91
6.2.1	Singleton	91
6.2.2	Builder	91
6.3	Design pattern strutturali	94
6.3.1	Facade	94
6.4	Design pattern comportamentali	94
6.4.1	Strategy	94
7	Stime di fattibilità e di bisogno di risorse	96
8	Tracciamento	97
8.1	Tracciamento Componenti - Requisiti	97
8.2	Tracciamento Requisiti - Componenti	105
A	Descrizione design pattern	117
A.1	Design pattern architetturali	117
A.1.1	Model-View-Presenter	117
A.2	Design pattern creazionali	118
A.2.1	Builder	118
A.2.2	Abstract Factory	119
A.2.3	Singleton	120
A.3	Design pattern strutturali	121
A.3.1	Adapter	121
A.3.2	Facade	122
A.4	Design pattern comportamentali	123
A.4.1	Strategy	123

Elenco delle figure

1	Diagramma dei package	7
2	Componente Sivodim	9
3	Componente Sivodim::Drama	10
4	Componente Sivodim::Libraries	11
5	Componente FFMPEG	11
6	Componente Volley	12
7	Componente tts	12
8	Componente Sivodim::Libraries::Model	13
9	Componente Sivodim::Libraries::Model::Integration	14
10	Componente Sivodim::Libraries::Model::EngineManager	15
11	Componente Sivodim::Libraries::Model::Mivoq	17
12	Componente Sivodim::Libraries::Model::Voice	20
13	Componente Sivodim::Libraries::Presenter	22
14	Componente Sivodim::Libraries::View	25
15	Componente Sivodim::Drama::Model	29
16	Componente Sivodim::Drama::Presenter	30
17	Componente Sivodim::Drama::View	37
18	Componente Sivodim::Drama::Model::Utilities	44
19	Componente Sivodim::Drama::Model::Chapter	48
20	Componente Sivodim::Drama::Model::Character	51
21	Componente Sivodim::Drama::Model::Screenplay	54
22	Diagramma delle attività di alto livello dell’applicazione degli sceneggiati	59
23	Diagramma delle attività per l’accesso al Drawer	60
24	Diagramma delle attività per la creazione di uno sceneggiato	62
25	Diagramma delle attività per la gestione di uno sceneggiato	63
26	Diagramma delle attività per la creazione di un capitolo	64
27	Diagramma delle attività per la condivisione	66
28	Diagramma delle attività per l’esportazione	68
29	Diagramma delle attività per la gestione di un personaggio	70
30	Diagramma delle attività per la gestione di uno capitolo	71
31	Diagramma delle attività per la creazione di una battuta	73
32	Diagramma delle attività per la gestione di una battuta	75
33	Diagramma delle attività per la modifica dello sfondo di un capitolo	77
34	Diagramma delle attività per la modifica del sottofondo sonoro di un capitolo	79
35	Diagramma delle attività per la creazione di un nuovo personaggio	81
36	Diagramma delle attività di alto livello dell’app di configurazione	83
37	Diagramma delle attività per la creazione di una voce	85
38	Diagramma delle attività per la modifica di una voce	87
39	Diagramma di sequenza per la creazione di uno sceneggiato	90
40	Utilizzo di Singleton	91
41	Utilizzo di Builder	92
42	Utilizzo di Builder	93
43	Utilizzo di Builder	93
44	Utilizzo di Facade	94
45	Utilizzo di Strategy	95
46	Struttura del design pattern MVP	117
47	Struttura del design pattern Builder	118

48	Struttura del design pattern Abstract Factory	119
49	Struttura del design pattern Singleton	120
50	Struttura del design pattern Adapter	121
51	Struttura del design pattern Facade	122
52	Struttura del design pattern Strategy	123

1 Introduzione

1.1 Scopo del documento

Questo documento definisce la progettazione ad alto livello che i membri del gruppo Stark Labs hanno stilato per la realizzazione del progetto SiVoDiM: Sintesi Vocale per Dispositivi Mobili. A seguire, viene presentata l'architettura logica secondo la quale verranno organizzate le varie componenti software e vengono descritti i *design pattern_G* utilizzati.

1.2 Scopo del progetto

Lo scopo del progetto risiede nello sviluppo di un'applicazione utile a dimostrare efficacemente le potenzialità del motore di sintesi vocale FA-TTS_G, realizzato dall'azienda MIVOQ s.r.l. e messo a disposizione del gruppo di lavoro. Si devono realizzare due applicazioni per sistemi Android_G:

- **Applicazione di configurazione:** deve permettere all'utente di interfacciarsi direttamente con il sistema operativo per configurare, salvare e modificare le voci ereditate dal motore di sintesi FA-TTS di MIVOQ;
- **Applicazione per la creazione di sceneggiati:** permette la creazione e il salvataggio di racconti e sceneggiati, che possono essere esportati in formato audio attraverso l'utilizzo del motore FA-TTS.

Entrambe le applicazioni devono interfacciarsi con un modulo di basso livello:

- **Modulo di sistema:** permette di interfacciarsi tramite connessione di rete al motore FA-TTS. Fornisce una libreria contenente tutte le funzionalità offerte da FA-TTS, utile nell'ottica di un riuso futuro del *software*.

Lo sviluppo di tutte e quattro le suddette componenti è a carico del gruppo Stark Labs.

1.3 Glossario

Al fine di aumentare la comprensione del testo ed evitare eventuali ambiguità, viene fornito un glossario (*Glossario v2.0.0*) contenente le definizioni degli acronimi e dei termini tecnici utilizzati nei documenti. Ogni vocabolo che ha un riferimento contenuto nel glossario è contrassegnato dal pedice "_G".

1.4 Riferimenti

1.4.1 Normativi

- *Norme di Progetto v2.0.0*;
- *Analisi dei Requisiti v2.0.0*.

1.4.2 Informativi

- Slide dell'insegnamento Ingegneria del Software modulo A & B (*Design pattern_G*):
<http://www.math.unipd.it/~tullio/IS-1/2015/>
<http://www.math.unipd.it/~rcardin/sweb.html>

- Documentazione per Android Studio e sviluppo su Android_G:
<http://developer.android.com/tools/studio/index.html>
<http://developer.android.com/guide/index.html>
- Documentazione Java 7:
<http://docs.oracle.com/javase/7/docs/>
- Sito ufficiale di FFmpeg:
<https://ffmpeg.org/>
- Documentazione Volley su Android:
<http://developer.android.com/training/volley/index.html>
- Documentazione TextToSpeech su Android:
<http://developer.android.com/reference/android/speech/tts/TextToSpeech.html>
- Documentazione TextToSpeechService su Android:
<http://developer.android.com/reference/android/speech/tts/TextToSpeechService.html>
- Documentazione SynthesisRequest su Android:
<http://developer.android.com/reference/android/speech/tts/SynthesisRequest.html>

2 Tecnologie utilizzate

In questa sezione vengono descritte le tecnologie utilizzate per lo sviluppo di SiVoDiM e viene presentata una breve analisi dei vantaggi e degli svantaggi riscontrati nel loro uso.

2.1 Android

2.1.1 Descrizione

Android_G è un sistema operativo sviluppato da Google per dispositivi mobili, dispositivi indossabili e televisori. Le versioni di riferimento sono la 4.0 e superiori. L'utilizzo di questa tecnologia è conseguenza di una maggiore familiarità dei membri del gruppo, nei confronti di questo sistema operativo, rispetto a quelli concorrenti.

2.1.2 Vantaggi

Android è stato scelto come sistema operativo di riferimento per i seguenti motivi:

- Disponibile su un vasto numero di dispositivi;
- Sistema operativo per dispositivi mobili più diffuso al mondo;
- Quasi completamente *open-source*_G.

2.1.3 Svantaggi

Lo sviluppo su Android presenta le seguenti criticità:

- Estremamente frammentato a causa di una moltitudine di versioni del software su differenti dispositivi;
- Necessità di sviluppare applicazioni per dispositivi con caratteristiche tecniche molto differenti.

2.2 Android Studio

Android Studio è l'IDE_G ufficiale per lo sviluppo di applicazioni Android_G, basato su IntelliJ IDEA. In aggiunta all'editor e agli strumenti di sviluppo ereditati da IntelliJ, Android Studio offre le seguenti caratteristiche pensate per lo sviluppo specifico di applicazioni Android_G:

- **Gradle:** supporto alla compilazione basata su Gradle_G;
- **Templates:** disponibili dei template_G predefiniti per lo sviluppo di applicazioni che fanno uso di componenti ed elementi di design comuni all'ambiente Android;
- **Layout editor:** l'editor integrato permette di personalizzare le componenti dell'UI_G attraverso un'interfaccia *drag-and-drop*_G con possibilità di testare l'UI su diverse configurazioni di schermo;
- **Google Cloud Platform:** supporto integrato alla piattaforma *cloud*_G di Google, che permette l'integrazione con i servizi Google Cloud Messaging e App Engine.
- **Multipiattaforma:** disponibile per Windows_G, OS X_G e Linux_G.

2.2.1 Vantaggi

Android Studio è stato scelto per i seguenti motivi:

- Supportato ufficialmente da Google per lo sviluppo su Android_G;
- Maggiore facilità nella manutenibilità e nel riuso del codice su Android;
- Esportazione rapida e intuitiva delle apk_G per varie versioni di Android;
- Disponibile gratuitamente sotto licenza Apache 2.0;
- Familiarità del team nell'uso di IntelliJ;
- Multipiattaforma.

2.2.2 Svantaggi

L'utilizzo di Android Studio presenta le seguenti criticità:

- Estraneo alle conoscenze del gruppo: i membri dovranno studiarne le modalità di utilizzo;
- Sviluppo mirato al solo sistema operativo Android.

2.3 Java

Java è uno dei più diffusi linguaggi di programmazione orientato agli oggetti. Oltre ad avere un'ottima documentazione è supportato da una moltitudine di librerie. Viene utilizzato per lo sviluppo delle applicazioni incluse nel progetto SiVoDiM e per la realizzazione della libreria richiesta.

2.3.1 Vantaggi

Java è stato scelto per i seguenti motivi:

- Linguaggio più diffuso e maggiormente utilizzato per lo sviluppo di applicazioni in ambiente Android_G;
- Linguaggio utilizzato da Android Studio;
- Ampia documentazione disponibile;
- Ampio numero di librerie disponibili;
- Portabilità su diversi sistemi operativi.

2.3.2 Svantaggi

L'utilizzo di Java presenta le seguenti criticità:

- Java è un linguaggio noto per essere particolarmente verboso;
- Poca esperienza del team nella programmazione in Java.

2.4 FFmpeg

FFmpeg è un software gratuito che produce librerie e programmi per la gestione di dati multimediali. Il software include:

- **ffmpeg**: strumento da riga di comando per convertire un file video da un formato a un altro;
- **ffplay**: riproduttore multimediale basato su librerie FFmpeg;
- **libavcodec**: libreria gratuita e *open-source_G* di *codec* per la codifica e la decodifica di dati audio e video;
- **libavformat**: libreria contenente *parser* e generatori condivisi per tutti i formati audio e video.

2.4.1 Vantaggi

L'utilizzo di librerie FFmpeg potrebbe semplificare il processo di esportazione degli sceneggiati nei formati audio e video.

2.4.2 Svantaggi

FFmpeg nasce come **non nativo** per lo sviluppo in Java, pertanto dovrà essere opportunamente integrato nel progetto attraverso un componente Adapter.

2.5 Volley

Volley è una libreria HTTP_G che semplifica e velocizza le operazioni di connettività delle applicazioni Android_G. Volley offre le seguenti caratteristiche:

- *Scheduling* automatico delle richieste di rete;
- Connessioni di rete concorrenti multiple;
- Ordinamento forte che permette di popolare correttamente e facilmente l'interfaccia utente con dati raccolti in modo asincrono dalla rete;
- Strumenti di *debugging* e tracciabilità.

2.5.1 Vantaggi

La libreria Volley è stata scelta per i seguenti motivi:

- Si integra facilmente con tutti i protocolli;
- Permette di concentrarsi sullo sviluppo della logica che specifica l'applicazione.

2.5.2 Svantaggi

L'utilizzo di Volley presenta le seguenti criticità:

- Non è adatta per il *download* di dati di grandi dimensioni;
- Non è adatta per l'esecuzione di operazioni di *streaming_G*.

2.6 TextToSpeech

TextToSpeech è una classe offerta dal sistema operativo Android_G che rende disponibile la sintesi vocale per una riproduzione immediata o per la creazione di un file audio.

2.6.1 Vantaggi

L'utilizzo di TextToSpeech offre i seguenti benefici:

- Utile per la riproduzione e la creazione di file audio in caso di assenza di connessione per accedere al servizio offerto da MIVOQ s.r.l.;
- Disponibile internamente ad Android Studio.

2.6.2 Svantaggi

Documentazione scarna. Le poche informazioni disponibili in rete ne rendono più ostico lo studio da parte dei membri del *team*.

2.7 TextToSpeechService

TextToSpeechService è una classe astratta offerta dal sistema operativo Android_G per l'implementazione del motore di sintesi vocale nativo.

2.7.1 Vantaggi

L'utilizzo di TextToSpeechService offre metodi per la gestione della lingua.

2.7.2 Svantaggi

Documentazione scarna. Le poche informazioni disponibili in rete ne rendono più ostico lo studio da parte dei membri del *team*.

3 Descrizione architettura

3.1 Metodo e formalismo di specifica

Per descrivere l'architettura dell'applicazione viene utilizzato un approccio di tipo *top-down*, attraverso il quale si specifica l'architettura a partire da una visione generale del sistema che solo successivamente viene rifinita con l'aggiunta di maggiori dettagli progettuali.

Secondo tale logica, la prima parte del documento contiene la descrizione dei *package_G*; seguono le descrizioni delle singole classi di cui sono specificati il tipo, l'obiettivo, la funzione e le relazioni in ingresso e uscita.

Successivamente vengono illustrati degli esempi d'uso dei *design pattern_G* applicati all'architettura di sistema, la cui spiegazione generale è riportata in Appendice A. Per rappresentare graficamente *package* e classi si utilizza il formalismo dato da UML_G 2.1.

3.2 Architettura generale

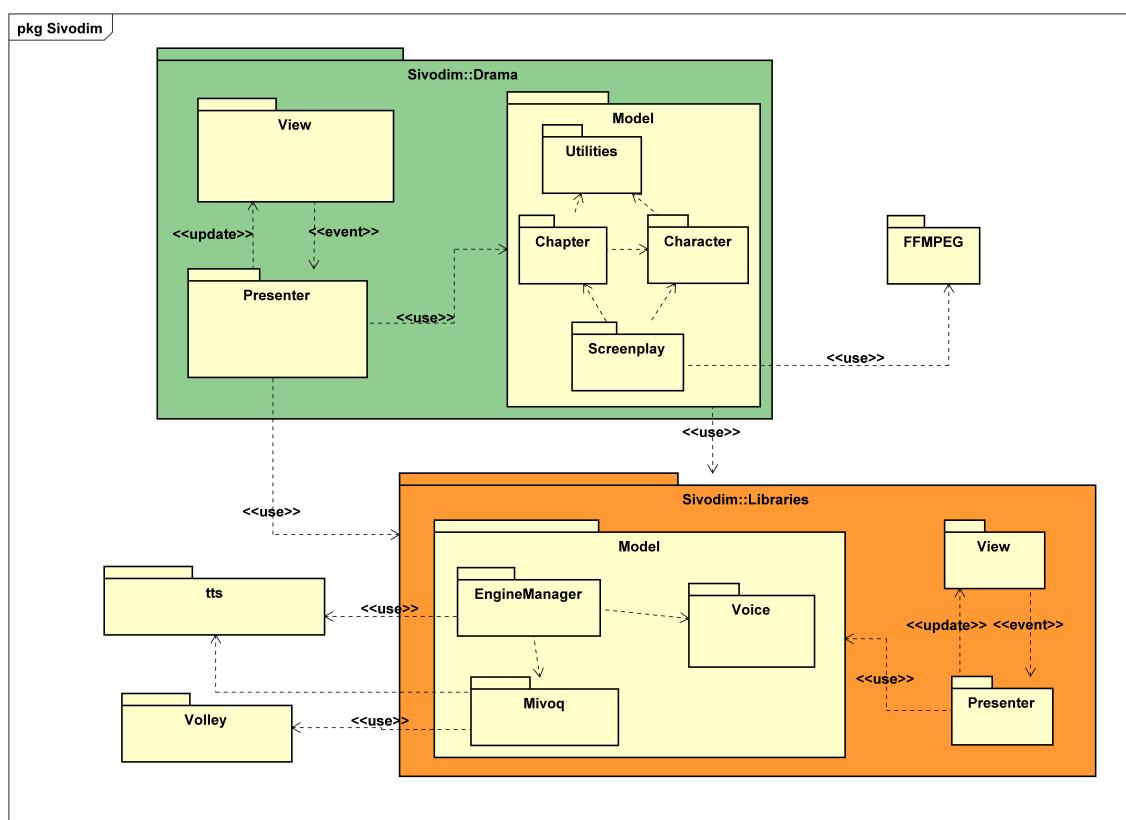


Figura 1: Diagramma dei package

L'architettura del software segue il *design pattern_G* model-view-presenter (MVP), una derivazione del model-view-controller (MVC), utilizzato principalmente per la realizzazione di interfacce utente. MVP facilita l'automatizzazione dei test di unità e migliora la separazione delle responsabilità nella *presentation logic_G*:

- **Model:** modulo che definisce i dati da visualizzare;
- **Presenter:** funge da *man-in-the-middle_G* tra il *model* e la *view*. Ricava i dati dal *model* e li elabora per visualizzarli nella *view*;

- **View:** interfaccia passiva che visualizza le informazioni del *model* passate dal *presenter* e trasferisce i comandi dell'utente (attraverso eventi) al *presenter* per agire sui dati.

Il diagramma in Figura 1 rappresenta l'architettura ad alto livello delle due componenti *software*: applicazione per gli sceneggiati (verde) e modulo di sistema (arancione). Vengono indicati i $package_G$ e le relazioni fra di essi. Per ogni componente si segue il *design pattern* MVP.

3.2.1 Model

Nel *model* sono presenti gli oggetti che rappresentano:

- Lo sceneggiato composto da un elenco di capitoli e personaggi;
- I personaggi da associare alle battute dei capitoli;
- I capitoli che compongono ogni sceneggiato;
- Le battute che compongono ogni capitolo.

Il *model* mette a disposizione tutti i metodi utili a creare, modificare e gestire gli oggetti sopra descritti. Tutte le informazioni e le operazioni contenute nel *model* sono gestite dal *presenter* per comunicare con la *view*.

3.2.2 View

Nella *view* è contenuta la grafica attraverso cui l'utente interagisce con le funzionalità offerte dal *software*. Ogni schermata dell'applicazione corrisponde a una *view* specifica. La richiesta dei dati verso il *model* viene effettuata attraverso il *presenter*, che si occupa di reperirli ed aggiornare la *view*.

3.2.3 Presenter

Il *presenter* funge da *man-in-the-middle* utile per istanziare gli oggetti del *model* da esporre all'*Activity*, con l'obiettivo di diminuire la formazione di errori relativi alla gestione della memoria del dispositivo. Inoltre si evita di fondere la logica dell'applicazione con la sua rappresentazione grafica, migliorando così stabilità e manutenibilità del *software*.

4 Componenti e Classi

Di seguito si riportano tutte le componenti e le classi.

4.1 Sivodim

4.1.1 Informazioni sul package

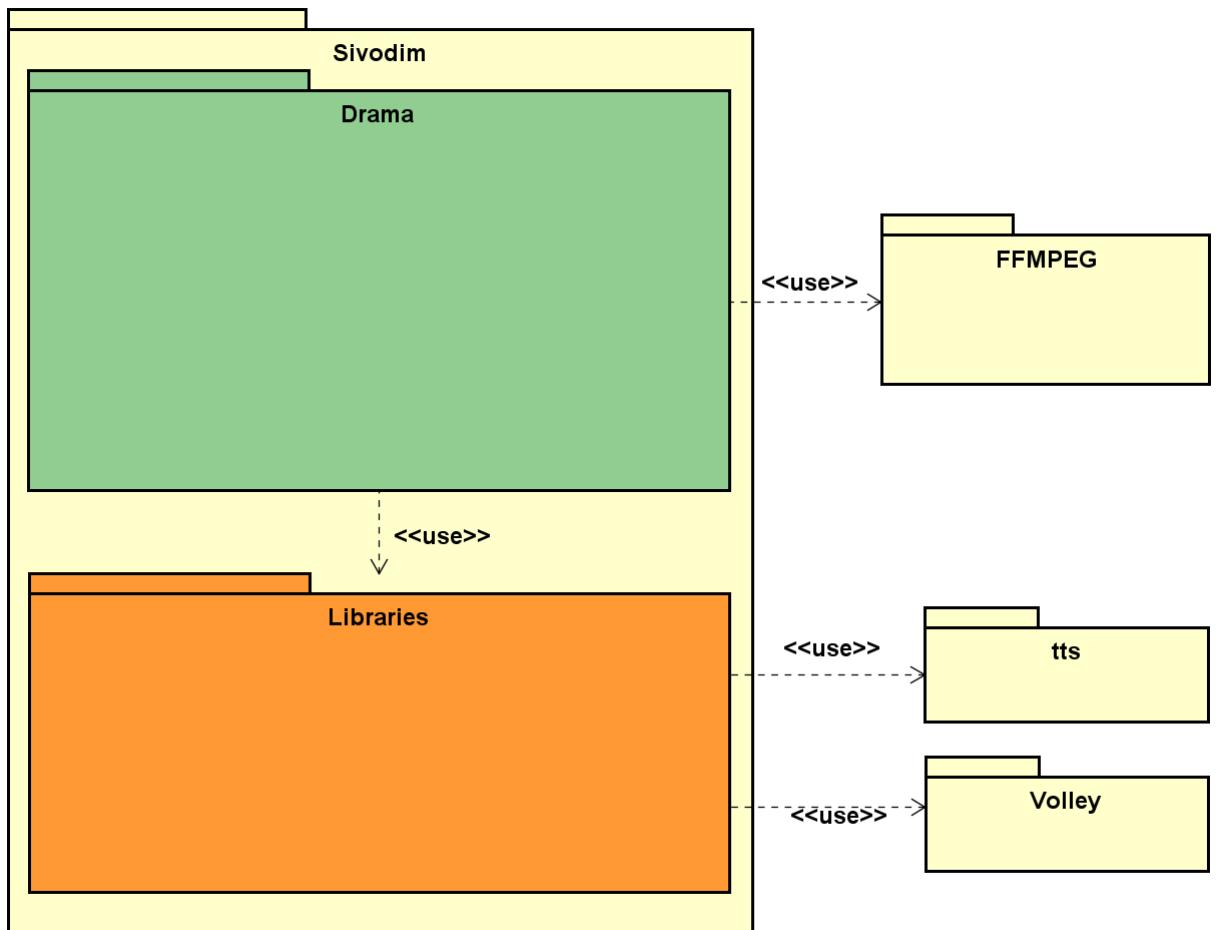


Figura 2: Componente Sivodim

4.1.2 Descrizione

Il $package_G$ rappresenta una visione ad alto livello dell'intero progetto.

4.1.3 Package contenuti

- Sivodim::Drama;
- Sivodim::Libraries.

4.2 Sivodim::Drama

4.2.1 Informazioni sul package

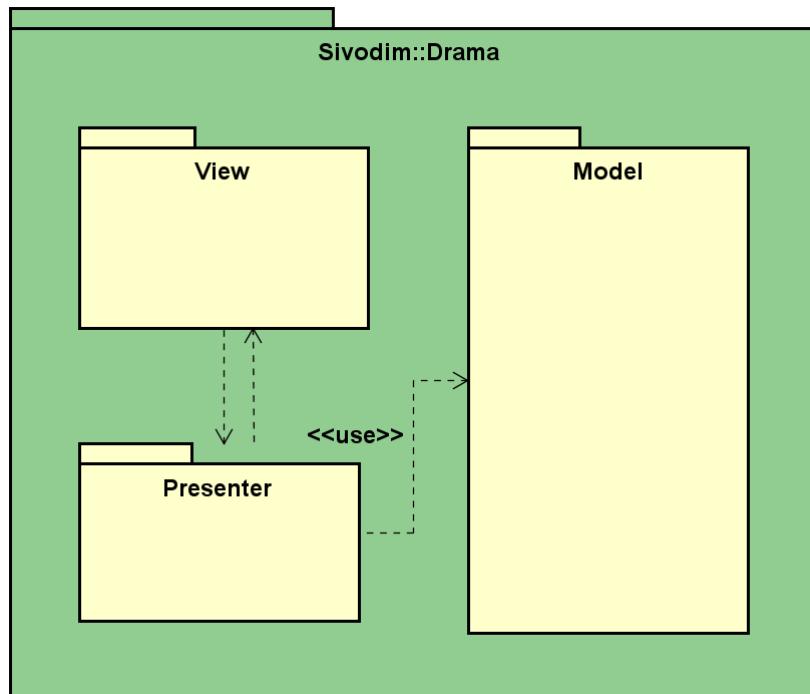


Figura 3: Componente Sivodim::Drama

4.2.2 Descrizione

Il $package_G$ rappresenta l'applicazione per gli sceneggiati, strutturata secondo il $pattern$ MVP_G.

4.2.3 Package contenuti

- Sivodim::Drama::Model;
- Sivodim::Drama::Presenter;
- Sivodim::Drama::View.

4.2.4 Interazioni con altre componenti

- Sivodim::Libraries.

4.3 Sivodim::Libraries

4.3.1 Informazioni sul package

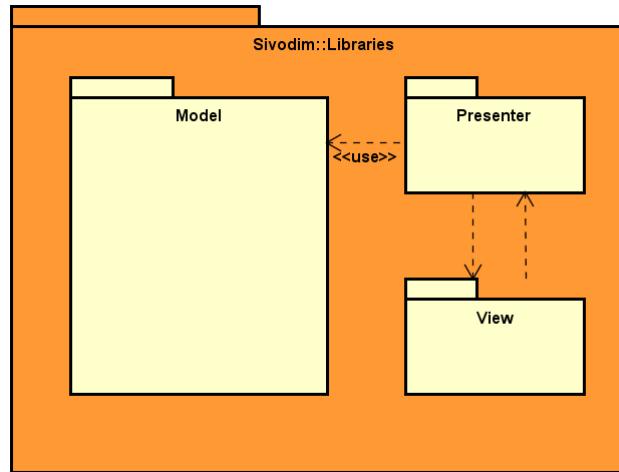


Figura 4: Componente Sivodim::Libraries

4.3.2 Descrizione

Il $package_G$ racchiude le librerie per la comunicazione con il server di MIVOQ s.r.l. e l'integrazione dei servizi di $Text-To-Speech_G$ offerti, con il sistema Android_G. È prevista un'interfaccia grafica.

4.3.3 Package contenuti

- Sivodim::Libraries::Model;
- Sivodim::Libraries::Presenter;
- Sivodim::Libraries::View.

4.3.4 Interazioni con altre componenti

- Sivodim::Drama.

4.4 FFMPEG

4.4.1 Informazioni sul package

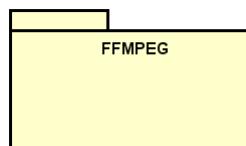


Figura 5: Componente FFMPEG

4.4.2 Descrizione

Si tratta di una libreria esterna, che viene utilizzata dall'applicazione degli sceneggiati.

4.4.3 Interazioni con altre componenti

- Sivodim::Drama::Model::Screenplay.

4.5 Volley

4.5.1 Informazioni sul package

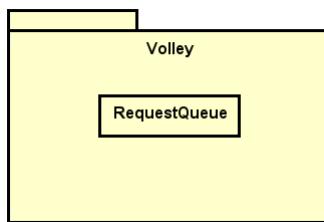


Figura 6: Componente Volley

4.5.2 Descrizione

Si tratta di una libreria usata per la comunicazione con il server di MIVOQ s.r.l..

4.5.3 Interazioni con altre componenti

- Sivodim::Libraries::Model::Mivoq.

4.5.4 Classi

- **Volley::RequestQueue**

- **Descrizione** - Classe del pacchetto java Volley, che offre funzionalità per gestire le connessioni $http_G$;
- **Utilizzo** - Viene utilizzata per connettersi al server di MIVOQ s.r.l.;
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl.

4.6 tts

4.6.1 Informazioni sul package

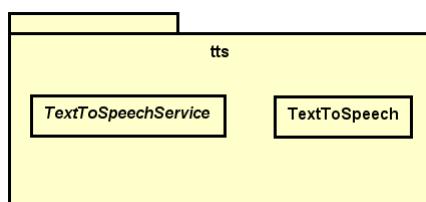


Figura 7: Componente tts

4.6.2 Descrizione

Libreria usata dalla componente Libraries per la gestione del *Text-To-Speech_G* in Android_G.

4.6.3 Interazioni con altre componenti

- Sivodim::Libraries::Model::EngineManager;
- Sivodim::Libraries::Model::Mivoq.

4.6.4 Classi

- **tts::TextToSpeechService**
 - **Descrizione** - Classe astratta interna di Android_G utilizzata per implementare gli engine_G del sistema;
 - **Utilizzo** - Viene usata per definire il contratto per l'implementazione degli engine_G del sistema;
 - **Classi figlie**
 - * Sivodim::Libraries::Model::Mivoq::MivoqTTSService.
- **tts::TextToSpeech**
 - **Descrizione** - Classe interna di Android_G che implementa i servizi di Text-To-Speech_G del sistema;
 - **Utilizzo** - Viene usata per rendere disponibili i servizi di Text-To-Speech del sistema Android_G ;
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::EngineManager::EngineImpl** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::Model::EngineManager::EngineImpl.

4.7 Sivodim::Libraries::Model

4.7.1 Informazioni sul package

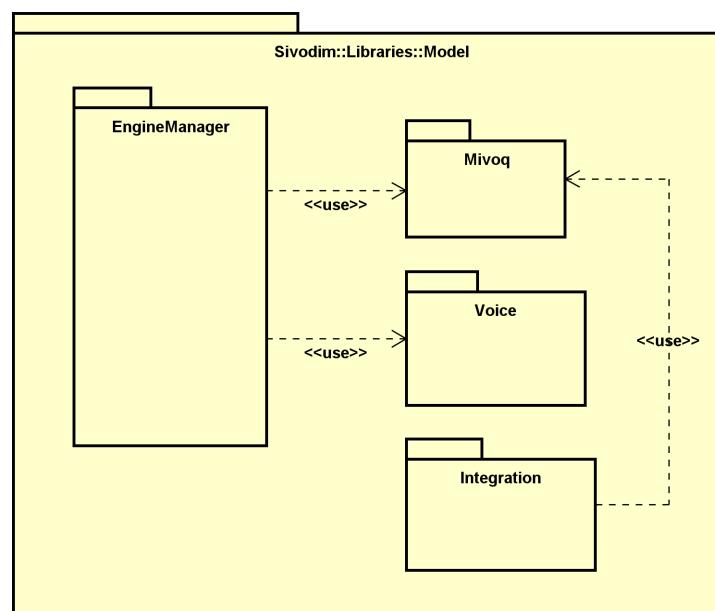


Figura 8: Componente Sivodim::Libraries::Model

4.7.2 Descrizione

Il $package_G$ racchiude la gestione del servizio di sintesi vocale ed espone attraverso un'unica interfaccia i suoi servizi. Al suo interno vengono definite anche le voci e la loro *business logic_G*.

4.7.3 Package contenuti

- Sivodim::Libraries::Model::Integration;
- Sivodim::Libraries::Model::EngineManager;
- Sivodim::Libraries::Model::Mivoq;
- Sivodim::Libraries::Model::Voice.

4.7.4 Interazioni con altre componenti

- Sivodim::Drama;
- Volley;
- tts.

4.8 Sivodim::Libraries::Model::Integration

4.8.1 Informazioni sul package

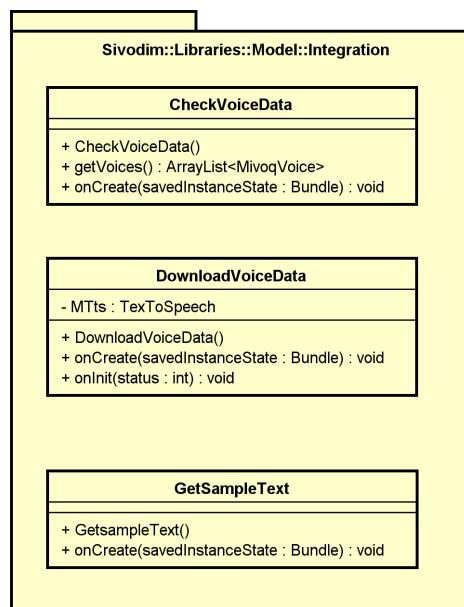


Figura 9: Componente Sivodim::Libraries::Model::Integration

4.8.2 Descrizione

Il $package_G$ gestisce le operazioni di interazione fra l'utente e l'interfaccia grafica delle impostazioni di Android_G, per la configurazione del motore di sintesi FA-TTS_G con il sistema.

4.8.3 Interazioni con altre componenti

- Sivodim::Libraries::Model::Mivoq;

4.8.4 Classi

- **Sivodim::Libraries::Model::Integration::CheckVoiceData**

- **Descrizione** - Classe che comunica al sistema tutte le voci disponibili create dall'utente;
- **Utilizzo** - Viene utilizzata per rendere disponibili al sistema le voci create dall'utente con l'applicazione di configurazione.

- **Sivodim::Libraries::Model::Integration::DownloadVoiceData**

- **Descrizione** - Classe per il funzionamento del servizio di sintesi vocale del sistema;
- **Utilizzo** - Viene utilizzata per il download di dati necessari al servizio di sintesi vocali.

- **Sivodim::Libraries::Model::Integration::GetSampleText**

- **Descrizione** - Classe che si occupa di trasmettere al sistema il testo di esempio da leggere per le varie lingue;
- **Utilizzo** - Classe che si occupa di trasmettere al sistema il testo di esempio da leggere per le varie lingue.

4.9 Sivodim::Libraries::Model::EngineManager

4.9.1 Informazioni sul package

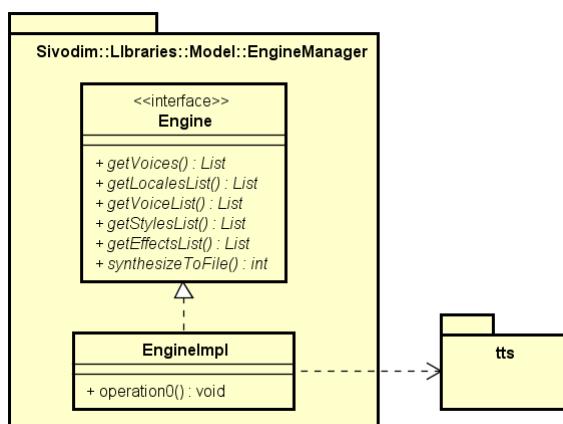


Figura 10: Componente Sivodim::Libraries::Model::EngineManager

4.9.2 Descrizione

Il *package_G* gestisce la sintesi vocale, scegliendo di utilizzare i servizi offerti da MIVOQ s.r.l. o quelli interni al dispositivo, nel caso i primi non fossero disponibili.

4.9.3 Interazioni con altre componenti

- tts;
- Sivodim::Libraries::Model::Mivoq;
- Sivodim::Libraries::Model::Voice;
- Sivodim::Libraries::Presenter;
- Sivodim::Drama::Model::Chapter;
- Sivodim::Drama::Model::Character;
- Sivodim::Drama::Model::Screenplay.

4.9.4 Classi

- **Sivodim::Libraries::Model::EngineManager::Engine**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il TTS_G di sistema utilizzando se disponibile il servizio MIVOQ s.r.l.;
 - **Utilizzo** - Viene usata per definire il contratto attraverso cui utilizzare i servizi di TTS;
 - **Classi figlie**
 - * Sivodim::Libraries::Model::EngineManager::Enginelmpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Model::Screenplay::ScreenplayImpl. La classe utilizza l'interfaccia per gestire le voci personalizzate e la sintesi delle battute;
 - * **Sivodim::Drama::Model::Character::CharacterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Model::Character::CharacterImpl. La classe utilizza l'interfaccia per gestire le voci personalizzate;
 - * **Sivodim::Drama::Model::Chapter::SpeechImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Model::Chapter::SpeechImpl. La classe utilizza l'interfaccia per gestire le voci personalizzate e la sintesi delle battute;
 - * **Sivodim::Libraries::Presenter::HomePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Presenter::HomePresenterImpl. La classe utilizza l'interfaccia per gestire le voci personalizzate dall'utente;
 - * **Sivodim::Libraries::Presenter::VoicePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Presenter::VoicePresenterImpl. La classe utilizza l'interfaccia per gestire le voci personalizzate dall'utente.
- **Sivodim::Libraries::Model::EngineManager::Enginelmpl**
 - **Descrizione** - Classe che implementa l'interfaccia Engine;
 - **Utilizzo** - Implementa il contratto della classe Engine che gestisce la sintesi vocale per la libreria Sivodim;
 - **Classi ereditate**

* Sivodim::Libraries::Model::EngineManager::Engine.

- Relazioni con altre classi

- * **Sivodim::Libraries::Model::Voice::MivoqVoice** - Relazione uscente, riferimento ad una classe che concretizza la classe Sivodim::Libraries::Model::Voice::MivoqVoice. Utilizza la classe per gestire le voci personalizzate e applicare gli effetti alla sintesi vocale;
- * **tts::TextToSpeech** - Relazione uscente, riferimento alla classe tts::TextToSpeech. La classe utilizza la relazione per dare la possibilità di utilizzare la sintesi vocale anche senza una connessione internet, sfruttando la sintesi vocale di sistema;
- * **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton** - Relazione uscente, riferimento alla classe Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton. La classe utilizza il riferimento per utilizzare la sintesi vocale offerta da MIVOQ s.r.l..

4.10 Sivodim::Libraries::Model::Mivoq

4.10.1 Informazioni sul package

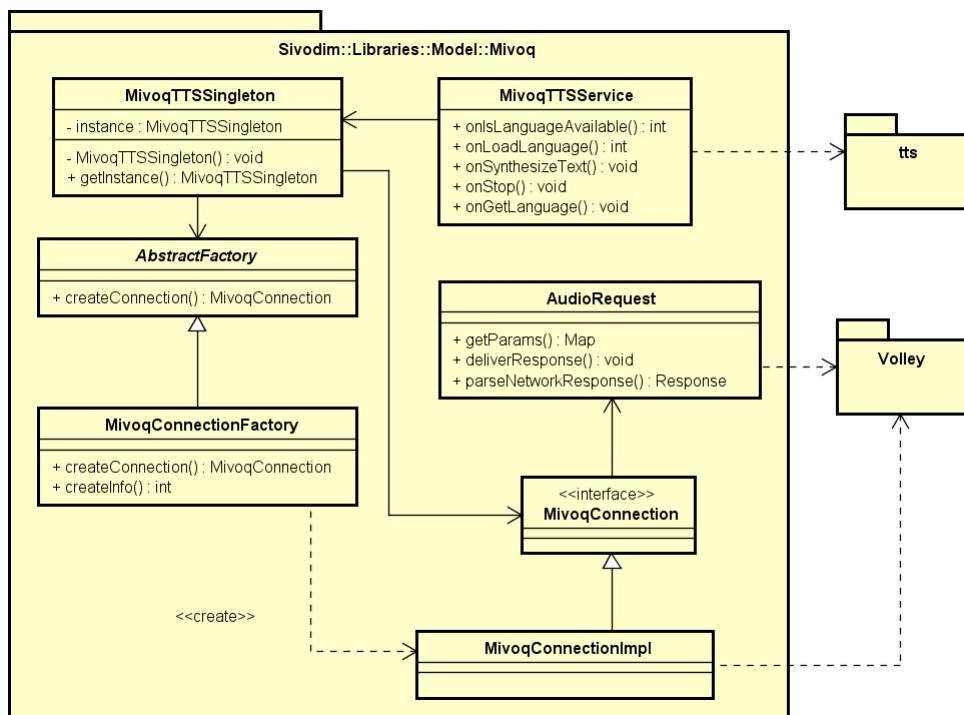


Figura 11: Componente Sivodim::Libraries::Model::Mivoq

4.10.2 Descrizione

Il pacchetto gestisce la connessione con MIVOQ s.r.l. per l'utilizzo dei servizi di sintesi offerti.

4.10.3 Interazioni con altre componenti

- Volley;

- tts;
- Sivodim::Libraries::Model::EngineManager.

4.10.4 Classi

- **Sivodim::Libraries::Model::Mivoq::AbstractFactory**
 - **Descrizione** - Classe astratta che rappresenta il contratto della classe Factory che ha il compito di creare istanze di oggetti di tipo MivoqConnection;
 - **Utilizzo** - Viene utilizzata per creare istanze di oggetti di tipo MivoqConnection;
 - **Classi figlie**
 - * Sivodim::Libraries::Model::Mivoq::MivoqConnectionFactory.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton. La classe utilizza l'interfaccia per istanziare oggetti di tipo MivoqConnection.
- **Sivodim::Libraries::Model::Mivoq::MivoqConnectionFactory**
 - **Descrizione** - Classe che implementa la classe AbstractFactory per la creazione di oggetti di tipo MivoqConnectionImpl;
 - **Utilizzo** - Viene utilizzato per istanziare oggetti concreti di tipo MivoqConnectionImpl;
 - **Classi ereditate**
 - * Sivodim::Libraries::Model::Mivoq::AbstractFactory.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl** - Relazione uscente, la classe Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl viene istanziata dagli oggetti di questa classe. La classe utilizza il *design pattern Singleton*.
- **Sivodim::Libraries::Model::Mivoq::MivoqConnection**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la connessione ai server MIVOQ s.r.l.;
 - **Utilizzo** - Viene usata per definire il contratto di interazione tra l' $engine_G$ e il server;
 - **Classi figlie**
 - * Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton. La classe viene utilizzata per gestire la connessione ai server MIVOQ s.r.l..
- **Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl**

- **Descrizione** - Classe che implementa l'interfaccia MivoqConnection;
- **Utilizzo** - Implementa il contratto di interazione con il protocollo attuale di connessione al server di MIVOQ s.r.l.;
- **Classi ereditate**
 - * Sivodim::Libraries::Model::Mivoq::MivoqConnection.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqConnectionFactory** - Relazione entrante, la classe Sivodim::Libraries::Model::Mivoq::MivoqConnectionFactory istanzia gli oggetti di questa classe;
 - * **Volley::RequestQueue** - Relazione uscente, utilizza la classe RequestQueue per la connessione ai server di MIVOQ s.r.l.. Utilizza il riferimento per gestire una coda condivisa per l'inoltro delle richieste di sintesi.
- **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton**
 - **Descrizione** - Classe Singleton_G che implementa l'*engine_G* di sintesi vocale;
 - **Utilizzo** - Viene utilizzato per la sintesi vocale attraverso i server MIVOQ s.r.l.;
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::AbstractFactory** - Relazione uscente, con la classe Sivodim::Libraries::Model::Mivoq::AbstractFactory. Utilizza la classe per istanziare le connessioni ai server MIVOQ s.r.l., per gestire le richieste di sintesi vocale;
 - * **Sivodim::Libraries::Model::Mivoq::MivoqConnection** - Relazione uscente, con la classe Sivodim::Libraries::Model::Mivoq::MivoqConnection. Utilizza la classe per gestire ogni singola connessione e richiesta di sintesi;
 - * **Sivodim::Libraries::Model::Mivoq::MivoqTTSService** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Mivoq::MivoqTTSService. La classe viene riferita per rendere il servizio di sintesi vocale al sistema operativo Android_G;
 - * **Sivodim::Libraries::Model::EngineManager::EngineImpl** - Relazione entrante, la classe è riferita da Sivodim::Libraries::Model::EngineManager::EngineImpl. La classe viene utilizzata per rendere disponibile il servizio di sintesi vocale.
- **Sivodim::Libraries::Model::Mivoq::MivoqTTSService**
 - **Descrizione** - Classe concreta che implementa il contratto di TextToSpeechService, per rendere disponibile al sistema il servizio di sintesi vocale di MIVOQ s.r.l.;
 - **Utilizzo** - Viene utilizzata per rendere disponibile al sistema Android_G il servizio di sintesi vocale di MIVOQ s.r.l.;
 - **Classi ereditate**
 - * tts::TextToSpeechService.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton** - Relazione uscente, con la classe Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton. Utilizza la classe per gestire la sintesi vocale offerta da MIVOQ s.r.l..

- **Sivodim::Libraries::Model::Mivoq::AudioRequest**
 - **Descrizione** - Estensione della classe Request del pacchetto Volley;
 - **Utilizzo** - Viene utilizzata per gestire una risposta in formato audio da una richiesta $http_G$ generica;
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl** - Relazione entrante, dalla classe Sivodim::Libraries::Model::Mivoq::MivoqConnectionImpl. Viene utilizzato dalla classe per richiedere l'audio ai server MIVOQ s.r.l..

4.11 Sivodim::Libraries::Model::Voice

4.11.1 Informazioni sul package

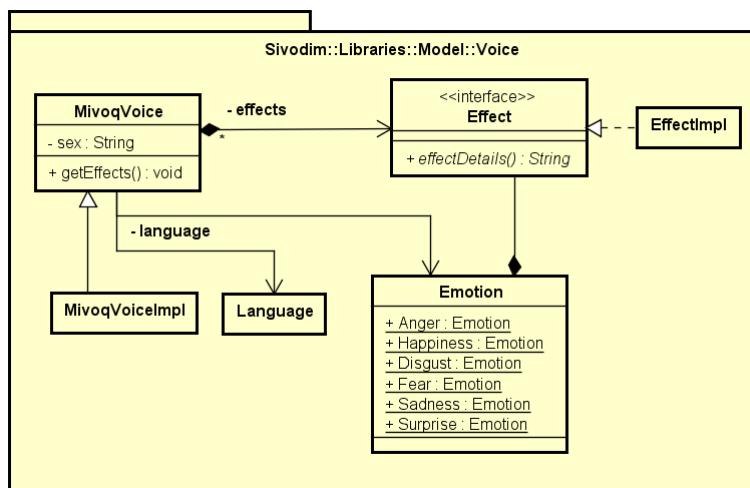


Figura 12: Componente Sivodim::Libraries::Model::Voice

4.11.2 Descrizione

Il $package_G$ racchiude le classi che definiscono le voci, che possono essere gestite dal motore di sintesi vocale di MIVOQ s.r.l.. In particolare vengono gestiti anche gli effetti e i sentimenti che possono esservi applicati.

4.11.3 Interazioni con altre componenti

- Sivodim::Libraries::Model::EngineManager.

4.11.4 Classi

• Sivodim::Libraries::Model::Voice::MivoqVoice

- **Descrizione** - Classe astratta che rappresenta una voce. È composta da una lingua ed un insieme di effetti ed emozioni;
- **Utilizzo** - Viene definito il contratto di rappresentazione di una voce all'interno dell' $engine_G$;
- **Relazioni con altre classi**

- * **Sivodim::Libraries::Model::Voice::Language** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::Voice::Language. La classe utilizza il riferimento per memorizzare la lingua con cui parlare;
 - * **Sivodim::Libraries::Model::Voice::Emotion** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::Voice::Emotion. La classe utilizza il riferimento per memorizzare l'emozione da applicare;
 - * **Sivodim::Libraries::Model::Voice::Effect** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::Voice::Effect. La classe utilizza il riferimento per memorizzare l'insieme degli effetti da applicare alla voce;
 - * **Sivodim::Libraries::Model::EngineManager::EngineImpl** - Relazione entrante, viene riferita dalla classe Sivodim::Libraries::Model::EngineManager::EngineImpl. La classe viene utilizzata per memorizzare le voci personalizzate dall'utente.
- **Sivodim::Libraries::Model::Voice::Language**
 - **Descrizione** - Classe che rappresenta i codici di lingua, paese e variante per la definizione del tipo di voce da utilizzare;
 - **Utilizzo** - Viene usata per rappresentare i codici di lingua, paese e variante per la definizione del tipo di voce;
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Voice::MivoqVoice** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Voice::MivoqVoice. La classe viene utilizzata per memorizzare la lingua della voce.
 - **Sivodim::Libraries::Model::Voice::Emotion**
 - **Descrizione** - Classe astratta che rappresenta un insieme di effetti che definiscono un'emozione;
 - **Utilizzo** - Viene utilizzata per rappresentare le emozioni essere date ad una voce;
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Voice::MivoqVoice** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Voice::Voice. La classe utilizza questa classe per applicare gli effetti che corrispondono ad una determinata emozione;
 - * **Sivodim::Libraries::Model::Voice::Effect** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::Voice::Effect. La classe utilizza il riferimento per memorizzare l'insieme degli effetti da applicare alla voce per simulare una determinata emozione.
 - **Sivodim::Libraries::Model::Voice::Effect**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare gli effetti di modifica della voce;
 - **Utilizzo** - Viene usata per definire il contratto della classe che rappresenta gli effetti di una voce;

- **Classi figlie**
 - * Sivodim::Libraries::Model::Voice::EffectImpl.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Voice::MivoqVoice** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Voice::Voice. La classe viene utilizzata per memorizzare gli effetti da applicare alla voce personalizzata;
 - * **Sivodim::Libraries::Model::Voice::Emotion** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Model::Voice::Emotion. La classe viene utilizzata per memorizzare gli effetti che compongono un'emozione.
- **Sivodim::Libraries::Model::Voice::EffectImpl**
 - **Descrizione** - Classe che implementa l'interfaccia Effect;
 - **Utilizzo** - Implementa il contratto della classe Effect che gestisce gli effetti da dare ad una voce;
 - **Classi ereditate**
 - * Sivodim::Libraries::Model::Voice::Effect.

4.12 Sivodim::Libraries::Presenter

4.12.1 Informazioni sul package

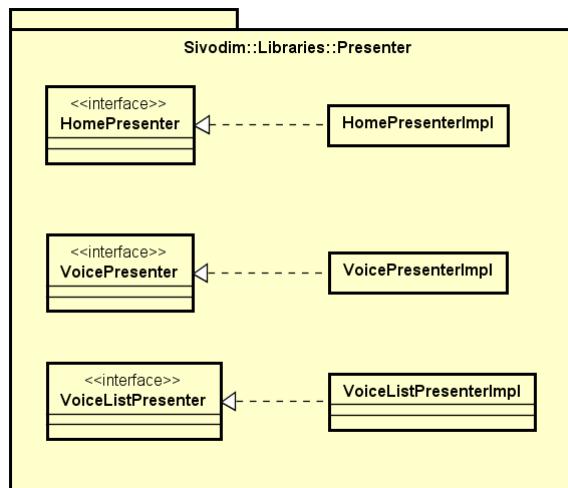


Figura 13: Componente Sivodim::Libraries::Presenter

4.12.2 Descrizione

Componente *man-in-the-middle* fra la View e il Model di Sivodim::Libraries. Si occupa di connettere le due componenti e gestire l'allocazione delle istanze della View.

4.12.3 Interazioni con altre componenti

- Sivodim::Libraries::Model;
- Sivodim::Libraries::View.

4.12.4 Classi

- **Sivodim::Libraries::Presenter::HomePresenter**

- **Descrizione** - Interfaccia che definisce il contratto attraverso cui implementare il *presenter* della pagina iniziale dell'applicazione di configurazione del motore di sintesi;
- **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alla HomeActivity contenuta nella componente Sivodim::Libraries::View;
- **Classi figlie**
 - * Sivodim::Libraries::Presenter::HomePresenterImpl.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::View::HomeActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::View::HomeActivityImpl. La classe viene utilizzata per memorizzare le istanze degli oggetti utilizzati nell'attività.

- **Sivodim::Libraries::Presenter::HomePresenterImpl**

- **Descrizione** - Classe che implementa l'interfaccia HomePresenter;
- **Utilizzo** - Viene utilizzata per implementare l'interfaccia HomePresenter;
- **Classi ereditate**
 - * Sivodim::Libraries::Presenter::HomePresenter.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::VoicePresenter** - Relazione uscente, ha un riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Presenter::VoicePresenter. La classe viene utilizzata per memorizzare un'istanza dell'oggetto che rappresenta una voce;
 - * **Sivodim::Libraries::View::HomeActivityInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::View::HomeActivityInterface. La classe viene utilizzata per permettere all'applicazione di tornare all'attività principale;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::EngineManager::Engine. Il riferimento permette di sfruttare il motore di sintesi vocale del testo.

- **Sivodim::Libraries::Presenter::VoicePresenter**

- **Descrizione** - Interfaccia che definisce il contratto attraverso cui implementare il *presenter* della pagina relativa alla gestione delle voci;
- **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alle NewVoiceActivity, VoiceListActivity, EditVoiceActivity presenti nella componente Sivodim::Libraries::View;
- **Classi figlie**
 - * Sivodim::Libraries::Presenter::VoicePresenterImpl.
- **Relazioni con altre classi**

- * **Sivodim::Drama::Presenter::HomePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Presenter::HomePresenterImpl. La classe permette di creare e memorizzare un'istanza della voce che si vuole creare o modificare;
 - * **Sivodim::Libraries::View::NewVoiceActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::View::NewVoiceActivityImpl. La classe permette di memorizzare un oggetto di tipo voce, che viene creato con le impostazioni dell'utente nell'attività;
 - * **Sivodim::Libraries::View::EditVoiceActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::View::EditVoiceActivityImpl. La classe permette di memorizzare un oggetto di tipo voce, che viene modificato con le impostazioni dell'utente nell'attività;
 - * **Sivodim::Libraries::View::VoiceListActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::View::VoiceListActivityImpl. La classe permette di memorizzare degli oggetti di tipo voce, per poterne visualizzare una lista per l'utente.
- **Sivodim::Libraries::Presenter::VoicePresenterImpl**
 - **Descrizione** - Classe che implementa l'interfaccia VoicePresenter;
 - **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle NewVoiceActivity, VoiceListActivity, EditVoiceActivity presenti nella componente Sivodim::Libraries::View;
 - **Classi ereditate**
 - * Sivodim::Libraries::Presenter::VoicePresenter.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::View::NewVoiceActivityInterface** - Relazione uscente, riferimento all'interfaccia Sivodim::Libraries::View::NewVoiceActivity. La classe permette di memorizzare un oggetto di tipo voce, che viene creato con le impostazioni dell'utente nell'attività;
 - * **Sivodim::Libraries::View::EditVoiceActivityInterface** - Relazione uscente, riferimento all'interfaccia Sivodim::Libraries::View::EditVoiceActivity. La classe permette di memorizzare un oggetto di tipo voce, che viene modificato con le impostazioni dell'utente nell'attività;
 - * **Sivodim::Libraries::View::VoiceListActivityInterface** - Relazione uscente, riferimento all'interfaccia Sivodim::Libraries::View::VoiceListActivity. La classe permette di memorizzare degli oggetti di tipo voce, per visualizzare la lista di voci disponibili per l'utente;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::EngineManager::Engine. La classe permette di utilizzare il servizio di sintesi vocale.
 - **Sivodim::Libraries::Presenter::VoiceListPresenter**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* della pagina relativa alla gestione della lista delle voci;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alla VoiceListActivity presenti nella componente Sivodim::Libraries::View;

- **Classi figlie**
 - * Sivodim::Libraries::Presenter::VoiceListPresenterImpl.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::View::VoiceListActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::View::VoiceListActivity. VoiceListActivity utilizza questo presenter per ricavare la lista delle voci da mostrare.
- **Sivodim::Libraries::Presenter::VoiceListPresenterImpl**
 - **Descrizione** - Classe che implementa l'interfaccia VoiceListPresenter;
 - **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle VoiceListActivity presenti nella componente Sivodim::Libraries::View;
 - **Classi ereditate**
 - * Sivodim::Libraries::Presenter::VoiceListPresenter.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::View::VoiceListActivityInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::VoiceListActivityInterface. Ha un riferimento alla VoiceListActivity-Interface che utilizza il presenter stesso, per poter ricavare la lista delle voci;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::EngineManager::Engine. VoicePresenterImpl utilizza la classe Engine per creare la sintesi sonora del testo contenuto nella battuta.

4.13 Sivodim::Libraries::View

4.13.1 Informazioni sul package

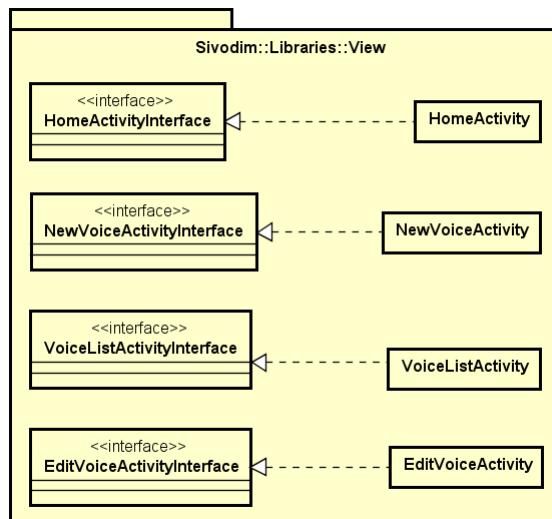


Figura 14: Componente Sivodim::Libraries::View

4.13.2 Descrizione

Il *package* racchiude l'interfaccia grafica della componente Sivodim::Libraries, che semplifica ad un utente la creazione di nuove voci per il sistema. Le classi contenute gestiscono le viste xml e catturano gli eventi generati.

4.13.3 Interazioni con altre componenti

- Sivodim::Libraries::Presenter.

4.13.4 Classi

- **Sivodim::Libraries::View::HomeActivityInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina principale dell'applicazione e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina principale dell'applicazione;
 - **Classi figlie**
 - * Sivodim::Libraries::View::HomeActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::HomePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Libraries::Presenter::HomePresenterImpl. La relazione permette di memorizzare le informazioni principali per visualizzare l'attività principale.
- **Sivodim::Libraries::View::HomeActivity**
 - **Descrizione** - Classe che implementa l'interfaccia HomeActivity;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina principale dell'applicazione;
 - **Classi ereditate**
 - * Sivodim::Libraries::View::HomeActivityInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::HomePresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Presenter::HomePresenter. La relazione permette di memorizzare gli oggetti principali;
 - * **Sivodim::Libraries::View::NewVoiceActivityInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::View::NewVoiceActivity. La relazione permette di creare un attività per la creazione di una nuova voce;
 - * **Sivodim::Libraries::View::VoiceListActivityInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::View::VoiceListActivity. La relazione permette di creare un attività per la visualizzazione della lista di voci disponibili.
- **Sivodim::Libraries::View::NewVoiceActivityInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view attraverso cui l'utente può creare una nuova voce personalizzata;
- **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina dedicata alla creazione di una nuova voce personalizzata;
- **Classi figlie**
 - * Sivodim::Libraries::View::NewVoiceActivity.
- **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::VoicePresenterImpl** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::Presenter::VoicePresenterImpl. La classe viene utilizzata per visualizzare la nuova voce che l'utente crea;
 - * **Sivodim::Libraries::View::HomeActivity** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::View::HomeActivity. La relazione permette di passare al menù di creazione della voce dall'attività principale.
- **Sivodim::Libraries::View::NewVoiceActivity**
 - **Descrizione** - Classe che implementa l'interfaccia NewVoiceActivity;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina dedicata alla creazione di nuove voci personalizzate;
 - **Classi ereditate**
 - * Sivodim::Libraries::View::NewVoiceActivityInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::VoicePresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Presenter::VoicePresenter. La relazione permette memorizzare un'istanza della voce che si vuole creare.
- **Sivodim::Libraries::View::EditVoiceActivityInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina dedicata alla modifica di una voce già esistente;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina dedicata alla modifica di una voce già esistente;
 - **Classi figlie**
 - * Sivodim::Libraries::View::EditVoiceActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Presenter::VoicePresenterImpl** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::Presenter::VoicePresenterImpl. La classe utilizza la relazione per memorizzare l'istanza dell'oggetto voce che si sta modificando.
- **Sivodim::Libraries::View::EditVoiceActivity**
 - **Descrizione** - Classe che implementa l'interfaccia EditVoiceActivity;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina dedicata alla modifica di una voce già esistente;
 - **Classi ereditate**

* Sivodim::Libraries::View::EditVoiceActivityInterface.

- **Relazioni con altre classi**

* **Sivodim::Libraries::Presenter::VoicePresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Presenter::VoicePresenter. La classe utilizza la relazione per applicare delle modifiche alla voce che si sta visualizzando.

• **Sivodim::Libraries::View::VoiceListActivityInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina contenente la lista completa delle voci offerte dall'applicazione;

- **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina contenente la lista completa delle voci offerte dall'applicazione;

- **Classi figlie**

* Sivodim::Libraries::View::VoiceListActivity.

- **Relazioni con altre classi**

* **Sivodim::Libraries::Presenter::VoicePresenterImpl** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::Presenter::VoicePresenterImpl. La relazione permette di memorizzare le istanze delle voci da memorizzare;

* **Sivodim::Libraries::View::HomeActivity** - Relazione entrante, è riferita dalla classe Sivodim::Libraries::View::HomeActivity. La relazione permette di visualizzare la lista delle voci disponibili dalla pagina principale.

• **Sivodim::Libraries::View::VoiceListActivity**

- **Descrizione** - Classe che implementa l'interfaccia VoiceListActivity;

- **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina contenente la lista completa delle voci offerte dall'applicazione;

- **Classi ereditate**

* Sivodim::Libraries::View::VoiceListActivityInterface.

- **Relazioni con altre classi**

* **Sivodim::Libraries::Presenter::VoicePresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Presenter::VoicePresenter. La relazione permette di memorizzare le voci da visualizzare nella lista di voci personali.

4.14 Sivodim::Drama::Model

4.14.1 Informazioni sul package

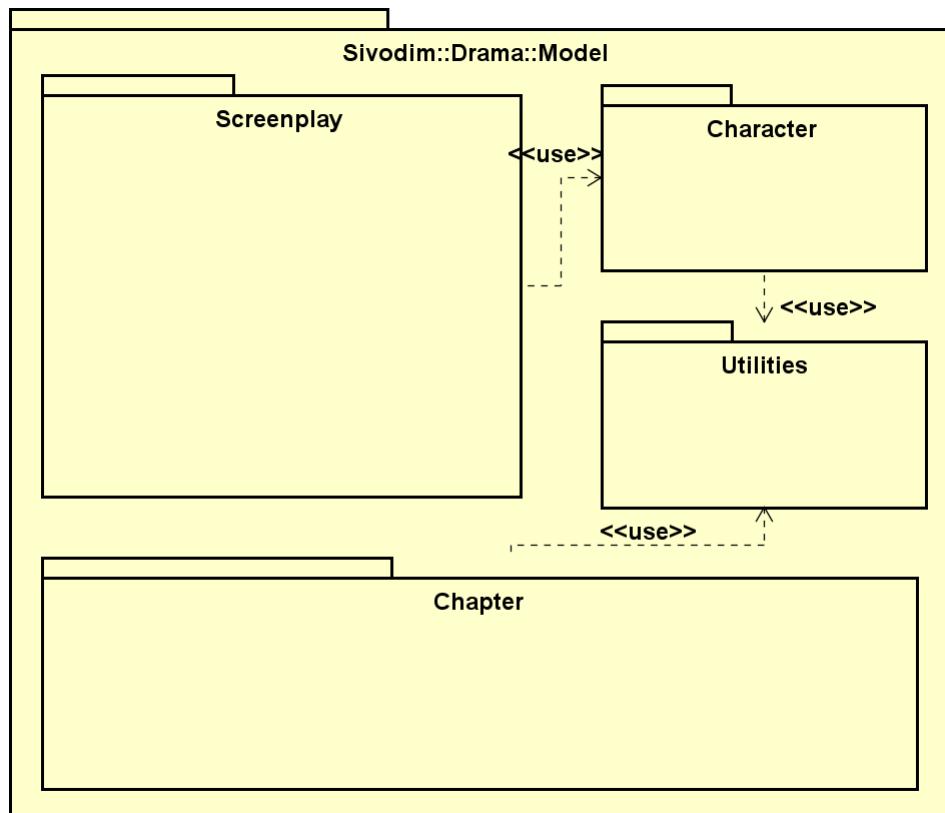


Figura 15: Componente Sivodim::Drama::Model

4.14.2 Descrizione

Il $package_G$ racchiude la $business\ logic_G$ della componente Sivodim::Drama. Si occupa di mantenere le informazioni necessarie all'applicazione degli sceneggiati e di definire le operazioni fra questi.

4.14.3 Package contenuti

- Sivodim::Drama::Model::Utilities;
- Sivodim::Drama::Model::Chapter;
- Sivodim::Drama::Model::Character;
- Sivodim::Drama::Model::Screenplay.

4.14.4 Interazioni con altre componenti

- FFMPEG;
- Sivodim::Libraries::Model;
- Sivodim::Drama::Presenter.

4.15 Sivodim::Drama::Presenter

4.15.1 Informazioni sul package

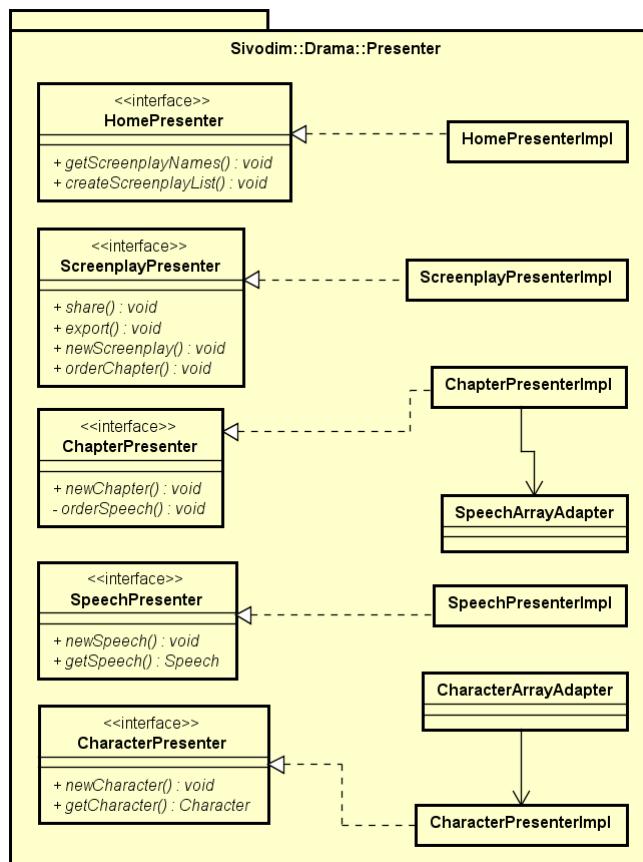


Figura 16: Componente Sivodim::Drama::Presenter

4.15.2 Descrizione

Il *package_G* gestisce la View di Sivodim::Drama e si relaziona con il Model della stessa.

4.15.3 Interazioni con altre componenti

- Sivodim::Drama::Model;
- Sivodim::Drama::View.

4.15.4 Classi

• Sivodim::Drama::Presenter::HomePresenter

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* della pagina iniziale dell'applicazione;
- **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alla *HomeActivity* presente nella componente Sivodim::Drama::View. Tale presenter dovrà gestire la vista principale dell'applicazione e fornirne metodi per la navigazione e il recupero di informazioni;

- **Classi figlie**
 - * Sivodim::Drama::Presenter::HomePresenterImpl.
- **Relazioni con altre classi**
 - * **Sivodim::Drama::View::HomeActivity** - Relazione entrante, la classe è riferita da Sivodim::Drama::View::HomeActivity. HomeActivity utilizza questo presenter per ricavare informazioni sugli sceneggiati presenti nel dispositivo e spostarsi ad altre activity.
- **Sivodim::Drama::Presenter::HomePresenterImpl**
 - **Descrizione** - Classe che implementa l'interfaccia HomePresenter;
 - **Utilizzo** - Viene usata per implementare l'interfaccia HomePresenter;
 - **Classi ereditate**
 - * Sivodim::Drama::Presenter::HomePresenter.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ScreenplayPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ScreenplayPresenter;
 - * **Sivodim::Drama::View::HomeInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::HomeActivity.
- **Sivodim::Drama::Presenter::ScreenplayPresenter**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* delle pagine relative alla gestione degli sceneggiati;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alle ListChapterActivity, ShareActivity, ExportActivity e NewScreenplayActivity presenti nella componente Sivodim::Drama::View;
 - **Classi figlie**
 - * Sivodim::Drama::Presenter::ScreenplayPresenterImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::HomePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::HomePresenterImpl. HomePresenterImpl istanzia lo ScreenplayPresenter nel momento in cui si passa ad una activity che questo deve gestire;
 - * **Sivodim::Drama::View::ListChapterActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::View::ListChapterActivity. ListChapterActivity utilizza questo presenter per ricavare informazioni sullo sceneggiato di cui deve mostrare i capitoli;
 - * **Sivodim::Drama::View::NewScreenplayActivity** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::View::NewScreenplayActivity. NewScreenplayActivity utilizza questo presenter per ricavare creare un nuovo sceneggiato sulla base dei dati raccolti dall'interazione con l'utente.
- **Sivodim::Drama::Presenter::ScreenplayPresenterImpl**
 - **Descrizione** - Classe che implementa l'interfaccia ScreenplayPresenter;

- **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle ListChapterActivity, ShareActivity, ExportActivity e NewScreenplayActivity presenti nella componente Sivodim::Drama::View;

- **Classi ereditate**

- * Sivodim::Drama::Presenter::ScreenplayPresenter.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Model::Screenplay::Screenplay** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Screenplay. Utilizza la classe Screenplay per reperire le informazioni dagli sceneggiati e passarle alle viste che le richiedono; inoltre può richiedere operazioni su essi;
 - * **Sivodim::Drama::View::HomeInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::HomeInterface. Ha un riferimento alla HomeInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::ListChapterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::ListChapterInterface. Ha un riferimento alla ListChapterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::NewScreenplayInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::NewScreenplayInterface. Ha un riferimento alla NewScreenplayInterface che utilizza il presenter stesso, per poterne modificare parametri.

- **Sivodim::Drama::Presenter::ChapterPresenter**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* che gestisce le attività relative ai capitoli di uno sceneggiato;

- **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alle ListSpeechesActivity, NewChapterActivity e EditChapterActivity presenti nel package Sivodim::Drama::View;

- **Classi figlie**

- * Sivodim::Drama::Presenter::ChapterPresenterImpl.

- **Relazioni con altre classi**

- * **Sivodim::Drama::View::NewChapterActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::NewChapterActivity. NewChapterActivity utilizza questo presenter per creare un nuovo capitolo sulla base delle informazioni ricavate dall'interazione con l'utente;
 - * **Sivodim::Drama::View::EditChapterActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::EditChapterActivity. EditChapterActivity utilizza questo presenter per ricavare informazioni sui capitoli dello sceneggiato e modificarli sulla base delle informazioni ricavate dall'interazione con l'utente;
 - * **Sivodim::Drama::View::ListSpeechesActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::ListSpeechesActivity. ListSpeechesActivity utilizza questo presenter per ricavare informazioni sulle battute del capitolo che deve mostrare.

- **Sivodim::Drama::Presenter::ChapterPresenterImpl**

- **Descrizione** - Classe che implementa l'interfaccia ChapterPresenter;
- **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle ListSpeechesActivity, NewChapterActivity e EditChapterActivity presenti nel package_G Sivodim::Drama::View;
- **Classi ereditate**
 - * Sivodim::Drama::Presenter::ChapterPresenter.
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::Chapter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Model::Chapter. Ha un riferimento ad un oggetto di tipo Chapter per poterlo gestire in base alle interazioni dell'utente che arrivano dalla activity correlata al presenter;
 - * **Sivodim::Drama::View::ListSpeechesInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::ListSpeechesInterface. Ha un riferimento alla ListSpeechesInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::NewChapterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::NewChapterInterface. Ha un riferimento alla NewChapterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::EditChapterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::EditChapterInterface. Ha un riferimento alla EditChapterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::Presenter::SpeechArrayAdapter** - Relazione uscente, istanzia oggetti di tipo Sivodim::Drama::Presenter::SpeechArrayAdapter.
- **Sivodim::Drama::Presenter::SpeechPresenter**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* delle pagine relative alla gestione delle battuta;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alle NewSpeechActivity ed EditSpeechActivity presenti nella componente Sivodim::Drama::View;
 - **Classi figlie**
 - * Sivodim::Drama::Presenter::SpeechPresenterImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::View::NewSpeechActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::NewSpeechActivity. NewSpeechActivity utilizza questo presenter per creare una nuova battuta sulla base delle informazioni ricavate dall'interazione con l'utente;
 - * **Sivodim::Drama::View::EditSpeechActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::EditSpeechActivity. EditSpeechActivity utilizza questo presenter modificare una battuta sulla base delle informazioni ricavate dall'interazione con l'utente.
- **Sivodim::Drama::Presenter::SpeechPresenterImpl**

- **Descrizione** - Classe che implementa l'interfaccia SpeechPresenter;
- **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle NewSpeechActivity ed EditSpeechActivity presenti nella componente Sivodim::Drama::View;
- **Classi ereditate**
 - * Sivodim::Drama::Presenter::SpeechPresenter.
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::Speech** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Model::Chapter::Speech. Ha un riferimento ad una battuta per poterla gestire e ricavarne le informazioni richieste dalle activities correlate;
 - * **Sivodim::Drama::View::NewSpeechInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::NewSpeechInterface. Ha un riferimento alla NewSpeechInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::EditSpeechInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::EditSpeechInterface. Ha un riferimento alla EditSpeechInterface che utilizza il presenter stesso, per poterne modificare parametri.
- **Sivodim::Drama::Presenter::CharacterPresenter**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare il *presenter* delle pagine relative alla gestione dei personaggi;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione del *presenter* associato alle NewCharacterActivity, ListCharacterActivity ed EditCharacterActivity presenti nella componente Sivodim::Drama::View;
 - **Classi figlie**
 - * Sivodim::Drama::Presenter::CharacterPresenterImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::View::NewCharacterActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::NewCharacterActivity. NewCharacterActivity utilizza questo presenter per creare nuovi personaggi sulla base delle informazioni raccolte dall'interazione con l'utente;
 - * **Sivodim::Drama::View::ListCharacterActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::ListCharacterActivity. ListCharacterActivity utilizza questo presenter per ricavare informazioni sugli personaggi che deve mostrare;
 - * **Sivodim::Drama::View::EditCharacterActivity** - Relazione entrante, è riferita dalla classe Sivodim::Drama::View::EditCharacterActivity. EditCharacterActivity utilizza questo presenter per ricavare informazioni sugli personaggi e modificarli in base alle interazioni con l'utente.
- **Sivodim::Drama::Presenter::CharacterPresenterImpl**
 - **Descrizione** - Classe che implementa l'interfaccia CharacterPresenter;

- **Utilizzo** - Viene usata per implementare il contratto del *presenter* associato alle NewCharacterActivity, EditCharacterActivity e ListCharacterActivity presenti nella componente Sivodim::Drama::View;

- **Classi ereditate**

- * Sivodim::Drama::Presenter::CharacterPresenter.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Model::Character::Character** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Model::Character::Character. Ha un riferimento al personaggio che deve gestire;
 - * **Sivodim::Drama::View::NewCharacterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::NewCharacterInterface. Ha un riferimento alla NewCharacterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::ListCharacterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::ListCharacterInterface. Ha un riferimento alla ListCharacterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::View::EditCharacterInterface** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::View::EditCharacterInterface. Ha un riferimento alla EditCharacterInterface che utilizza il presenter stesso, per poterne modificare parametri;
 - * **Sivodim::Drama::Presenter::CharacterAdapter** - Relazione uscente, istanzia oggetti di tipo Sivodim::Drama::Presenter::CharacterAdapter.

- **Sivodim::Drama::Presenter::FfmpegConnector**

- **Descrizione** - Classe astratta che si relaziona con FFMPEG, passandone i comandi e ricevendo i messaggi generati dall'esecuzione;
- **Utilizzo** - Viene utilizzata per comunicare con FFMPEG per passare comandi;
- **Classi figlie**
 - * Sivodim::Drama::Model::Screenplay::Mp3Converter;
 - * Sivodim::Drama::Model::Screenplay::AudioConcatenator;
 - * Sivodim::Drama::Model::Screenplay::AudioMixer.

- **Sivodim::Drama::Presenter::CharacterAdapter**

- **Descrizione** - Ridefinizione di ArrayAdapter, che utilizza un layout personale con più widget;
- **Utilizzo** - Viene utilizzato per mostrare nella stessa linea il nome di un personaggio e il suo avatar;
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Character::Character** - Relazione uscente, ha un riferimento ad una classe che implementa l'interfaccia Character, da cui ricava le informazioni da inserire nell'ArrayAdapter;
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, Sivodim::Drama::Model::Presenter::CharacterPresenterImpl istanzia oggetti di questa classe.

- **Sivodim::Drama::Presenter::SpeechArrayAdapter**

- **Descrizione** - Ridefinizione di ArrayAdapter, che utilizza un layout personale con più widget;
- **Utilizzo** - Viene utilizzato per mostrare nella stessa linea il testo di una battuta, il nome del personaggio associato e il suo avatar;
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::Speech** - Relazione uscente, ha un riferimento ad una classe che implementa l'interfaccia Speech, da cui ricava le informazioni da inserire nell'ArrayAdapter;
 - * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, Sivodim::Drama::Model::Presenter::ChapterPresenterImpl istanzia oggetti di questa classe.

4.16 Sivodim::Drama::View

4.16.1 Informazioni sul package

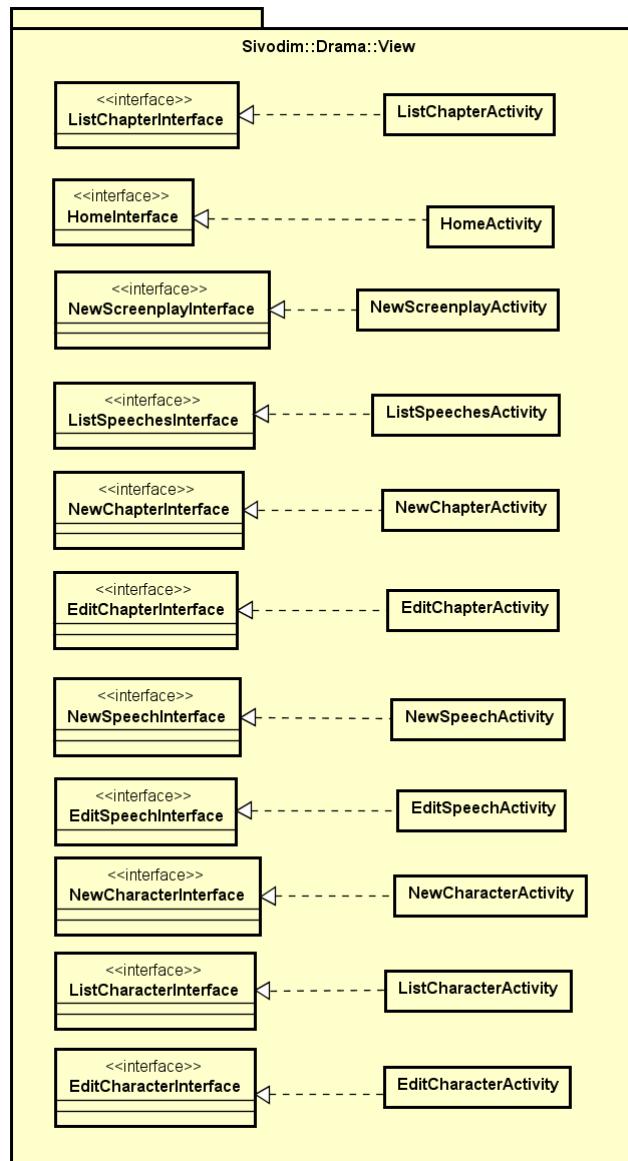


Figura 17: Componente Sivodim::Drama::View

4.16.2 Descrizione

Package_G per la componente View dell'architettura MVP.

4.16.3 Interazioni con altre componenti

- Sivodim::Drama::Presenter.

4.16.4 Classi

- **Sivodim::Drama::View::HomeInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina principale dell'applicazione e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina principale dell'applicazione;
 - **Classi figlie**
 - * Sivodim::Drama::View::HomeActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::HomePresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::HomePresenterImpl. HomePresenterImpl utilizza questa interfaccia per modificare la view;
 - * **Sivodim::Drama::Presenter::ScreenplayPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ScreenplayPresenterImpl. ScreenplayPresenterImpl utilizza questa interfaccia per la navigazione fra activity.
- **Sivodim::Drama::View::HomeActivity**
 - **Descrizione** - Classe che implementa l'interfaccia HomeActivity;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina principale dell'applicazione;
 - **Classi ereditate**
 - * Sivodim::Drama::View::HomeInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::HomePresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::HomePresenter. Ha un riferimento alla classe HomePresenter per reperire le informazioni da mostrare all'utente e richiamare le funzionalità richieste dall'interazione.
- **Sivodim::Drama::View::ListChapterInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina di visualizzazione dei capitoli e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina di visualizzazione dei capitoli dello sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::View::ListChapterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ScreenplayPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ScreenplayPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::ListChapterActivity**
 - **Descrizione** - Classe che implementa l'interfaccia ListChapterInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina della lista dei capitoli dello sceneggiato;

- **Classi ereditate**

- * Sivodim::Drama::View::ListChapterInterface.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Presenter::ScreenplayPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ScreenplayPresenter. Interagisce con i dati degli oggetti di tipo Chapter per mostrare una lista di capitoli relativi a un dato sceneggiato.

- **Sivodim::Drama::View::NewScreenplayInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la creazione di un nuovo sceneggiato e catturarne gli eventi;

- **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la creazione di un nuovo sceneggiato;

- **Classi figlie**

- * Sivodim::Drama::View::NewScreenplayActivity.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Presenter::ScreenplayPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ScreenplayPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.

- **Sivodim::Drama::View::NewScreenplayActivity**

- **Descrizione** - Classe che implementa l'interfaccia NewScreenplayInterface;

- **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la creazione di un nuovo sceneggiato;

- **Classi ereditate**

- * Sivodim::Drama::View::NewScreenplayInterface.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Presenter::ScreenplayPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ScreenplayPresenter. Istanzia e interagisce con i dati di uno sceneggiato di tipo Screenplay riferito da ScreenplayPresenter, al fine di creare un nuovo sceneggiato.

- **Sivodim::Drama::View::ListSpeechesInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina che mostra l'elenco delle battute del capitolo e catturarne gli eventi;

- **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina che mostra l'elenco delle battute del capitolo;

- **Classi figlie**

- * Sivodim::Drama::View::ListSpeechesActivity.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ChapterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare sotto forma di lista.
- **Sivodim::Drama::View::NewChapterInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la creazione di un nuovo capitolo e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la creazione di un nuovo capitolo;
 - **Classi figlie**
 - * Sivodim::Drama::View::NewChapterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ChapterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::NewChapterActivity**
 - **Descrizione** - Classe che implementa l'interfaccia NewChapterInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la creazione di un nuovo capitolo;
 - **Classi ereditate**
 - * Sivodim::Drama::View::NewChapterInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ChapterPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ChapterPresenter. Istanzia e interagisce con i dati di un capitolo di tipo Chapter riferito da ChapterPresenter, al fine di creare un nuovo capitolo.
- **Sivodim::Drama::View::EditChapterInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la modifica del capitolo di uno sceneggiato e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la modifica del capitolo di uno sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::View::EditChapterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ChapterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::EditChapterActivity**

- **Descrizione** - Classe che implementa l'interfaccia EditChapterInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la modifica di un capitolo di uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::View::EditChapterInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ChapterPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ChapterPresenter. Interagisce con i dati del capitolo di tipo Chapter riferito da ChapterPresenter, al fine di modificare i valori del capitolo dato.
- **Sivodim::Drama::View::NewSpeechInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la creazione di una nuova battuta e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la creazione di una nuova battuta;
 - **Classi figlie**
 - * Sivodim::Drama::View::NewSpeechActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::SpeechPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::SpeechPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::NewSpeechActivity**
 - **Descrizione** - Classe che implementa l'interfaccia NewSpeechInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la creazione di una nuova battuta;
 - **Classi ereditate**
 - * Sivodim::Drama::View::NewSpeechInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::SpeechPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::SpeechPresenter. Istanzia e interagisce con i dati di una battuta di tipo Speech riferita da SpeechPresenter.
- **Sivodim::Drama::View::EditSpeechInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la modifica di una battuta dello sceneggiato e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la modifica di una battuta dello sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::View::EditSpeechActivity.

- **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::SpeechPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::SpeechPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::EditSpeechActivity**
 - **Descrizione** - Classe che implementa l'interfaccia EditSpeechInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la modifica della battuta di uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::View::EditSpeechInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::SpeechPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::SpeechPresenter. Interagisce con i dati di una battuta di tipo Speech riferita da SpeechPresenter, al fine di modificare i valori della battuta data.
- **Sivodim::Drama::View::NewCharacterInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina che permette la creazione di un nuovo personaggio e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina che permette la creazione di un nuovo personaggio;
 - **Classi figlie**
 - * Sivodim::Drama::View::NewCharacterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::CharacterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::NewCharacterActivity**
 - **Descrizione** - Classe che implementa l'interfaccia NewCharacterInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la creazione di un nuovo personaggio;
 - **Classi ereditate**
 - * Sivodim::Drama::View::NewCharacterInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::CharacterPresenter. Istanzia e interagisce con i dati del personaggio di tipo Character riferiti da CharacterPresenter.
- **Sivodim::Drama::View::ListCharacterInterface**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina che mostra l'elenco dei personaggi dello sceneggiato e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina che mostra l'elenco dei personaggi dello sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::View::ListCharacterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::CharacterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::ListCharacterActivity**
 - **Descrizione** - Classe che implementa l'interfaccia ListCharacterInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina che mostra l'elenco dei personaggi di uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::View::ListCharacterInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::CharacterPresenter. Interagisce con i dati degli oggetti di tipo Character per mostrare una lista di personaggi relativi a un dato sceneggiato.
- **Sivodim::Drama::View::EditCharacterInterface**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la view della pagina per la modifica di un personaggio dello sceneggiato e catturarne gli eventi;
 - **Utilizzo** - Viene usata per definire il contratto di interazione con l'utente nella pagina per la modifica di un personaggio dello sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::View::EditCharacterActivity.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::CharacterPresenterImpl. La classe è riferita dal Presenter che ne gestisce i dati da visualizzare.
- **Sivodim::Drama::View::EditCharacterActivity**
 - **Descrizione** - Classe che implementa l'interfaccia EditCharacterInterface;
 - **Utilizzo** - Implementa il contratto dell'interfaccia EditCharacterInterface, necessaria per permettere all'utente di interagire con la schermata di modifica di un dato personaggio;
 - **Classi ereditate**
 - * Sivodim::Drama::View::EditCharacterInterface.

- **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::CharacterPresenter** - Relazione uscente, riferimento alla classe che implementa l'interfaccia Sivodim::Drama::Presenter::CharacterPresenter. Interagisce con i dati del personaggio di tipo Character riferiti da CharacterPresenter.
- **Sivodim::Drama::View::ListSpeechesActivity**
 - **Descrizione** - Classe che implementa l'interfaccia ListSpeechesInterface;
 - **Utilizzo** - Implementa il contratto di interazione con l'utente nella pagina per la visualizzazione delle battute di uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::View::ListSpeechesInterface.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ChapterPresenter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Presenter::ChapterPresenter. Interagisce con i dati di un capitolo di tipo Chapter riferito da ChapterPresenter, per visualizzare una lista dei capitoli contenuti in un dato sceneggiato.

4.17 Sivodim::Drama::Model::Utilities

4.17.1 Informazioni sul package

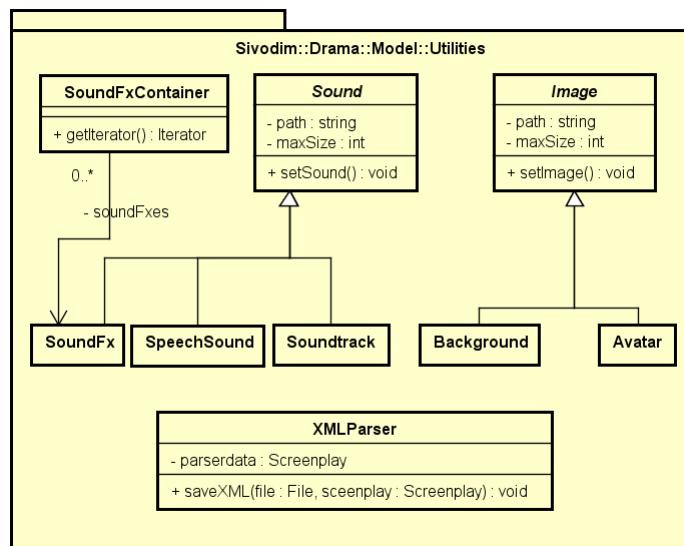


Figura 18: Componente Sivodim::Drama::Model::Utilities

4.17.2 Descrizione

Package che contiene classi di utilità alle altre componenti di Sivodim::Drama::Model, come la gestione delle immagini e dei suoni.

4.17.3 Interazioni con altre componenti

- Sivodim::Drama::Model::Chapter;
- Sivodim::Drama::Model::Character.

4.17.4 Classi

- **Sivodim::Drama::Model::Utilities::Sound**
 - **Descrizione** - Classe astratta che rappresenta un file audio generico. Viene concretizzata nelle classi che rappresentano un effetto sonoro, l'audio di una battuta e una colonna sonora;
 - **Utilizzo** - Viene usata per rappresentare un file audio di tipo generico;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Utilities::SoundFx;
 - * Sivodim::Drama::Model::Utilities::SpeechSound;
 - * Sivodim::Drama::Model::Utilities::Soundtrack.
- **Sivodim::Drama::Model::Utilities::SoundFx**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo SoundFx, ossia un effetto sonoro;
 - **Utilizzo** - Viene usata per concretizzare la classe Sound in un oggetto di tipo SoundFx;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Utilities::Sound.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::SpeechImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Chapter::SpeechImpl.
- **Sivodim::Drama::Model::Utilities::SpeechSound**
 - **Descrizione** - Classe che implementa un oggetto di tipo SpeechSound, ossia l'audio sintetizzato dal testo di una battuta;
 - **Utilizzo** - Viene usata per concretizzare la classe Sound in un oggetto nel tipo SpeechSound;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Utilities::Sound.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::SpeechImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Chapter::SpeechImpl. Ogni battuta viene sintetizzata in audio, e tale file audio è gestito dalla classe SpeechSound.
- **Sivodim::Drama::Model::Utilities::Soundtrack**
 - **Descrizione** - Classe che implementa un oggetto di tipo Soundtrack, ossia la colonna sonora relativa ad un capitolo;

- **Utilizzo** - Viene usata per concretizzare la classe Sound in un oggetto del tipo Soundtrack;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Utilities::Sound.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::Chapter** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Chapter::Chapter. Ogni Chapter contiene (opzionalmente) una traccia audio di sottofondo, rappresentata dalla classe Soundtrack.
- **Sivodim::Drama::Model::Utilities::Image**
 - **Descrizione** - Classe astratta che rappresenta un file di tipo immagine. Viene concretizzata nelle classi che rappresentano un'immagine di sfondo e un *avatar_G* da assegnare a un personaggio;
 - **Utilizzo** - Viene usata per rappresentare un file di tipo immagine attraverso un oggetto che contiene il percorso del file nella memoria del dispositivo e la sua grandezza massima;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Utilities::Background;
 - * Sivodim::Drama::Model::Utilities::Avatar.
- **Sivodim::Drama::Model::Utilities::Background**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo Background, ossia un'immagine di sfondo relativa a un capitolo;
 - **Utilizzo** - Viene usata per concretizzare la classe Image in un oggetto del tipo Background;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Utilities::Image.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::Chapter** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Chapter::Chapter. Ogni capitolo contiene (opzionalmente) un'immagine di sfondo, rappresentata dalla classe Background.
- **Sivodim::Drama::Model::Utilities::Avatar**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo Avatar;
 - **Utilizzo** - Viene usata per concretizzare la classe Image in un oggetto del tipo Avatar;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Utilities::Image.
 - **Relazioni con altre classi**

- * **Sivodim::Drama::Model::Character::CharacterImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Character::CharacterImpl. Ogni personaggio contiene un'immagine come avatar, rappresentata dalla classe Avatar.
- **Sivodim::Drama::Model::Utilities::MivoqTTSService**
 - **Descrizione** - Classe concreta che implementa il contratto di TextToSpeech-Service, per rendere disponibile al sistema il servizio di sintesi vocale di MIVOQ s.r.l.;
 - **Utilizzo** - Viene utilizzata per rendere disponibile al sistema Android_G il servizio di sintesi vocale di MIVOQ s.r.l.;
 - **Classi ereditate**
 - * tts::TextToSpeechService.
 - **Relazioni con altre classi**
 - * **Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton** - Relazione uscente, con la classe Sivodim::Libraries::Model::Mivoq::MivoqTTSSingleton. Utilizza la classe per gestire la sintesi vocale offerta da MIVOQ s.r.l..
- **Sivodim::Drama::Model::Utilities::XMLParser**
 - **Descrizione** - Classe di utilità che si occupa di creare file XMLG e farne parsingG;
 - **Utilizzo** - Viene utilizzata per salvare in memoria oggetti Screenplay sotto forma di files XML e riconvertirli poi in oggetti nella fase di load;
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::Screenplay** - Riferimento uscente, la classe istanzia oggetti di tipo Sivodim::Drama::Model::Screenplay::Screeenplay;
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Riferimento entrante, è riferito dalla classe Sivodim::Drama::Model::Screenplay::ScreeenplayImpl. La classe ScreenplayImpl lo utilizza per effettuare il load e il salvataggio di sceneggiati.

4.18 Sivodim::Drama::Model::Chapter

4.18.1 Informazioni sul package

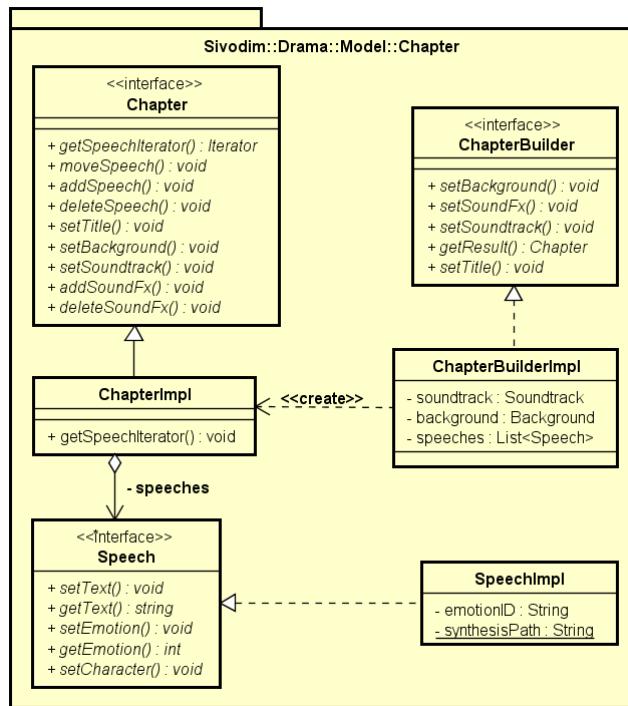


Figura 19: Componente Sivodim::Drama::Model::Chapter

4.18.2 Descrizione

Package_G che gestisce i capitoli di uno sceneggiato e le battute in essi contenute.

4.18.3 Interazioni con altre componenti

- Sivodim::Libraries::Model;
 - Sivodim::Drama::Model::Utilities;
 - Sivodim::Drama::Model::Character;
 - Sivodim::Drama::Model::Screenplay.

4.18.4 Classi

- Sivodim::Drama::Model::Chapter::Chapter

- **Descrizione** - Interfaccia che definisce il contratto di un oggetto di tipo capitolo.
Ha un riferimento all'interfaccia Speech che definisce il contratto di un oggetto di tipo battuta;
 - **Utilizzo** - Viene usata per rappresentare un oggetto di tipo capitolo;
 - **Classi figlie**

* Sivodim::Drama::Model::Chapter::ChapterImpl.

- Relazioni con altre classi

- * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Screenplay::ScreenplayImpl. Ogni sceneggiato contiene una lista di capitoli di tipo Chapter;
- * **Sivodim::Drama::Model::Utilities::Soundtrack** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Utilities::Soundtrack. Ogni capitolo contiene (opzionalmente) un oggetto Soundtrack che rappresenta un file audio come sottofondo musicale;
- * **Sivodim::Drama::Model::Utilities::Background** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Utilities::Background. Ogni capitolo contiene (opzionalmente) un oggetto Background che rappresenta un'immagine di sfondo;
- * **Sivodim::Drama::Model::Chapter::Speech** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Chapter::Speech. Ogni capitolo contiene una lista di oggetti Speech che rappresentano una battuta;
- * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Presenter::ChapterPresenterImpl. ChapterPresenterImpl istanzia un oggetto Chapter per garantire una corretta gestione del capitolo.

- **Sivodim::Drama::Model::Chapter::ChapterImpl**

- **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo capitolo;

- **Utilizzo** - Viene usata per concretizzare la classe Chapter in un oggetto del tipo ChapterImpl;

- **Classi ereditate**

- * Sivodim::Drama::Model::Chapter::Chapter.

- **Relazioni con altre classi**

- * **Sivodim::Drama::Model::Chapter::ChapterBuilderImpl** - Relazione entrante, la classe è istanziata dal builder Sivodim::Drama::Model::Chapter::ChapterBuilderImpl.

- **Sivodim::Drama::Model::Chapter::ChapterBuilder**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la creazione di un oggetto di tipo capitolo;

- **Utilizzo** - Viene usata per definire il contratto di implementazione del Builder necessario per la creazione di un oggetto di tipo capitolo;

- **Classi figlie**

- * Sivodim::Drama::Model::Chapter::ChapterBuilderImpl.

- **Sivodim::Drama::Model::Chapter::ChapterBuilderImpl**

- **Descrizione** - Classe che implementa un oggetto di tipo ChapterBuilderImpl necessario per la creazione di un oggetto ChapterImpl;

- **Utilizzo** - Viene usata per implementare il Builder necessario per la creazione di un oggetto di tipo capitolo;

- **Classi ereditate**
 - * Sivodim::Drama::Model::Chapter::ChapterBuilder.
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::ChapterImpl** - Relazione uscente, istanza un oggetto di tipo Sivodim::Drama::Model::Chapter::ChapterImpl.
- **Sivodim::Drama::Model::Chapter::Speech**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare un oggetto di tipo battuta;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione di un oggetto di tipo battuta;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Chapter::SpeechImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::SpeechPresenterImpl** - Relazione entrante, l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::SpeechPresenterImpl. Lo SpeechPresenter ne mantiene un riferimento per poter gestire la battuta in modo sicuro in relazione agli eventi delle relative Activities.
- **Sivodim::Drama::Model::Chapter::SpeechImpl**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo battuta. Un oggetto battuta contiene un riferimento ad un oggetto di tipo personaggio;
 - **Utilizzo** - Viene usata per implementare l'interfaccia del tipo battuta;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Chapter::Speech.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Character::Character** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Model::Character::Character. Ogni battuta ha un riferimento al personaggio associato che la pronuncia;
 - * **Sivodim::Drama::Model::Utilities::SoundFx** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Utilities::SoundFx. Alla fine di ogni battuta è possibile inserire un effetto sonoro rappresentato dalla classe SoundFx;
 - * **Sivodim::Drama::Model::Utilities::SpeechSound** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Utilities::SpeechSound. Ogni battuta dovrà essere sintetizzata in un file audio, che è gestito dalla classe SpeechSound. Un riferimento a tale file audio viene mantenuto dalle battute stesse;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Libraries::Model::EngineManager::Engine. SpeechImpl utilizza la classe Engine per creare la sintesi sonora del testo contenuto nella battuta.
- **Sivodim::Drama::Model::Chapter::SpeechBuilder**

- **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la creazione di un oggetto di tipo Speech;
- **Utilizzo** - Viene usata per definire il contratto di implementazione del extitBuilder necessario per la creazione di un oggetto di tipo Speech.
- **Sivodim::Drama::Model::Chapter::SpeechBuilderImpl**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto extitBuilder;
 - **Utilizzo** - Viene usata per implementare l'interfaccia di tipo SpeechBuilder. La creazione di un oggetto di tipo battuta è infatti compito della classe SpeechBuilderImpl strutturata secondo il extitdesign pattern Builder;
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::SpeechImpl** - Relazione uscente, crea un oggetto di tipo SpeechImpl.

4.19 Sivodim::Drama::Model::Character

4.19.1 Informazioni sul package

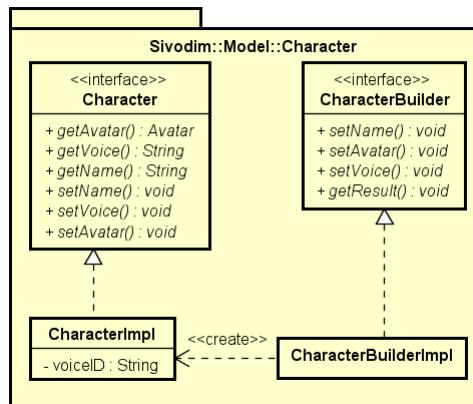


Figura 20: Componente Sivodim::Drama::Model::Character

4.19.2 Descrizione

Package_G parte del Model di Sivodim::Drama, relativo alla gestione dei personaggi.

4.19.3 Interazioni con altre componenti

- Sivodim::Libraries::Model;
- Sivodim::Drama::Model::Utilities;
- Sivodim::Drama::Model::Chapter;
- Sivodim::Drama::Model::Screenplay.

4.19.4 Classi

- **Sivodim::Drama::Model::Character::Character**
 - **Descrizione** - Interfaccia che definisce il contratto attraverso cui implementare un oggetto di tipo personaggio;
 - **Utilizzo** - Viene usata per definire il contratto da rispettare per l'implementazione di un oggetto di tipo personaggio;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Character::CharacterImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Chapter::SpeechImpl** - Relazione entrante, l'interfaccia è riferita da una classe di tipo Sivodim::Drama::Model::Chapter::SpeechImpl. Ogni battuta ha un riferimento ad un personaggio di tipo Character;
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, l'interfaccia è riferita da una classe di tipo Sivodim::Drama::Presenter::CharacterPresenterImpl. Il CharacterPresenter ne mantiene un riferimento per poter gestire il personaggio in modo sicuro in relazione agli eventi delle relative Activities.
- **Sivodim::Drama::Model::Character::CharacterImpl**
 - **Descrizione** - Classe che implementa la rappresentazione di un oggetto di tipo personaggio. La classe ha un riferimento a un oggetto di tipo Avatar;
 - **Utilizzo** - Viene usata per implementare l'interfaccia del tipo personaggio;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Character::Character.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Character::CharacterBuilderImpl** - Relazione entrante, la classe è istanziata da Sivodim::Drama::Model::Character::CharacterBuilderImpl;
 - * **Sivodim::Drama::Model::Utilities::Avatar** - Relazione uscente, riferimento ad un oggetto della classe Sivodim::Drama::Model::Utilities::Avatar che si occupa di mantenere un'immagine come *avatar_G* del personaggio;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento alla classe Sivodim::Libraries::Model::EngineManager::Engine per la gestione della voce del personaggio.
- **Sivodim::Drama::Model::Character::CharacterBuilder**
 - **Descrizione** - Interfaccia che rappresenta il contratto attraverso cui implementare la creazione di un oggetto di tipo personaggio;
 - **Utilizzo** - Viene usata per definire il contratto di implementazione del *Builder* necessario per la creazione di un oggetto di tipo personaggio;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Character::CharacterBuilderImpl.
- **Sivodim::Drama::Model::Character::CharacterBuilderImpl**

- **Descrizione** - Classe che implementa la rappresentazione di un oggetto *Builder*;
- **Utilizzo** - Viene usata per implementare l'interfaccia di tipo CharacterBuilder. La creazione di un oggetto di tipo personaggio è infatti compito della classe CharacterBuilderImpl strutturata secondo il *design pattern Builder*;
- **Classi ereditate**
 - * Sivodim::Drama::Model::Character::CharacterBuilder.
- **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Character::CharacterImpl** - Relazione uscente, crea un oggetto di tipo CharacterImpl.
- **Sivodim::Drama::Model::Character::CharacterContainer**
 - **Descrizione** - Classe che implementa il contenitore dei personaggi relativi a uno sceneggiato di tipo Screenplay. Contiene una lista di personaggi di tipo Character;
 - **Utilizzo** - Necessaria per importare i personaggi da uno sceneggiato a un altro;
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, riferita dalla classe Sivodim::Drama::Model::Screenplay::ScreenplayImpl. La classe viene istanziata per tenere traccia della lista dei personaggi relativi a un dato sceneggiato di tipo Screenplay;
 - * **Sivodim::Drama::Model::Character::Character** - Relazione uscente, riferimento all'interfaccia Sivodim::Drama::Model::Character::Character. La classe istanzia una lista di oggetti personaggio di tipo Character;
 - * **Sivodim::Drama::Presenter::ChapterPresenterImpl** - Relazione entrante, riferita dalla classe Sivodim::Drama::Presenter::ChapterPresenterImpl. La classe viene riferita per permettere lo scambio di personaggi tra lo sceneggiato di tipo Screenplay e se singole battute di tipo Speech;
 - * **Sivodim::Drama::Presenter::SpeechPresenterImpl** - Relazione entrante, riferita dalla classe Sivodim::Drama::Presenter::SpeechPresenterImpl. La classe viene riferita per permettere di selezionare un personaggio dalla lista e associarlo alla battuta di tipo Speech;
 - * **Sivodim::Drama::Presenter::CharacterPresenterImpl** - Relazione entrante, riferita dalla classe Sivodim::Drama::Presenter::CharacterPresenterImpl. La classe viene riferita per permettere la gestione dei personaggi che contiene.

4.20 Sivodim::Drama::Model::Screenplay

4.20.1 Informazioni sul package

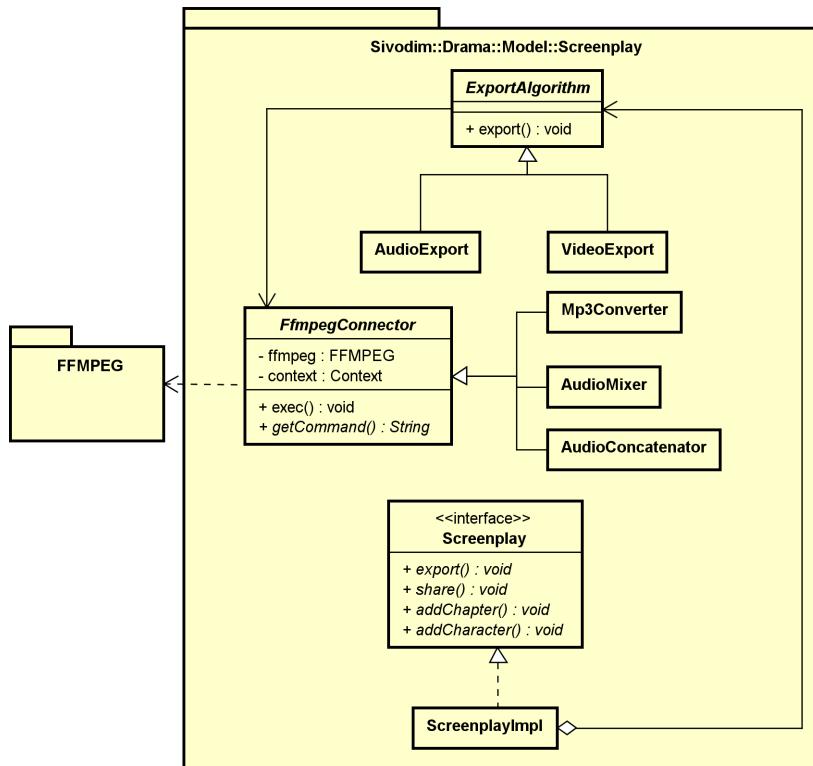


Figura 21: Componente Sivodim::Drama::Model::Screenplay

4.20.2 Descrizione

Package_G parte del Model di Sivodim::Drama, relativo alla gestione della parte logica degli sceneggiati.

4.20.3 Interazioni con altre componenti

- FFMPEG;
- Sivodim::Libraries::Model;
- Sivodim::Drama::Presenter;
- Sivodim::Drama::Model::Chapter;
- Sivodim::Drama::Model::Character.

4.20.4 Classi

- **Sivodim::Drama::Model::Screenplay::Screenplay**
 - **Descrizione** - Interfaccia che determina il contratto di un oggetto di tipo sceneggiato e che mette a disposizione i metodi di esportazione e condivisione dello sceneggiato;

- **Utilizzo** - Viene usata per determinare il contratto di un oggetto di tipo sceneggiato;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Screenplay::ScreenplayImpl.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Presenter::ScreenplayPresenterImpl** - Relazione entrante: l'interfaccia è riferita dalla classe Sivodim::Drama::Presenter::ScreenplayPresenterImpl. Questa classe mantiene un riferimento ad un oggetto Screenplay per poterlo gestire in modo sicuro in relazione agli eventi delle relative Activities;
 - * **Sivodim::Drama::Model::Utilities::XMLParser** - Relazione entrante, la classe è riferita da Sivodim::Drama::Model::Utilities::XMLParser, che ne può istanziare degli oggetti.
- **Sivodim::Drama::Model::Screenplay::ScreenplayImpl**
 - **Descrizione** - Classe che rappresenta l'implementazione di un oggetto di tipo sceneggiato. Ha come attributi una struttura dati contenente un certo numero di oggetti capitolo, una struttura dati contenente un certo numero di oggetti personaggio e un'istanza degli algoritmi di esportazione e condivisione;
 - **Utilizzo** - Viene utilizzata per istanziare un oggetto di tipo sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::Screenplay.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ExportAlgorithm** - Relazione uscente, riferimento alla classe astratta Sivodim::Drama::Model::Screenplay::ExportAlgorithm. La classe istanzia un oggetto ExportAlgorithm per utilizzarne le funzionalità di esportazione in formato audio ($MP3_G$ o WAV_G) o video ($MP4_G$);
 - * **Sivodim::Drama::Model::Chapter::Chapter** - Relazione uscente, riferimento ad una classe che implementa l'interfaccia Sivodim::Drama::Model::Chapter::Chapter. La classe mantiene una lista di Chapter che rappresentano i capitoli dello sceneggiato;
 - * **Sivodim::Libraries::Model::EngineManager::Engine** - Relazione uscente, riferimento all'interfaccia Sivodim::Libraries::Model::EngineManager::Engine per l'ottenimento della sintesi vocale;
 - * **Sivodim::Drama::Model::Character::CharacterContainer** - Relazione uscente, riferimento alla classe Sivodim::Drama::Model::Character::CharacterContainer. La classe istanzia un oggetto CharacterContainer per mantenere la lista dei personaggi relativi allo sceneggiato;
 - * **Sivodim::Drama::Model::Utilities::XMLParser** - Relazione uscente, la classe utilizza Sivodim::Drama::Model::Utilities::XMLParser per le funzionalità di load e save di Screenplay dalla memoria;
 - * **Sivodim::Drama::Model::Screenplay::Mp3Converter** - Relazione uscente, ha un riferimento alla classe Sivodim::Drama::Model::Screenplay::Mp3Converter che la utilizza nella fase di esportazione;
 - * **Sivodim::Drama::Model::Screenplay::AudioConcatenator** - Relazione uscente, ha un riferimento alla classe Sivodim::Drama::Model::Screenplay::AudioConcatenator che la utilizza nella fase di esportazione;

- * **Sivodim::Drama::Model::Screenplay::AudioMixer** - Relazione uscente, ha un riferimento alla classe Sivodim::Drama::Model::Screenplay::AudioMixer che la utilizza nella fase di esportazione.
- **Sivodim::Drama::Model::Screenplay::ExportAlgorithm**
 - **Descrizione** - Classe astratta che racchiude gli algoritmi di esportazione;
 - **Utilizzo** - Viene definito il contratto di un algoritmo di esportazione;
 - **Classi figlie**
 - * Sivodim::Drama::Model::Screenplay::AudioExport;
 - * Sivodim::Drama::Model::Screenplay::VideoExport.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, la classe è riferita dalla classe Sivodim::Drama::Model::Screenplay::ScreenplayImpl. Viene istanziata per invocare una fra le opzioni di esportazione dello sceneggiato (MP3_G, WAV_G o MP4_G).
- **Sivodim::Drama::Model::Screenplay::AudioExport**
 - **Descrizione** - Classe che implementa l'algoritmo di esportazione di un file audio;
 - **Utilizzo** - Viene usata per esportare nella memoria del dispositivo il file audio ottenuto da uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::ExportAlgorithm.
- **Sivodim::Drama::Model::Screenplay::VideoExport**
 - **Descrizione** - Classe che implementa l'algoritmo di esportazione di un file video;
 - **Utilizzo** - Viene usata per esportare nella memoria del dispositivo il file video ottenuto da uno sceneggiato;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::ExportAlgorithm.
- **Sivodim::Drama::Model::Screenplay::Mp3Converter**
 - **Descrizione** - Classe concreta che estende FfmpegConnector e ne implementa il metodo astratto getCommand per gestire la conversione audio di file WAVG in MP3G;
 - **Utilizzo** - Viene utilizzata dalla classe ScreenplayImpl nella fase di esportazione per convertire l'audio in MP3;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::FfmpegConnector.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, è riferita da Sivodim::Drama::Model::Screenplay::ScreenplayImpl che la utilizza nella fase di esportazione.
- **Sivodim::Drama::Model::Screenplay::AudioConcatenator**

- **Descrizione** - Classe concreta che estende FfmpegConnector e ne implementa il metodo astratto getCommand per gestire la concatenazione di file audio di tipo WAVG;
 - **Utilizzo** - Viene utilizzata dalla classe ScreenplayImpl nella fase di esportazione per concatenare i file audio generati a partire dalle battute;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::FfmpegConnector.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, è riferita da Sivodim::Drama::Model::Screenplay::ScreenplayImpl che la utilizza nella fase di esportazione.
- **Sivodim::Drama::Model::Screenplay::AudioMixer**
 - **Descrizione** - Classe concreta che estende FfmpegConnector e ne implementa il metodo astratto getCommand per gestire la sovrapposizione di tracce audio;
 - **Utilizzo** - Viene utilizzata dalla classe ScreenplayImpl nella fase di esportazione per sovrapporre la traccia audio contenente la sintesi delle battute alla soundtrack del capitolo;
 - **Classi ereditate**
 - * Sivodim::Drama::Model::Screenplay::FfmpegConnector.
 - **Relazioni con altre classi**
 - * **Sivodim::Drama::Model::Screenplay::ScreenplayImpl** - Relazione entrante, è riferita da Sivodim::Drama::Model::Screenplay::ScreenplayImpl che la utilizza nella fase di esportazione.

5 Diagrammi di attività

In questa sezione vengono presentati i diagrammi di attività utili a descrivere le interazioni dell’utente con le componenti *software* del progetto SiVoDiM. I diagrammi sono suddivisi in due macro-categorie: applicazione per gli sceneggiati e applicazione di configurazione. Nell’ordine, vengono proposti per ognuna:

- Un diagramma ad alto livello che descrive tutte le attività possibili;
- Dei sotto-diagrammi specifici per ogni attività che può essere scomposta in sottoparti.

Per semplificare la lettura dei diagrammi, le attività più ad alto livello sono evidenziate in colore più scuro allo scopo di differenziarle dai riquadri che rappresentano singole azioni. Tali attività vengono quindi descritte attraverso dei sotto-diagrammi.

5.1 Applicazione per gli sceneggiati

5.1.1 Attività di alto livello

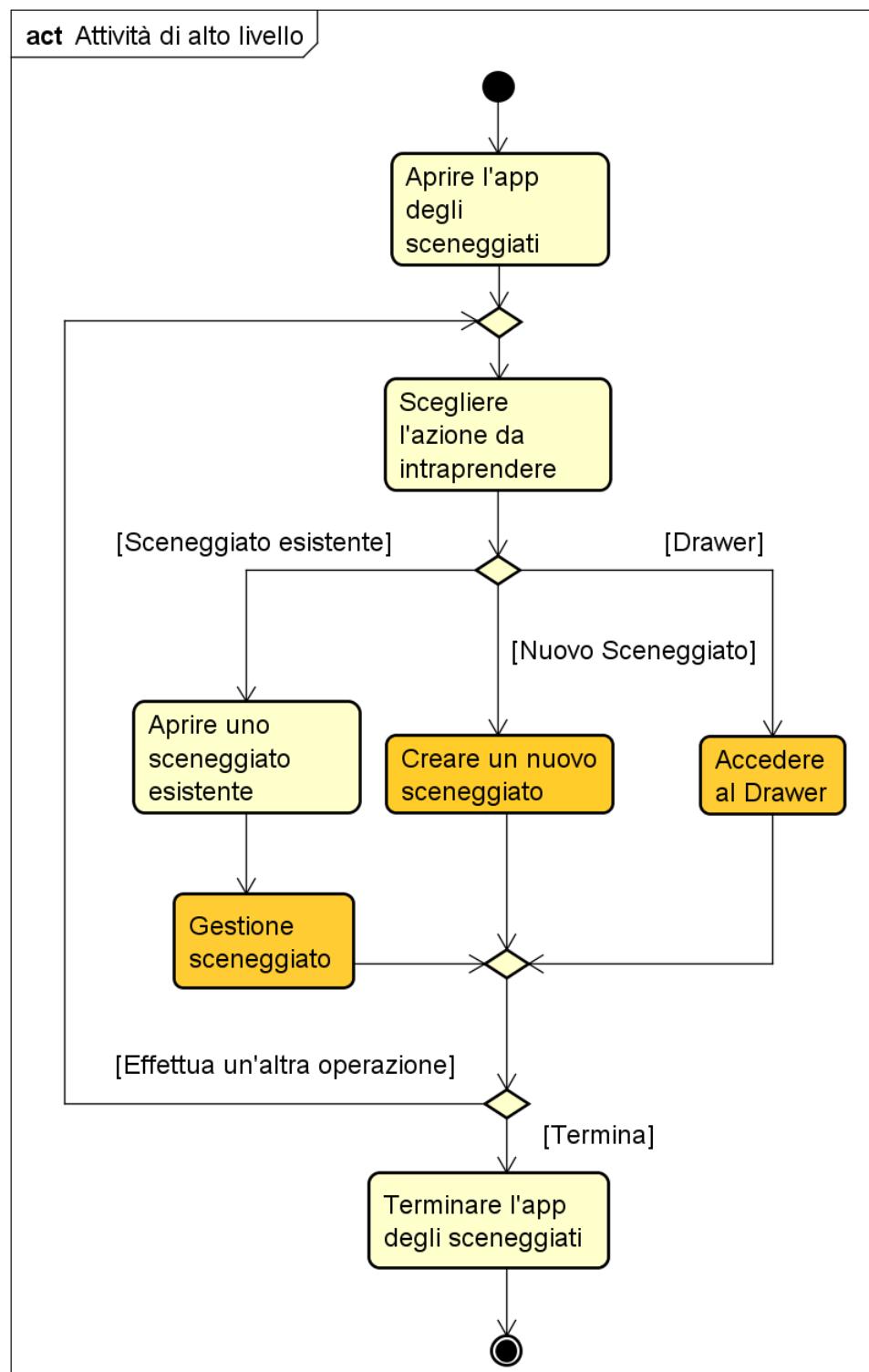


Figura 22: Diagramma delle attività di alto livello dell'applicazione degli sceneggiati

Come illustrato in Figura 22, a seguito dell'apertura dell'applicazione l'utente può eseguire le seguenti azioni principali:

- Aprire uno sceneggiato già esistente per poterlo gestire e apportare delle modifiche (di default l'applicazione fornisce uno sceneggiato di prova);
- Creare un nuovo sceneggiato;
- Accedere al menu *drawer*.

5.1.2 Accesso al Drawer

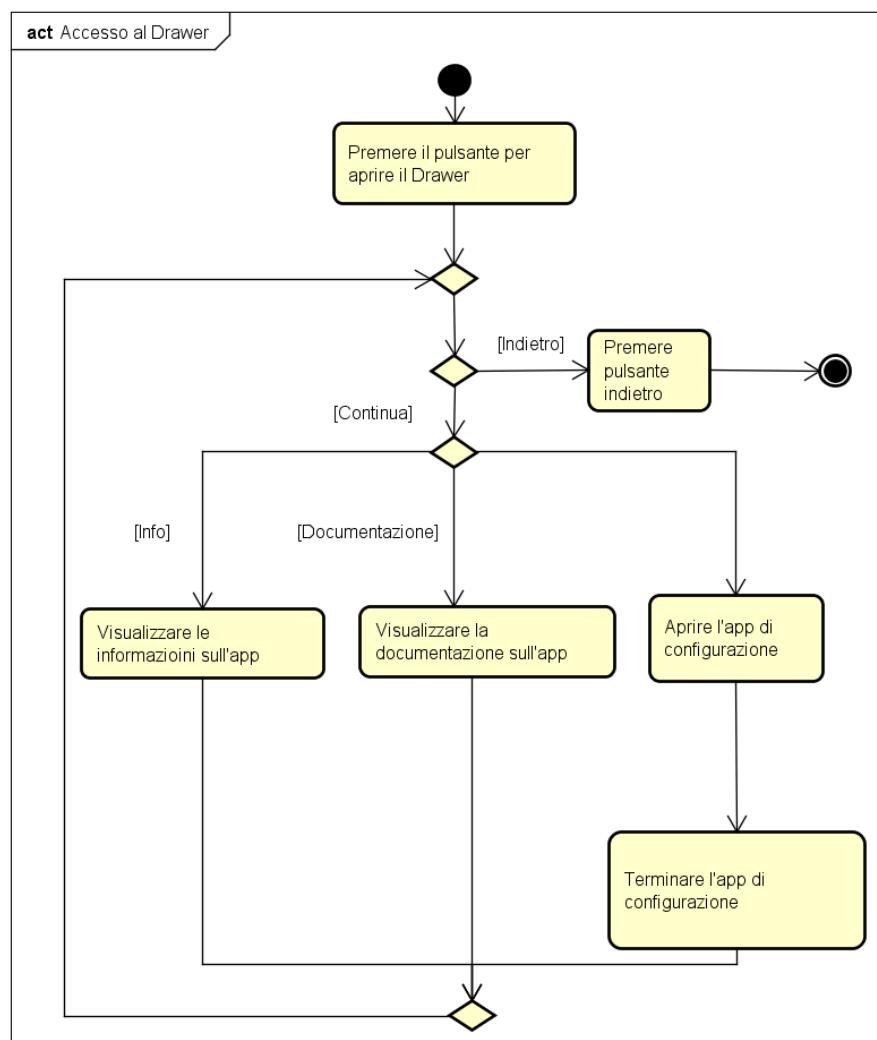


Figura 23: Diagramma delle attività per l'accesso al Drawer

L'accesso al Drawer_G illustrata in Figura 23 prevede le seguenti azioni:

- Accesso al Drawer_G attraverso la pressione dell'apposito pulsante;
- L'utente può decidere di procedere con una delle operazioni rese disponibili dal Drawer_G o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Visualizzare le informazioni dell'applicazione;
- Visualizzare la documentazione relativa all'applicazione;
- Aprire l'applicazione di configurazione.

5.1.3 Creazione sceneggiato

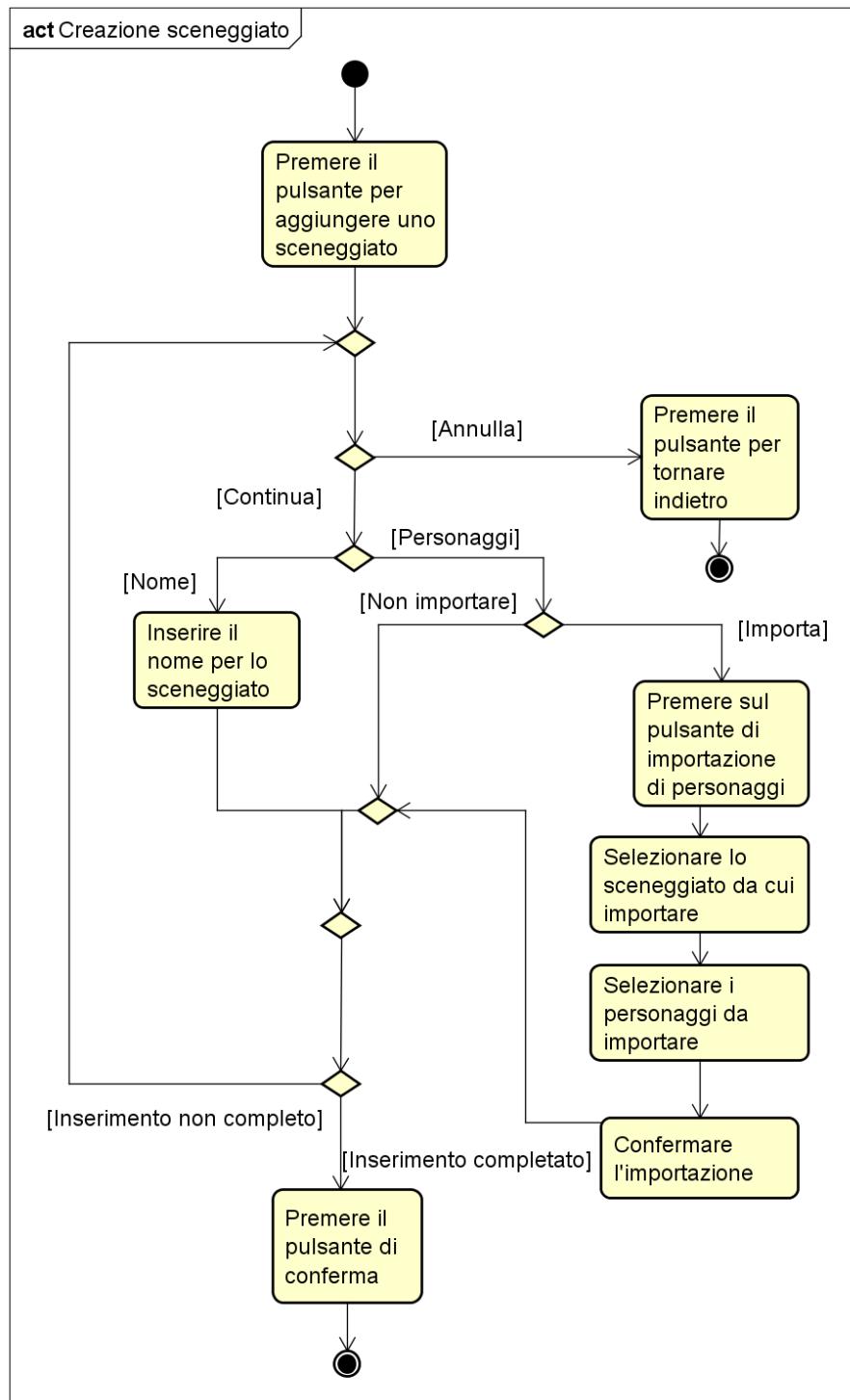


Figura 24: Diagramma delle attività per la creazione di uno sceneggiato

La creazione di un nuovo sceneggiato illustrata in Figura 24 prevede le seguenti azioni:

- Avvio dell'operazione di creazione attraverso la pressione dell'apposito pulsante;
- L'utente può decidere di procedere con la creazione dello sceneggiato o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Inserimento di un nome da assegnare allo sceneggiato (operazione obbligatoria per procedere con la creazione);
- Importazione di una lista di personaggi già esistenti appartenente a un altro sceneggiato già creato. È possibile importare l'intera lista o selezionare solamente alcuni dei personaggi contenuti nella stessa;
- Conferma dell'operazione di creazione attraverso la pressione di un apposito pulsante.

5.1.4 Gestione sceneggiato

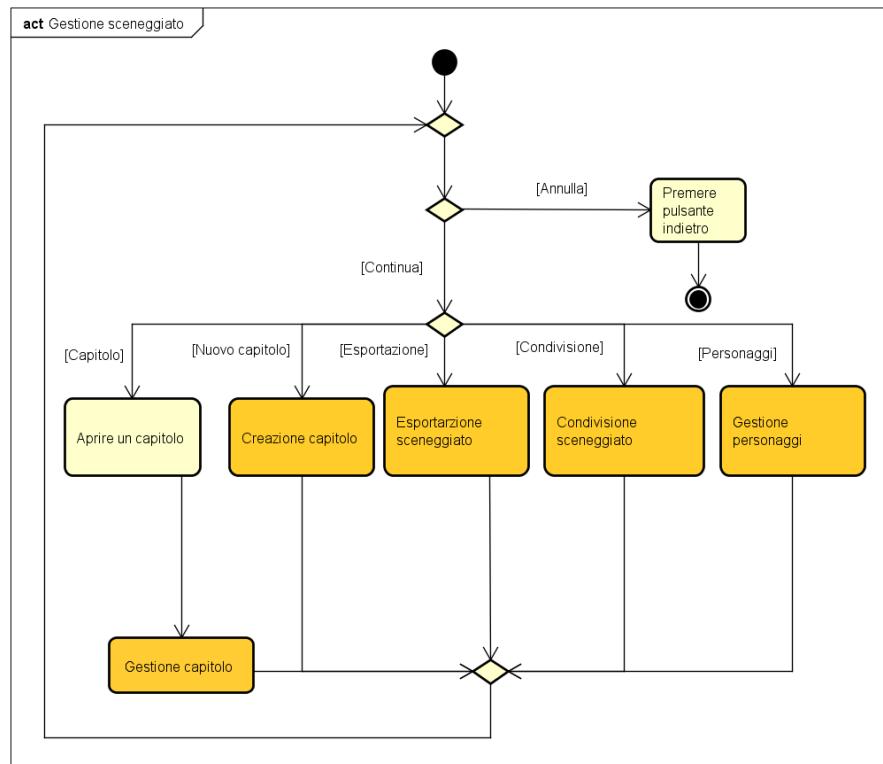


Figura 25: Diagramma delle attività per la gestione di uno sceneggiato

La gestione di uno sceneggiato illustrata in Figura 25 prevede le seguenti azioni:

- Apertura di un capitolo esistente;
- Creazione di un nuovo capitolo;
- Gestione di un capitolo creato;
- Esportazione di uno sceneggiato;
- Condivisione di uno sceneggiato;
- Gestione dei personaggi.

5.1.5 Creazione capitolo

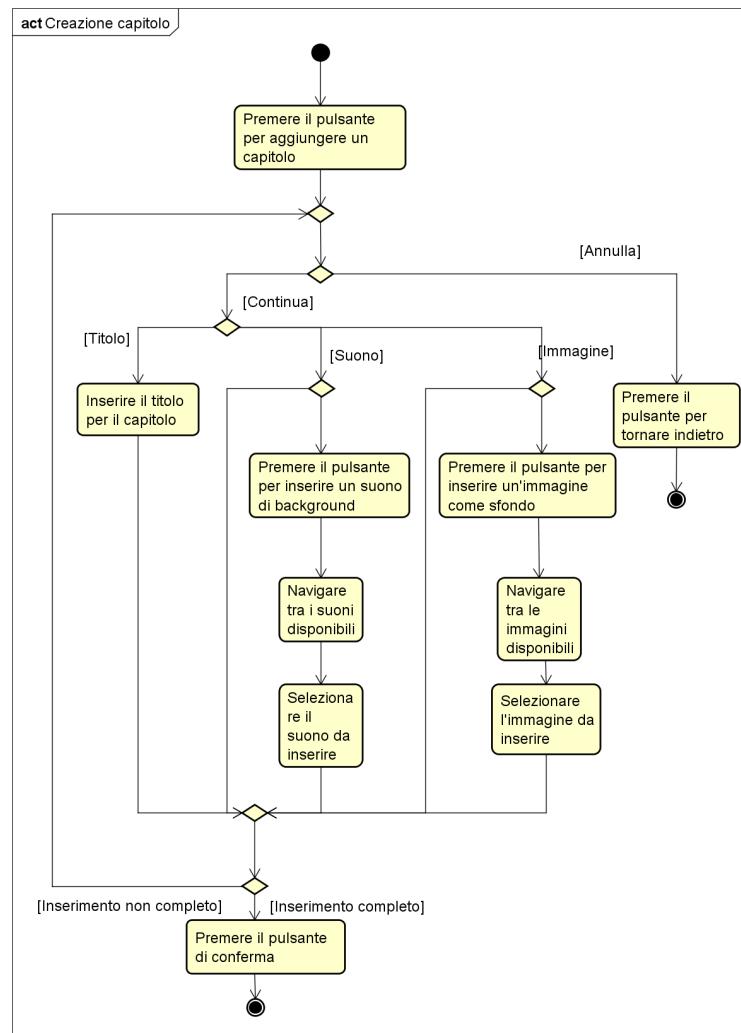


Figura 26: Diagramma delle attività per la creazione di un capitolo

La creazione di un nuovo capitolo illustrata in Figura 26 prevede le seguenti azioni:

- Avvio dell'operazione di creazione attraverso la pressione dell'apposito pulsante;
- L'utente può decidere di procedere con la creazione del capitolo o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Inserimento di un titolo da assegnare al capitolo (operazione obbligatoria per procedere con la creazione);
- Inserimento di un suono o un'immagine di *background* attraverso la pressione di un pulsante;
- Conferma dell'operazione di creazione attraverso la pressione di un pulsante.
- Avvio dell'inserimento di un'immagine o un suono di *background* attraverso la pressione di un pulsante. A questo punto l'utente può:
 - Annullare l'operazione attraverso la pressione del tasto *back*;
 - Procedere con l'inserimento selezionando un *file* fra le immagini o gli audio presenti nella memoria del telefono e confermare quindi l'operazione.
- Conferma dell'operazione di creazione attraverso la pressione di un apposito pulsante.

5.1.6 Condivisione sceneggiato

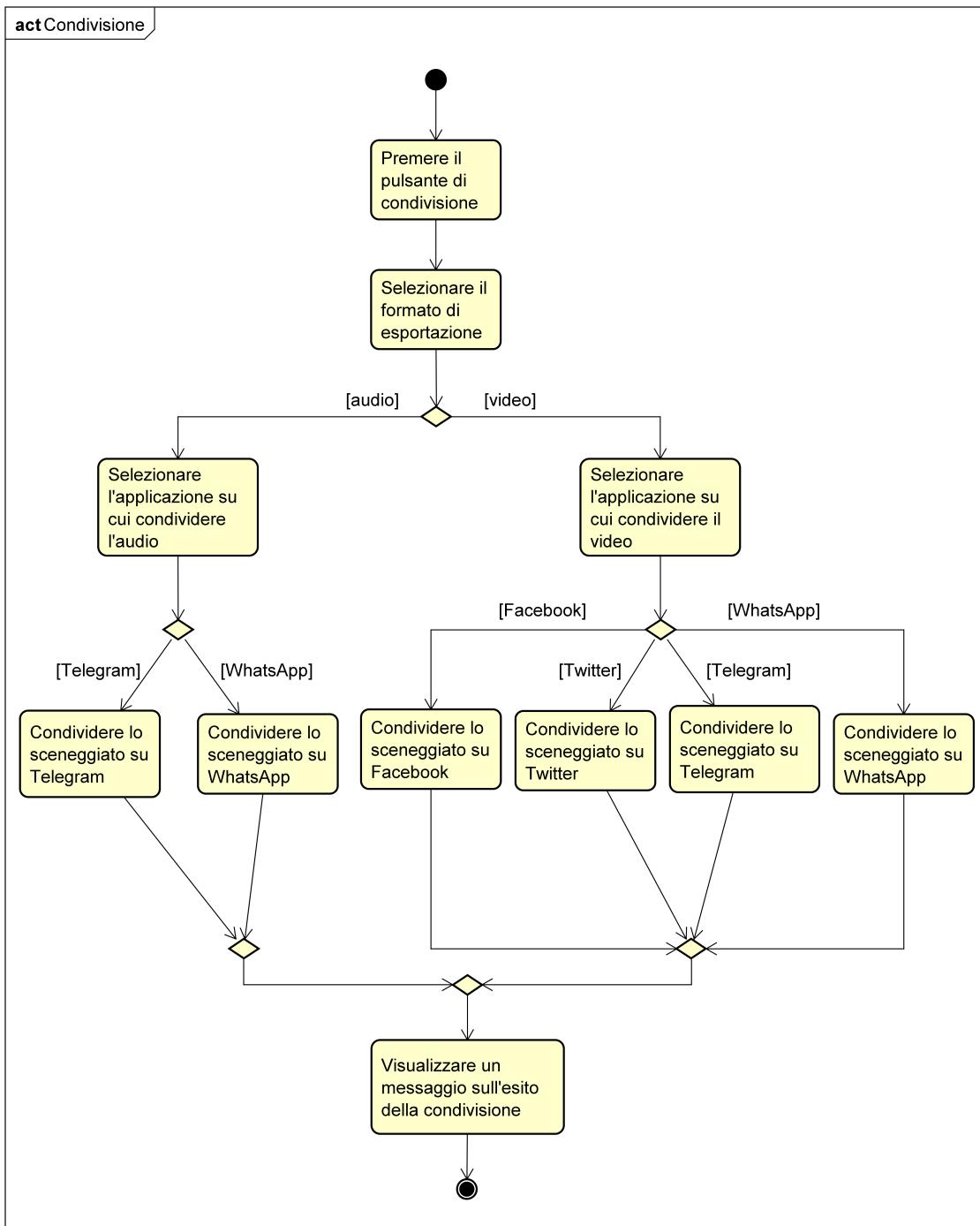


Figura 27: Diagramma delle attività per la condivisione

La condivisione di uno sceneggiato, illustrata in Figura 27, prevede le seguenti azioni:

- Avvio dell'operazione di condivisione attraverso la pressione dell'apposito pulsante;
- Selezione del formato di esportazione per procedere con la condivisione del *file* generato;
- L'utente può decidere se condividere un *file* audio o video e di conseguenza si aprono diverse possibilità di condivisione:
 - **Audio:**
 - * Condivisione dello sceneggiato su applicazioni di messaggistica (Telegram_G e Whatsapp_G).
 - **Video:**
 - * Condivisione dello sceneggiato su applicazioni di messaggistica (Telegram e Whatsapp);
 - * Condivisione dello sceneggiato su Facebook e Twitter.
- Visualizzazione di un messaggio che notifica all'utente l'esito della condivisione.

5.1.7 Esportazione sceneggiato

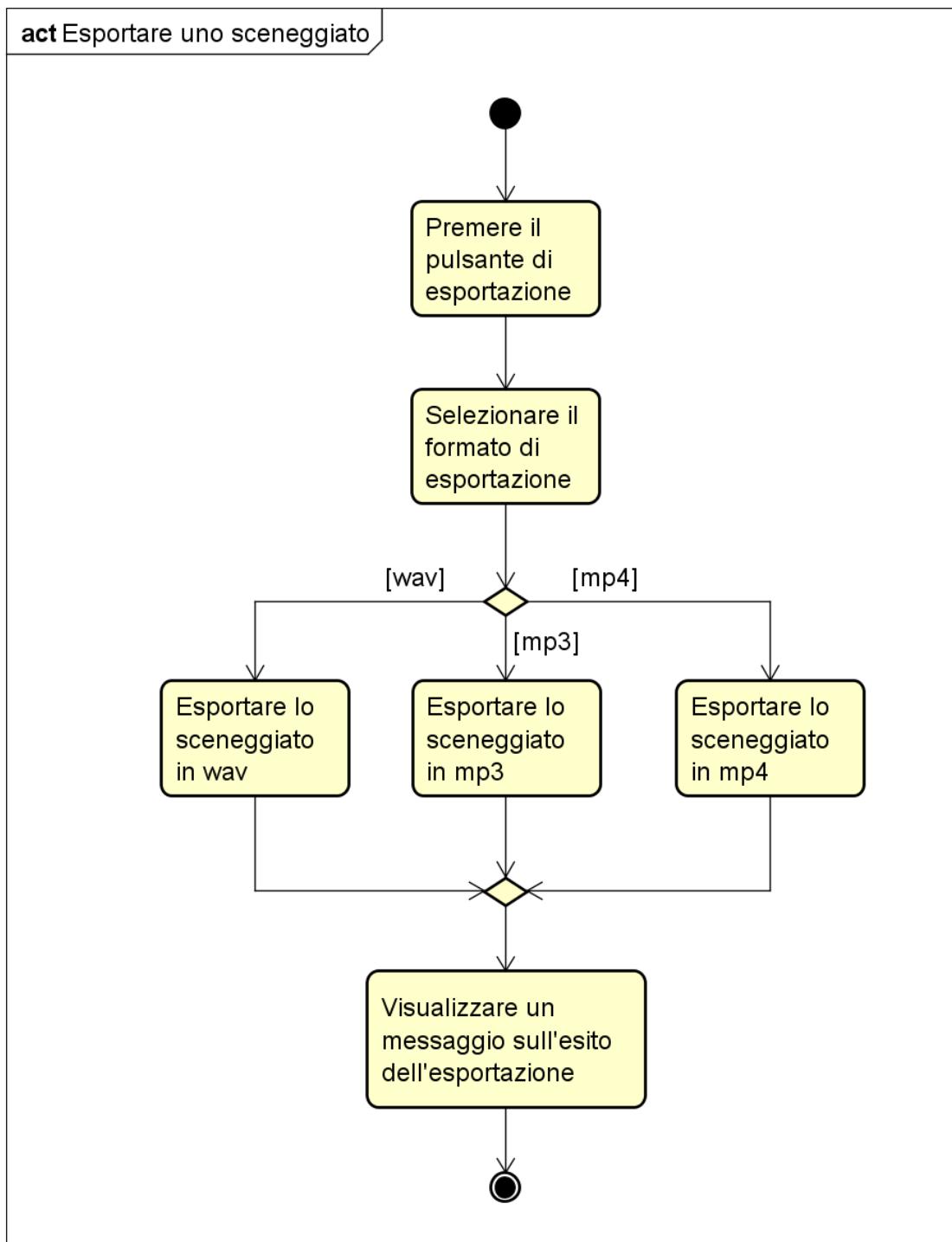


Figura 28: Diagramma delle attività per l'esportazione

L'esportazione di uno sceneggiato illustrata in Figura 28 prevede le seguenti azioni:

- Avvio dell'operazione di esportazione attraverso la pressione dell'apposito pulsante;
- Selezione del formato di esportazione. L'utente può decidere se condividere un *file* audio o video attraverso diverse opzioni di esportazione:
 - **Audio:**
 - * Esportazione dello sceneggiato in formato WAV_G o MP3_G .
 - **Video:**
 - * Esportazione dello sceneggiato in formato MP4_G .
- Visualizzazione di un messaggio che notifica all'utente l'esito dell'esportazione.

5.1.8 Gestione personaggio

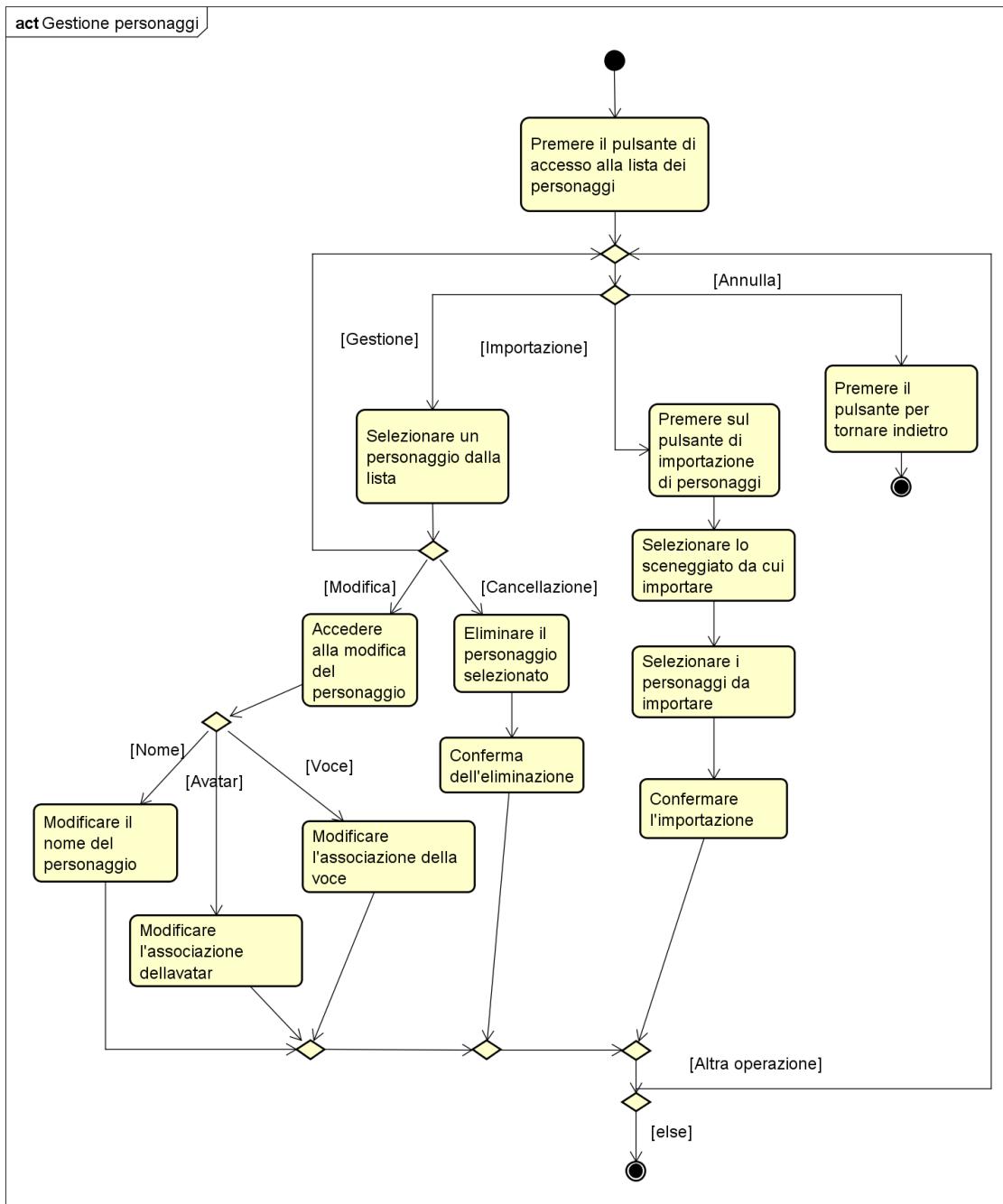


Figura 29: Diagramma delle attività per la gestione di un personaggio

La gestione di un personaggio illustrata in Figura 29 prevede le seguenti azioni da effettuare su ogni personaggio già creato:

- L'utente può decidere se procedere con le operazioni di gestione del personaggio o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Dalla lista dei personaggi l'utente può selezionare un personaggio e procedere con le seguenti operazioni:
 - Modifica del personaggio selezionato. L'attività prevede di scegliere fra le seguenti azioni:
 - * Modifica del nome associato al personaggio;
 - * Modifica della voce associata al personaggio;
 - * Modifica dell'*avatar_G* associato al personaggio.
 - Eliminazione del personaggio selezionato a cui segue una conferma che l'utente deve approvare per procedere con l'operazione.
- Importazione di una lista di personaggi da uno sceneggiato già esistente;
- Creazione di un nuovo personaggio a cui devono essere necessariamente associati: nome, voce e *avatar_G*.

5.1.9 Gestione capitolo

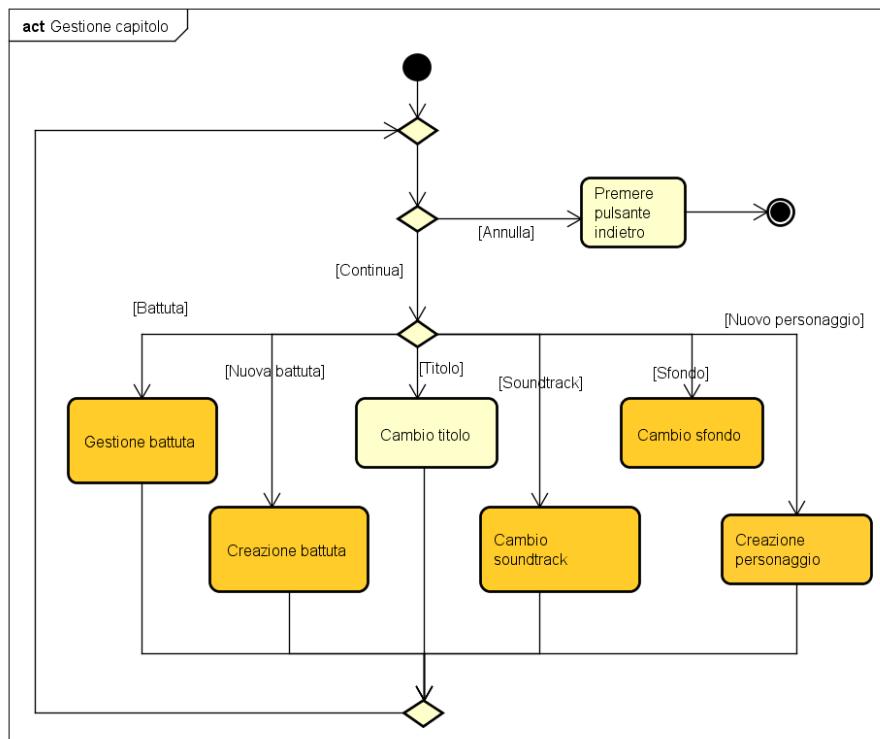


Figura 30: Diagramma delle attività per la gestione di uno capitolo

La gestione di un capitolo illustrata in Figura 30 prevede le seguenti azioni:

- Gestire una battuta creata;
- Aggiungere una nuova battuta;
- Cambiare il titolo del capitolo;
- Creare un nuovo personaggio;
- Cambiare la *soundtrack* del capitolo;
- Cambiare lo sfondo del capitolo.

5.1.10 Creazione battuta

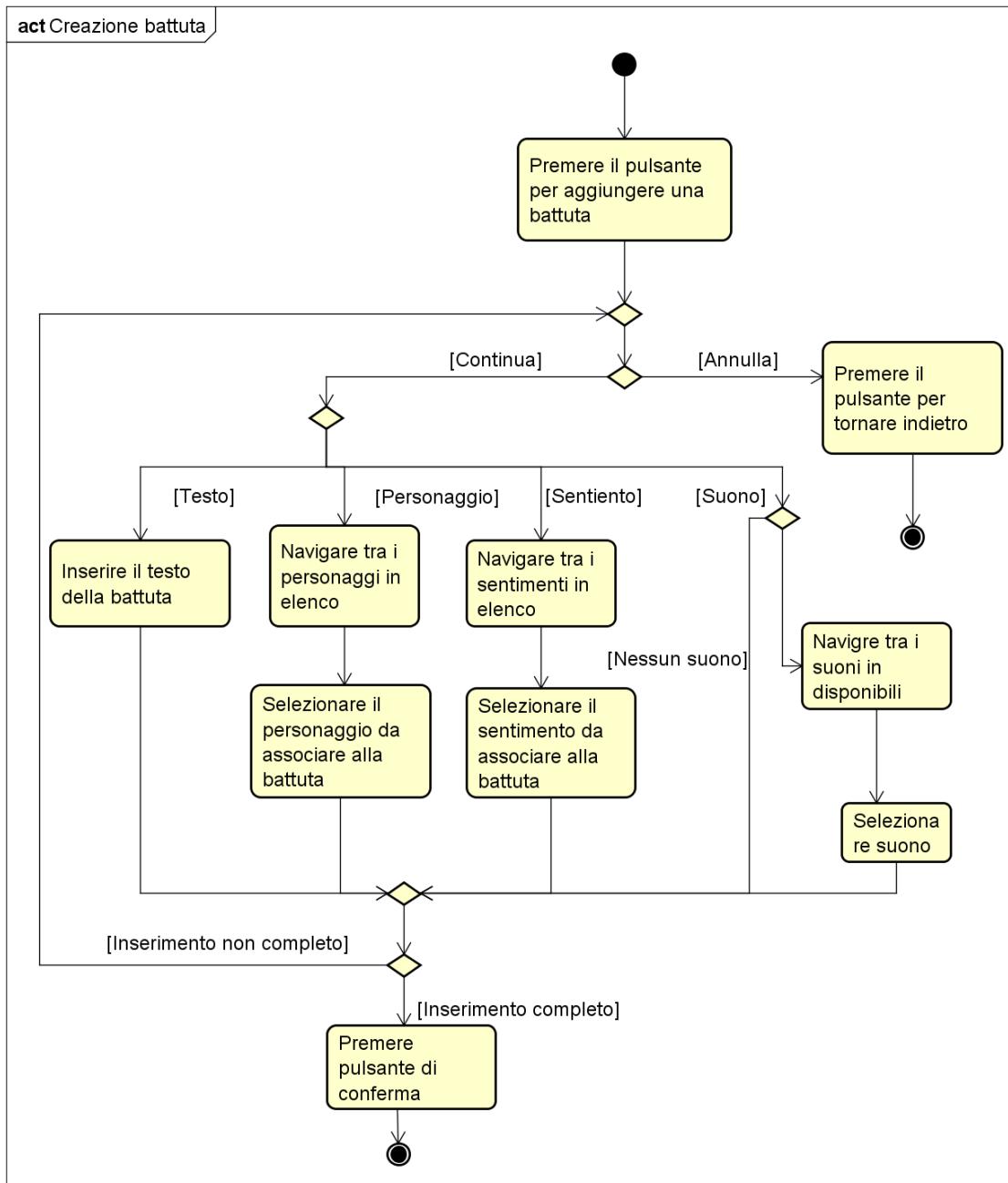


Figura 31: Diagramma delle attività per la creazione di una battuta

La creazione di una nuova battuta illustrata in Figura 31 prevede le seguenti azioni:

- Avvio dell'operazione di creazione attraverso la pressione dell'apposito pulsante;
- L'utente può decidere di procedere con la creazione della battuta o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Inserimento del testo da assegnare alla battuta (operazione obbligatoria per procedere con la creazione);
- Associazione di un personaggio alla battuta (operazione obbligatoria per procedere con la creazione);
- Associazione di un sentimento alla battuta: il sentimento modifica il modo in cui il personaggio pronuncia il testo della battuta. Il sentimento viene scelto da una lista predefinita fornita direttamente dall'applicazione;
- Associazione di un suono alla battuta: l'effetto sonoro viene scelto da una lista di effetti predefiniti forniti direttamente dall'applicazione;
- Conferma dell'operazione di creazione attraverso la pressione di un apposito pulsante.

5.1.11 Gestione battuta

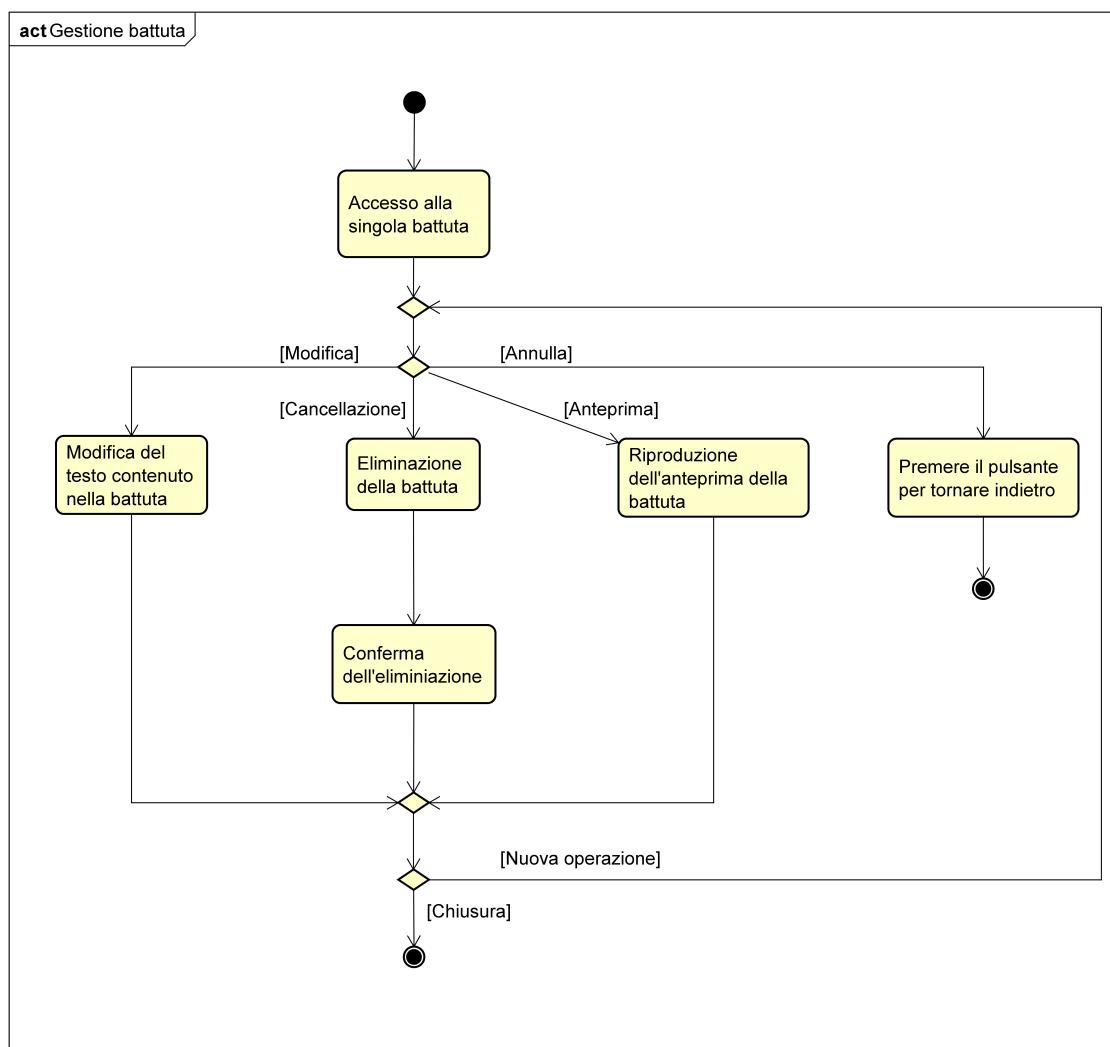


Figura 32: Diagramma delle attività per la gestione di una battuta

La gestione di una battuta illustrata in Figura 32 prevede le seguenti azioni da effettuare su una battuta già creata:

- L'utente può decidere se procedere con le operazioni di gestione della battuta o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Modifica del testo contenuto nella battuta;
- Eliminazione della battuta, seguita dalla conferma attraverso un apposito messaggio che richiede approvazione;
- Riproduzione di un'anteprima della battuta con il testo e la voce impostati;
- Ripetere alcune di queste azioni o terminare l'attività di gestione.

5.1.12 Cambio sfondo

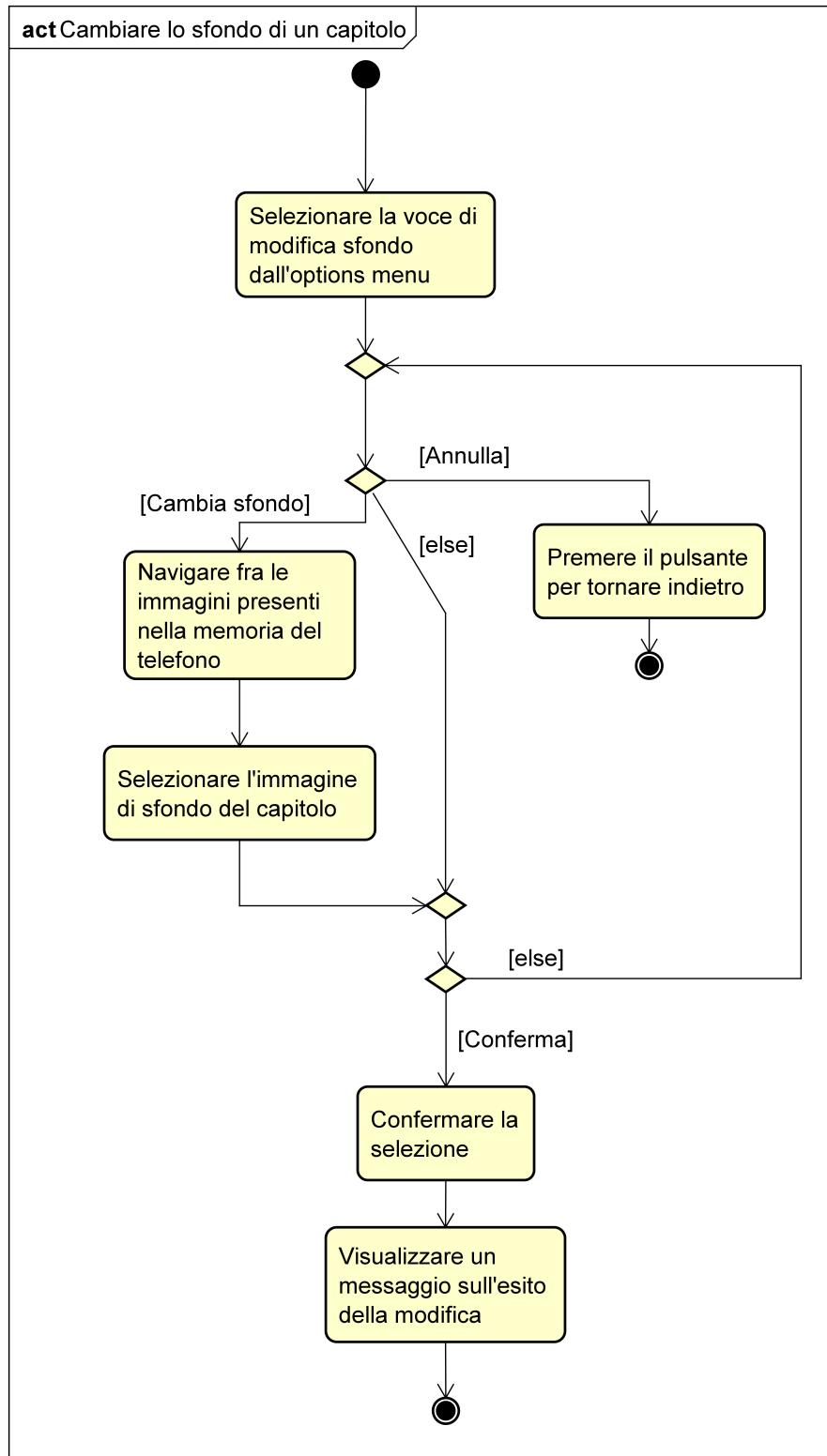


Figura 33: Diagramma delle attività per la modifica dello sfondo di un capitolo

La modifica dello sfondo di una battuta illustrata in Figura 33 prevede le seguenti azioni:

- Selezione della voce di modifica dello sfondo dal menu delle opzioni presente nella schermata relativa al capitolo;
- L'utente può decidere se procedere con il cambio dello sfondo o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Navigazione fra le immagini presenti nella memoria del telefono;
- Selezione di un'immagine da associare allo sfondo del capitolo;
- Conferma della selezione e visualizzazione di un messaggio che espone all'utente l'esito della modifica.

5.1.13 Cambio soundtrack

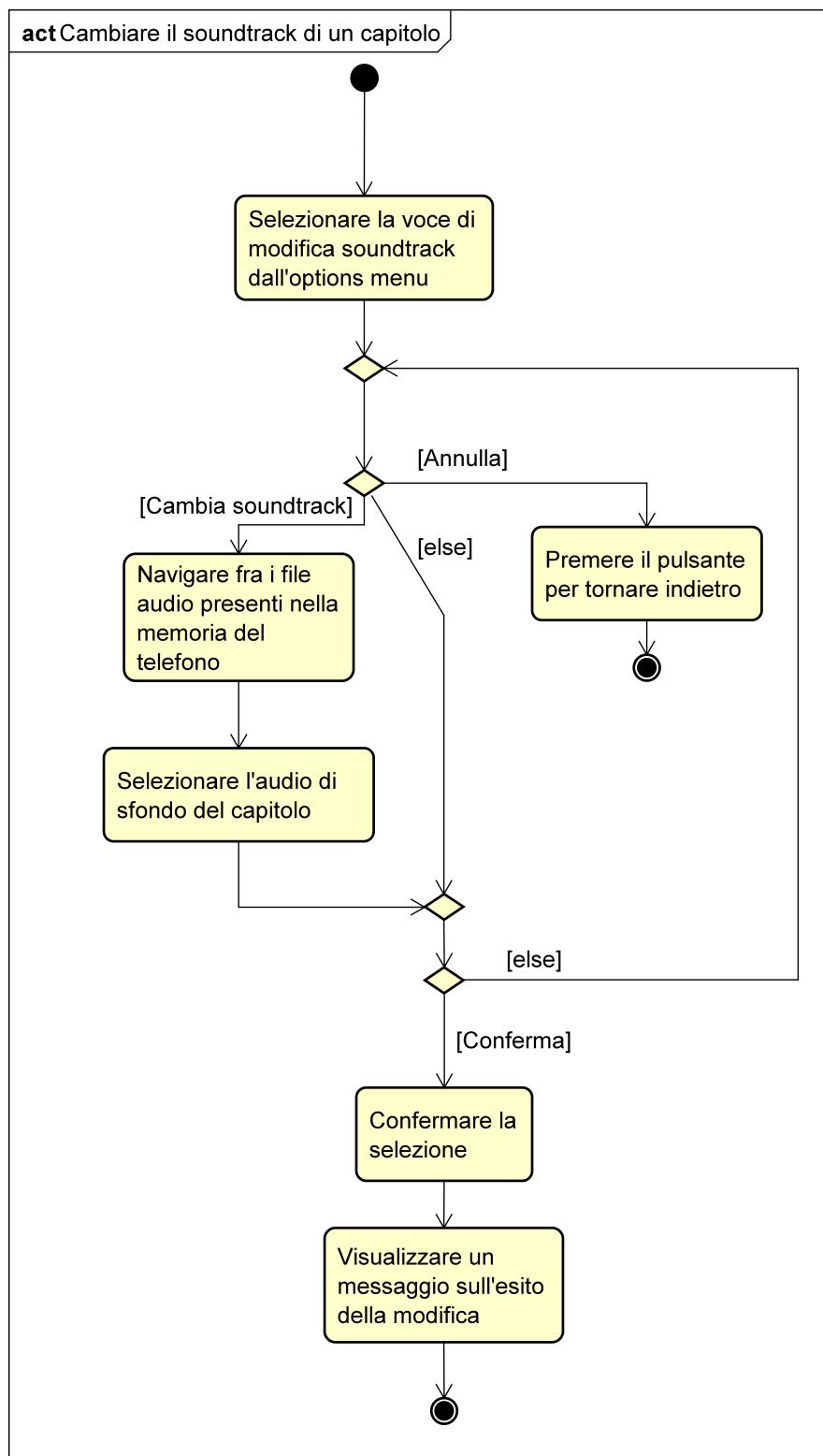


Figura 34: Diagramma delle attività per la modifica del sottofondo sonoro di un capitolo

La modifica della *soundtrack* di una battuta, illustrata in Figura 34 prevede le seguenti azioni:

- Selezione della voce di modifica della *soundtrack* dal menu delle opzioni presente nella schermata relativa al capitolo;
- L'utente può decidere se procedere con il cambio della *soundtrack* o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Navigazione fra i *file* audio presenti nella memoria del telefono;
- Selezione di un *file* audio da associare alla *soundtrack* del capitolo;
- Conferma della selezione e visualizzazione di un messaggio che notifica all'utente l'esito della modifica.

5.1.14 Creazione personaggio

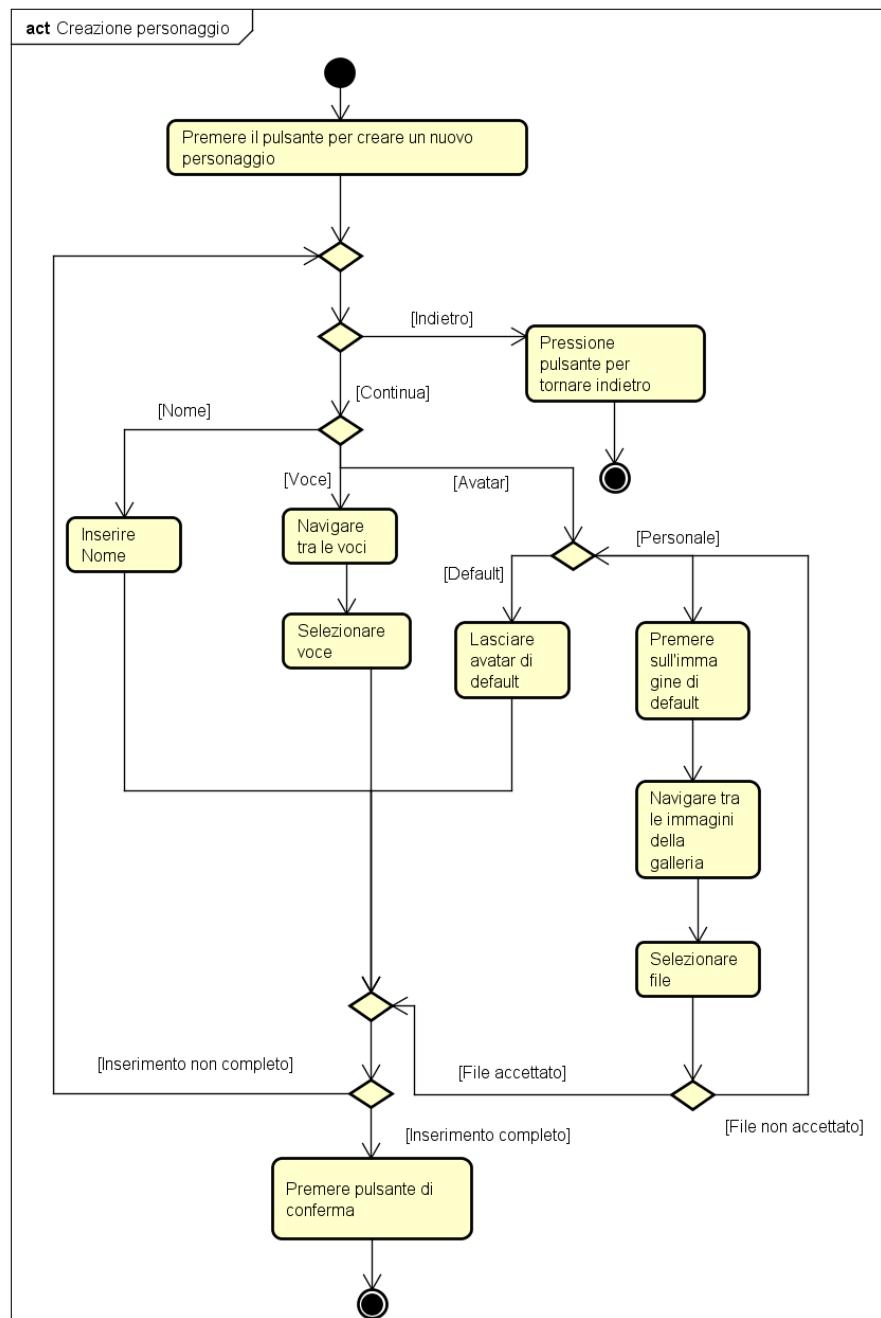


Figura 35: Diagramma delle attività per la creazione di un nuovo personaggio

La creazione di un personaggio, illustrata in Figura 35 prevede le seguenti azioni:

- Pressione del pulsante per creare un nuovo personaggio;
- L'utente può decidere se procedere con la creazione di un personaggio o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Inserire il nome del nuovo personaggio;
- Selezione di una voce da assegnare al personaggio tra quelle disponibili in elenco;
- L'utente può scegliere se assegnare un avatar_G al personaggio, scegliendo un file immagine dalla sua galleria che soddisfi i requisiti di dimensione, oppure lasciare l'avatar di *default*;
- Conferma la creazione del personaggio premendo il pulsante apposito.

5.2 Applicazione di configurazione

5.2.1 Attività di alto livello

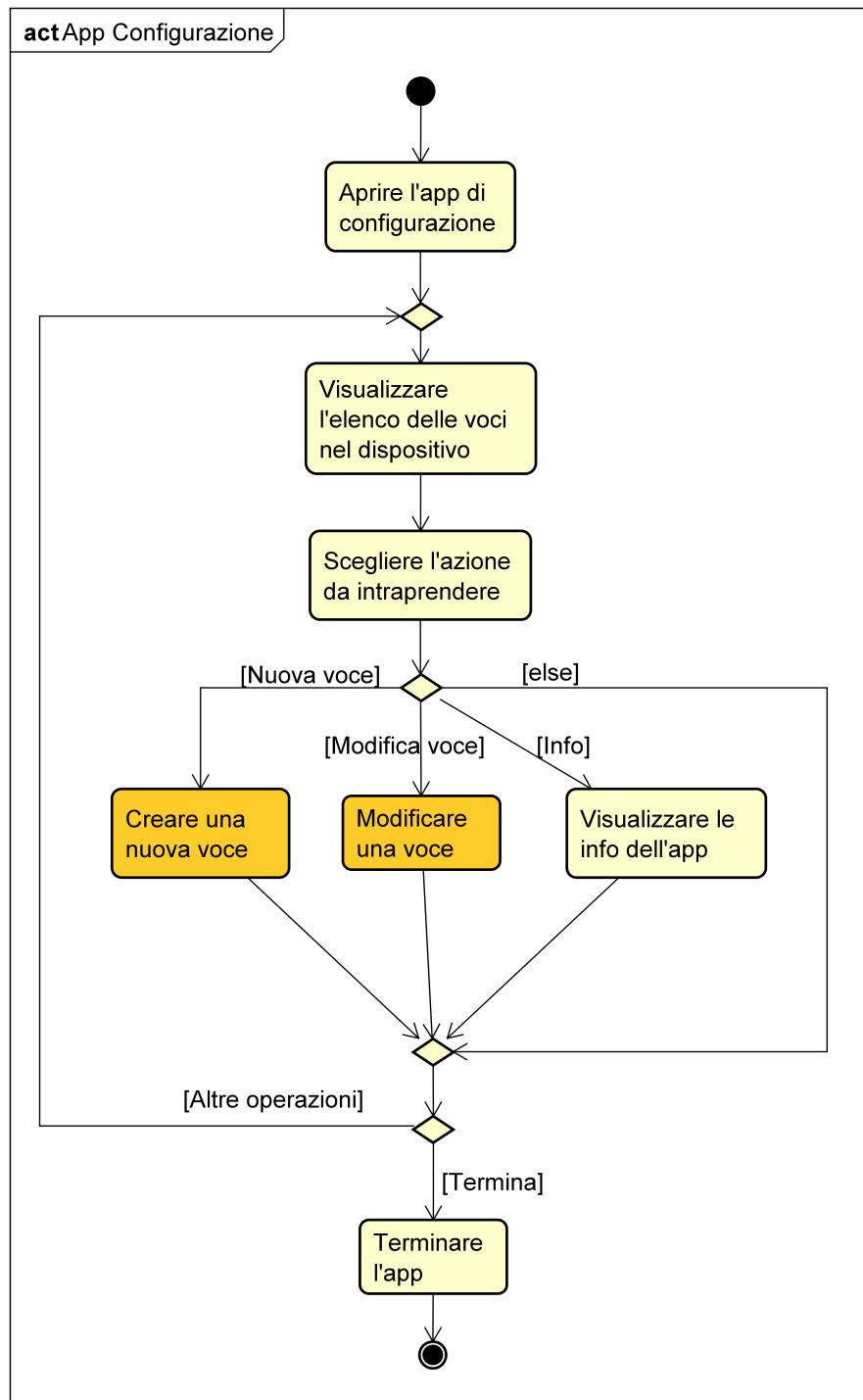


Figura 36: Diagramma delle attività di alto livello dell'app di configurazione

Come illustrato in Figura 36, a seguito dell'apertura dell'app l'utente può eseguire le seguenti azioni principali:

- Visualizzazione dell'elenco delle voci disponibili nel dispositivo (ne verranno fornite alcune di *default*);
- Creazione di una nuova voce attraverso la pressione di un apposito pulsante;
- Accesso alla modifica di una singola voce;
- Accedere al menu *drawer* per:
 - Visualizzazione della documentazione contenente una guida per il corretto utilizzo dell'app;
 - Visualizzazione delle informazioni sull'applicazione e dei suoi creatori.

5.2.2 Creazione voce

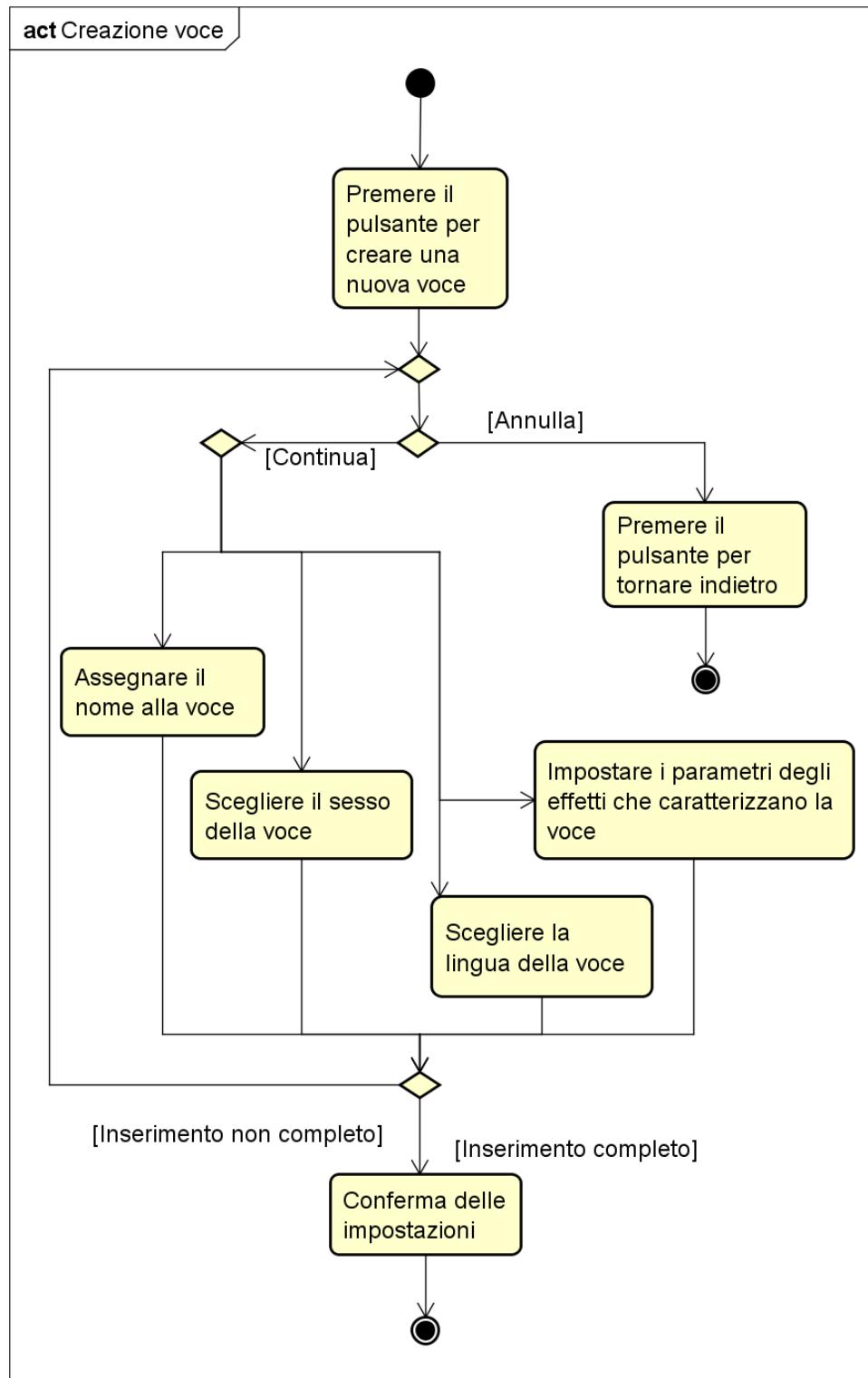


Figura 37: Diagramma delle attività per la creazione di una voce

La creazione di una nuova voce illustrata in Figura 37 prevede le seguenti azioni:

- Avvio dell'operazione di creazione attraverso la pressione dell'apposito pulsante;
- L'utente può decidere di procedere con la creazione della voce o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Inserimento di un nome da assegnare alla voce (operazione obbligatoria per procedere con la creazione);
- Scelta del sesso da associare alla voce (operazione obbligatoria per procedere con la creazione);
- Scelta della lingua da associare alla voce (operazione obbligatoria per procedere con la creazione);
- Impostazione dei parametri relativi agli effetti che caratterizzano una voce (operazione obbligatoria per procedere con la creazione). Verranno forniti dei *preset* per facilitare la configurazione di tali parametri;
- Conferma dell'operazione di creazione attraverso la pressione di un apposito pulsante.

5.2.3 Modifica voce

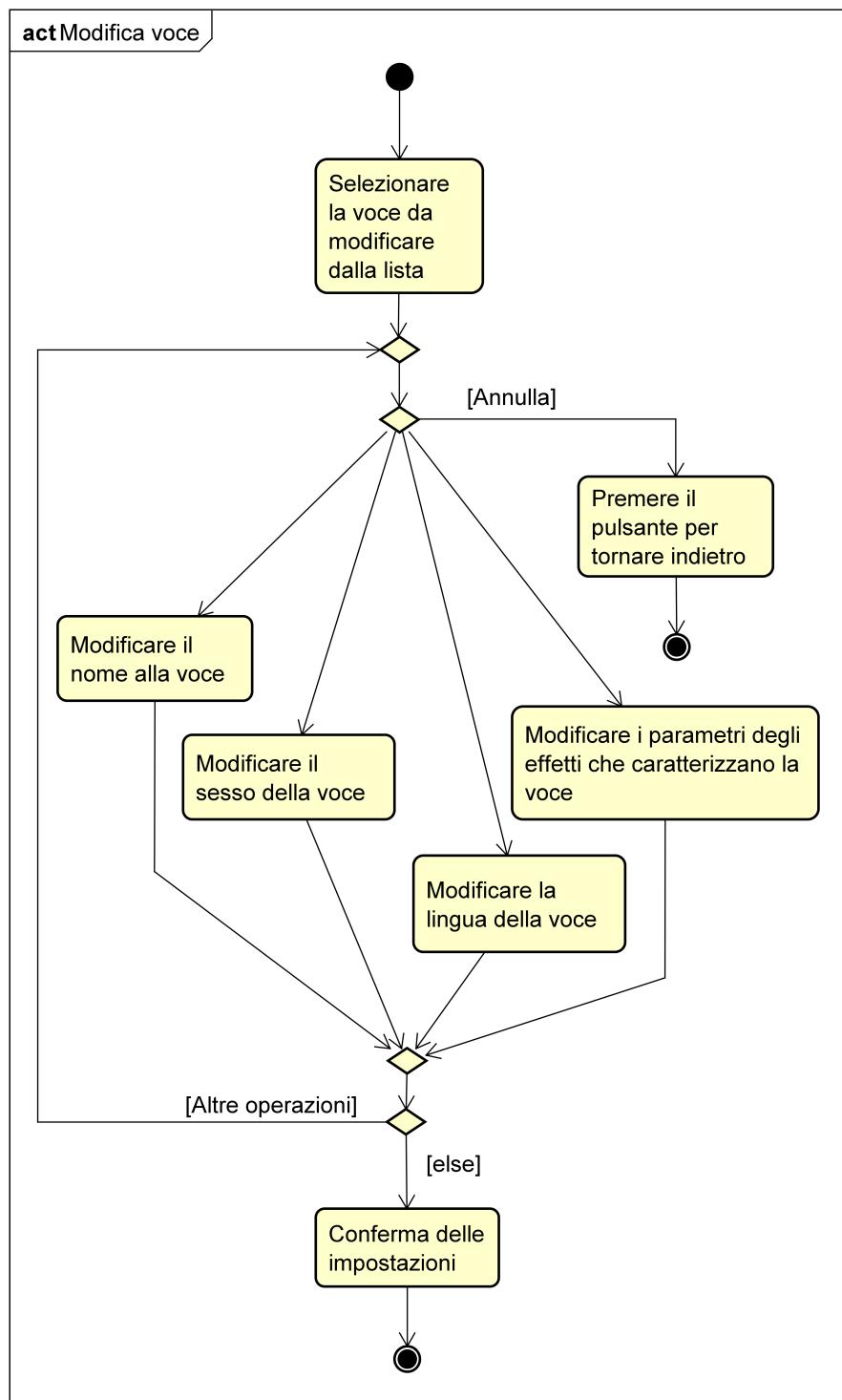


Figura 38: Diagramma delle attività per la modifica di una voce

La modifica di una voce preesistente illustrata in Figura 38 prevede le seguenti azioni:

- Selezione della voce da modificare dalla lista contenente tutte le voci;
- L'utente può decidere di procedere con la modifica della voce o di tornare alla schermata precedente attraverso la pressione del tasto *back*;
- Modifica del nome associato alla voce;
- Modifica del sesso associato alla voce;
- Modifica della lingua associata alla voce;
- Modifica dei parametri relativi agli effetti che caratterizzano una voce;
- Conferma dell'operazione di modifica attraverso la pressione di un apposito pulsante.

6 Design pattern

Nell'ambito dell'ingegneria del software, un *design pattern_G* è una soluzione progettuale generale a un problema ricorrente. L'utilizzo di un *design pattern* semplifica l'attività di progettazione, favorisce il riutilizzo del codice e rende l'architettura più manutenibile. Tali *best-practice_G* sono suddivise in:

- **Design pattern architetturali:** definiscono l'architettura del software al livello più alto e generale;
- **Design pattern creazionali:** rendono un sistema indipendente dall'implementazione concreta delle sue componenti, nascondendo i tipi concreti e i dettagli di composizione e creazione; riducono accoppiamento e flessibilità;
- **Design pattern strutturali:** affrontano problemi che riguardano la composizione di classi e oggetti; consentono di riutilizzare classi preesistenti, fornendo un'interfaccia più adatta e sfruttando l'ereditarietà e l'aggregazione;
- **Design pattern comportamentali:** definiscono soluzioni per le funzioni svolte dagli oggetti e per le interazioni che intercorrono fra di essi.

Per maggiori informazioni sul significato dei singoli *design pattern_G* utilizzati, si veda l'Appendice A. Per la realizzazione del progetto SiVoDiM si sono scelti i *design pattern* presentati nelle sezioni seguenti.

6.1 Design pattern architetturali

6.1.1 Model-view-presenter

- **Scopo:** il principio di *separation of concerns* non è un punto forte dei sistemi operativi Android_G, fortunatamente questo *design pattern* permette di ovviare a questa mancanza separando la logica dell'applicazione dalla sua rappresentazione grafica così da rendere l'applicazione espandibile, mantenibile e facilmente testabile. MVP separa l'architettura nei tre livelli Model, View e Presenter ognuno dei quali possiede le proprie responsabilità, e la comunicazione tra le diverse parti è interamente gestita dal Presenter;
- **Contesto d'utilizzo:** per implementare il pattern MVP verrà utilizzata la *suite* di sviluppo Android Studio e sarà applicato sia all'applicazione per la realizzazione degli sceneggiati sia a quella di configurazione. In entrambe viene applicata la separazione delle tre componenti Model, View e Presenter. Per la creazione di schermate grafiche si fa uso di classi Activity_G. Le azioni che compie l'utente attraverso l'interfaccia grafica vengono gestite tramite Listener_G definiti nell'Activity. Il Presenter funge da *man-in-the-middle* utile per istanziare gli oggetti del Model da esporre all'Activity, con l'obiettivo di diminuire la formazione di errori relativi alla gestione della memoria del dispositivo. Inoltre si evita di fondere la logica dell'applicazione con la sua rappresentazione grafica, migliorando così stabilità e manutenibilità del software.

Di seguito viene presentato un esempio di utilizzo concreto del pattern MVP mediante un diagramma di sequenza. In Figura 38 è possibile vedere il processo di creazione di uno sceneggiato.

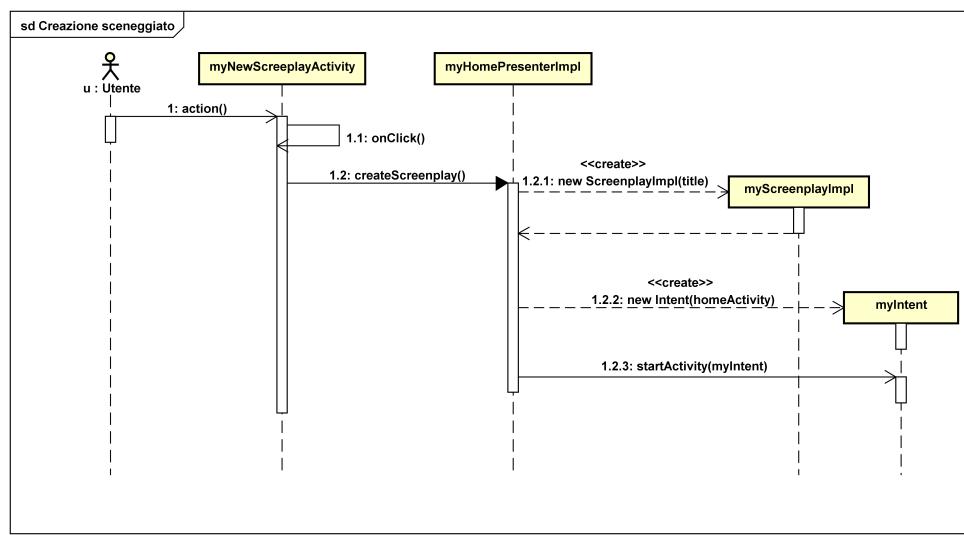


Figura 39: Diagramma di sequenza per la creazione di uno sceneggiato

6.2 Design pattern creazionali

6.2.1 Singleton

- **Scopo dell'utilizzo:** il *Singleton* è un *pattern* utilizzato per le classi che devono avere un'unica istanza, accessibile globalmente, nel corso dell'intera esecuzione del programma;
- **Contesto d'utilizzo:** la classe che segue il *pattern Singleton* è Sivodim::Libraries-::Model::Mivoq::MivoqTTSSingleton; la classe possiede un attributo privato ourInstance corrispondente ad un'istanza della classe stessa, un costruttore privato e un metodo pubblico getInstance() che, se la classe non è stata istanziata la istanzia e la ritorna, altrimenti restituisce l'istanza stessa della classe. Il *pattern Singleton* è stato utilizzato per avere un'unica istanza della classe MivoqTTSSingleton responsabile della gestione della connessione con il server di MIVOQ s.r.l..

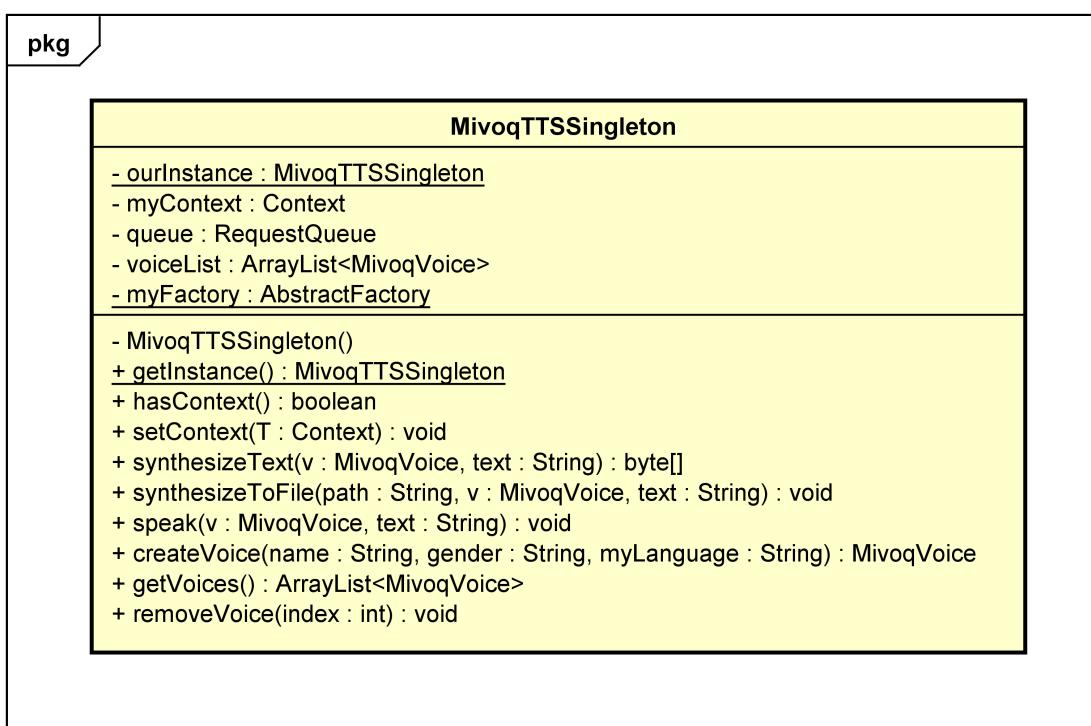


Figura 40: Utilizzo di Singleton

6.2.2 Builder

- **Scopo dell'utilizzo:** il *Builder* è un *pattern* utilizzato per facilitare la creazione di oggetti complessi; il suo scopo è quello di separare la costruzione di un oggetto complesso dalla sua rappresentazione cosicché il processo di costruzione stesso possa creare diverse rappresentazioni. Semplifica la classe da costruire permettendole di focalizzarsi sulle sue funzionalità prendendosi carico della responsabilità della creazione di oggetti tramite una classe separata;
- **Contesto d'utilizzo:** il pattern *Builder* viene utilizzato all'interno della componente Sivodim::Drama::Model dalle seguenti classi per modularizzare la costruzione dell'oggetto e ridurre l'effetto *telescoping*:

- Sivodim::Drama::Model::Chapter::ChapterBuilderImpl: estensione dell'interfaccia ChapterBuilder consente di costruire un oggetto di tipo ChapterImpl corrispondente ad un capitolo dello sceneggiato; i suoi metodi consentono di costruire un oggetto settandone uno o più dei seguenti campi: sfondo sonoro, sfondo video, titolo ed effetti sonori;
- Sivodim::Drama::Model::Character::CharacterBuilderImpl: estensione dell'interfaccia CharacterBuilder consente di costruire un oggetto di tipo CharacterImpl corrispondente ad un personaggio dello sceneggiato; i suoi metodi consentono di costruire un oggetto settandone uno o più dei seguenti campi: nome, voce e avatar_G associato;
- Sivodim::Drama::Model::Character::SpeechBuilderImpl: estensione dell'interfaccia SpeechBuilder consente di costruire un oggetto di tipo SpeechImpl corrispondente ad una battuta dello sceneggiato; i suoi metodi consentono di costruire un oggetto settandone uno o più dei seguenti campi: testo, personaggio associato, emozione ed effetto sonoro associato.

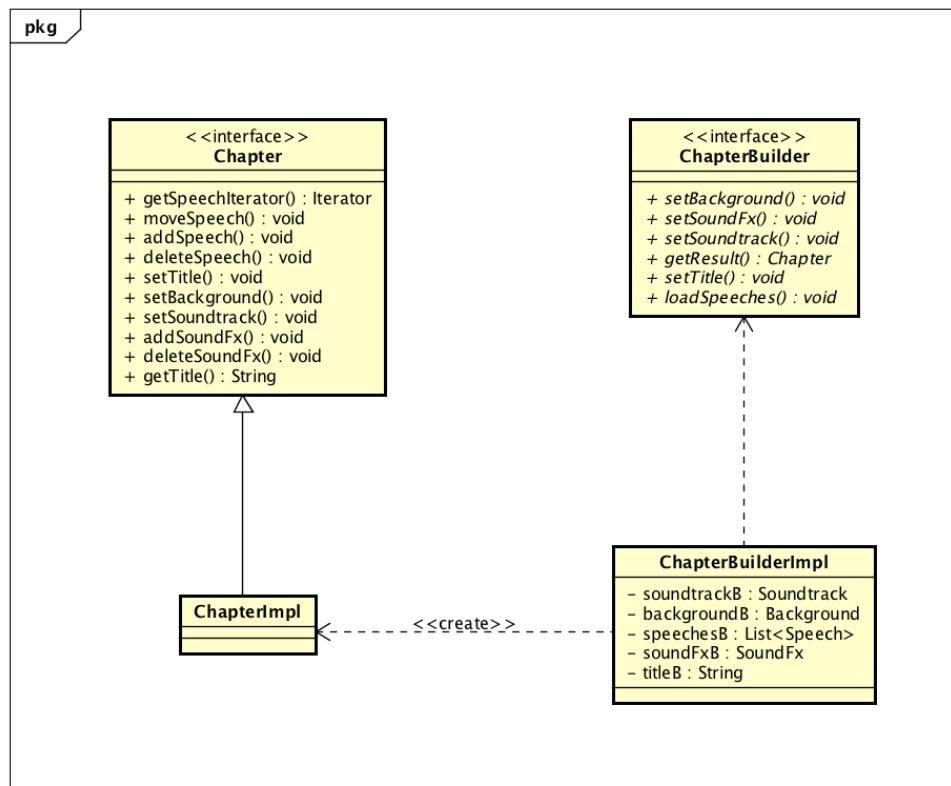


Figura 41: Utilizzo di Builder

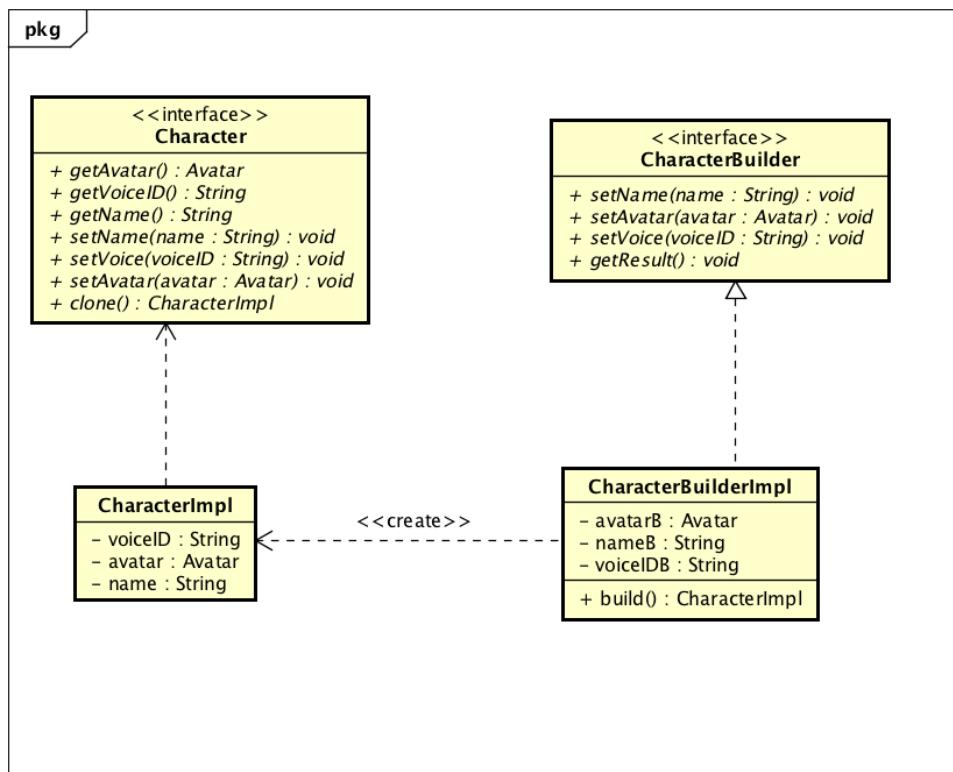


Figura 42: Utilizzo di Builder

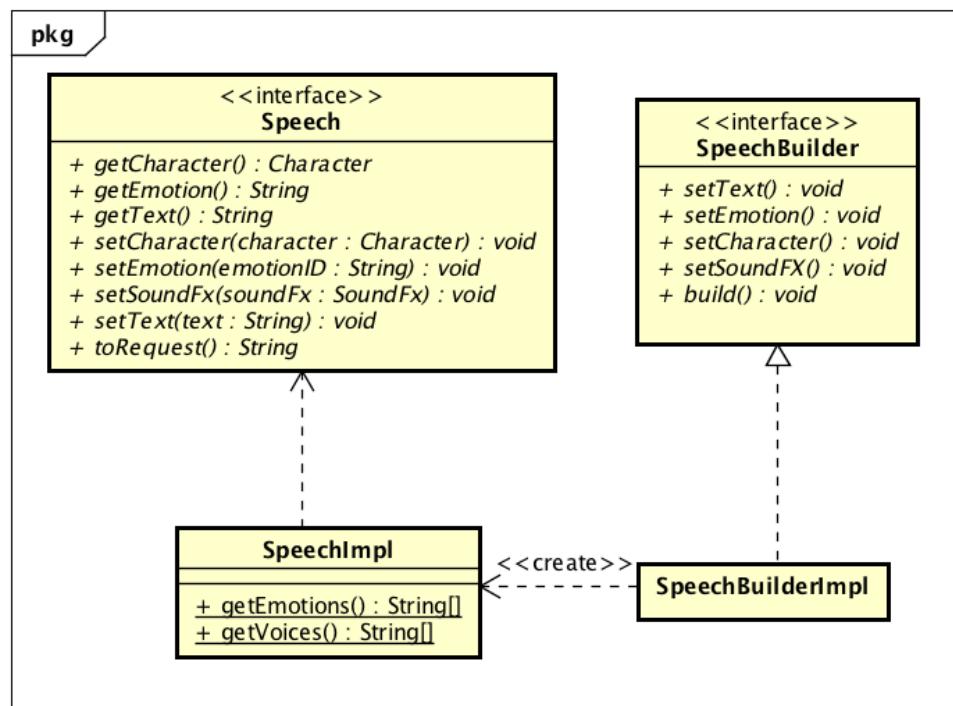


Figura 43: Utilizzo di Builder

6.3 Design pattern strutturali

6.3.1 Facade

- **Scopo dell'utilizzo:** il *Facade* è un *pattern* utilizzato per fornire un'interfaccia unica e semplice per un sottosistema complesso; il suo utilizzo aumenta la manutenibilità, il riuso e la testabilità del codice e diminuisce le dipendenze tra sottoinsiemi;
- **Contesto d'utilizzo:** l'interfaccia che esprime il *pattern Facade* è Sivodim::Libraries::Model::EngineManager::Engine per poter nascondere i dettagli implementativi della gestione di una sintesi vocale.

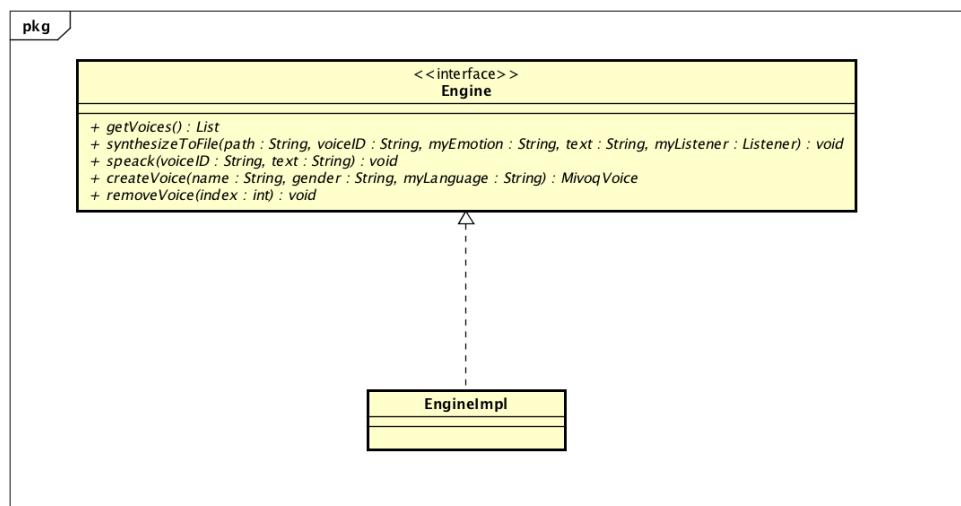


Figura 44: Utilizzo di Facade

6.4 Design pattern comportamentali

6.4.1 Strategy

- **Scopo dell'utilizzo:** lo *Strategy* è un *pattern* utilizzato per definire una famiglia di algoritmi rendendoli interscambiabili e indipendenti dal *client*; è molto utile nelle situazioni dove sia necessario modificare dinamicamente gli algoritmi utilizzati da un'applicazione; la famiglia di algoritmi che implementa una funzionalità esporta sempre la stessa interfaccia, in questo modo il *client* non si deve preoccupare di quale sia effettivamente la strategia che verrà istanziata in un dato momento su un particolare oggetto;
- **Contesto d'utilizzo:** la classi che seguono il *pattern Strategy* è Sivodim::Drama::Model::Screenplay::Export per differenziare gli algoritmi di esportazione fra i vari formati; questo permette all'utente di esportare uno sceneggiato o in formato video o in formato audio.

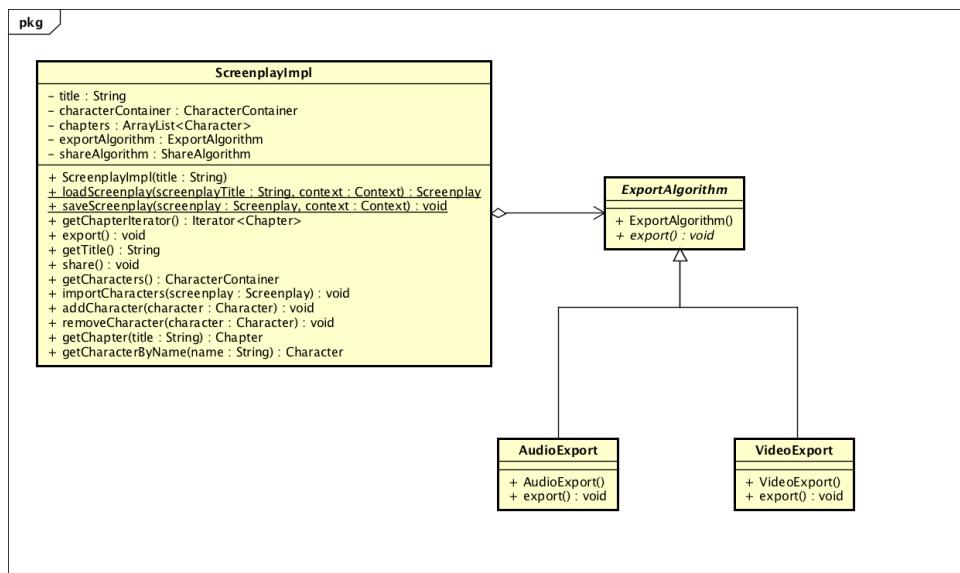


Figura 45: Utilizzo di Strategy

7 Stime di fattibilità e di bisogno di risorse

L'architettura definita per le applicazioni e il modulo *software* presentano un livello di dettaglio sufficiente per fornire una stima sulla fattibilità e sul bisogno di risorse. Attraverso l'analisi dell'architettura si è potuto constatare che le tecnologie scelte risultano adeguate per lo sviluppo del progetto **SiVoDiM**. Le esigenze progettuali vengono pertanto soddisfatte. Android Studio è un ambiente di sviluppo nuovo ai membri del team, ma familiare vista la sua vicinanza ad altri IDE noti come Qt_G e l'integrazione dell'*editor* IntelliJ. Uno studio più approfondito dovrà essere dedicato all'implementazione delle librerie per l'esportazione audio e video dei progetti creati attraverso l'applicazione per gli sceneggiati.

Tramite l'estensione e il riutilizzo delle classi TextToSpeech fornite nativamente dal sistema operativo Android, è possibile implementare un motore di sintesi personalizzato, che nel caso del progetto SiVoDiM andrà a rendere disponibile FA-TTS_G come opzione di sintesi per il sistema operativo. Malgrado non esistano alternative valide per realizzare tale *feature* in modo più semplice, il soddisfacimento di tale requisito potrà risultare ostico visto il funzionamento in remoto del motore FA-TTS e il poco tempo a disposizione nel periodo di progettazione e codifica.

Riassumendo, grazie all'utilizzo di Android Studio, Java e delle librerie FFmpeg e Volley, è possibile garantire la realizzazione dei dettagli architettonici descritti nel corso di tale documento per le applicazioni da distribuire su piattaforma Android_G incluse nel progetto **SiVoDiM**.

8 Tracciamento

In questa sezione vengono riportate le tabelle di tracciamento tra componenti e requisiti. Per maggiore chiarezza, i requisiti associati ai figli di un $package_G$ non vengono trascritti nel padre. I $package$ relativi al *presenter* non hanno requisiti tracciati. Il *presenter* funge infatti da *man-in-the-middle_G* per lo scambio di informazioni fra *model* (dati) e *view* (grafica).

8.1 Tracciamento Componenti - Requisiti

Componenti	Requisiti
Sivodim::Libraries::Model	R0F2 R0F5 R0F7 R0F9.3 R0F9.1 R0F9.2 R0F5.1 R0F9.5.1 R0F9.1.1 R0F5.2 R0F2.1.5 R0F2.1.3 R0F2.1.4 R0F2.1.2 R0F2.1.6 R0F2.5.4 R0F2.5.3 R0F2.5.5 R0F2.5.1 R0F9.5.5 R0F9.5.7 R0F8.11.5.4 R0F8.11.5.2.5 R0F8.11.5.2.3 R0F8.11.5.2.2 R0F9.5.3 R0F14 R2F15 R0F16
Sivodim::Libraries::Model::Mivoq	R0F2.1 R0F9 R0F9.5 R0F9.4 R0F2.1.1 R0F2.5.2

Sivodim::Libraries::Model::Voice	R0F2.4 R0F2.3 R0F2.2 R0F2.1 R0F2.5 R0F9 R0F9.5 R0F9.4 R0F9.5.2 R0F9.5.4 R0F9.5.6 R0F2.1.1
Sivodim::Libraries::Presenter	R2F13

Sivodim::Libraries::View	R0F2 R0F5 R0F2.4 R0F2.1 R0F2.5 R0F9 R0F9.5 R0F9.3 R0F9.4 R0F9.1 R0F9.2 R0F5.1 R0F9.5.1 R0F9.5.2 R0F9.1.1 R0F5.2 R0F2.1.5 R0F2.1.3 R0F2.1.4 R0F2.1.2 R0F2.1.6 R0F2.5.4 R0F2.5.3 R0F2.5.5 R0F2.5.1 R0F9.5.4 R0F9.5.5 R0F9.5.6 R0F9.5.7 R0F8.11.5.4 R0F8.11.5.2.5 R0F8.11.5.2.3 R0F8.11.5.2.2 R0F9.5.3 R0F2.1.1 R2F13 R0F14
Sivodim::Drama::Model	R0F8.3 R0F8.11.2 R0F10 R0F11 R2F8.11.7.1 R0F8.3.2 R1F8.3.1 R0F19

Sivodim::Drama::Presenter	R2F13
---------------------------	-------

Sivodim::Drama::View	R0F1 R0F1.1 R1F8.9 R0F8.1 R0F8.1.1 R0F8.7.4 R1F8.7.5 R1F8.7.1 R0F8.7.7 R0F8.7.7.1 R1F8.7.3 R1F8.7.2 R0F8.7.8 R1F8.7.6 R0F8.10 R0F8 R0F8.11 R0F8.7 R1F8.8 R0F8.6 R1F8.12 R0F8.3 R0F8.2 R0F8.4 R0F8.5 R0F8.10.1 R1F8.11.11 R0F8.11.3 R0F8.11.4 R0F8.11.5 R0F8.11.2 R0F8.11.6 R2F8.11.7 R2F8.11.8 R0F8.11.9 R1F8.11.3.2 R1F8.11.3.1 R1F8.11.3.1.2 R1F8.11.3.1.1 R1F8.11.3.3 R1F8.11.3.3.2 R0F8.11.4.1 R0F8.11.4.2 R0F8.11.5.3 R0F8.11.5.1 R0F8.11.5.2 R1F9.6
----------------------	---

Sivodim::Drama::View	R0F8.11.5.2.6 R0F8.11.5.2.1 R0F8.11.5.2.4 R0F8.11.4.1.1 R0F12 R2F13 R0F8.3.2 R0F8.12.2 R0F14
Sivodim::Drama::Model::Utilities	R1F8.7.1 R1F8.7.3 R1F8.7.2 R0F8.11.3 R2F8.11.7 R1F8.11.3.2 R1F8.11.3.1 R1F8.11.3.1.2 R1F8.11.3.1.1 R1F8.11.3.3 R1F8.11.3.3.2 R1F8.11.3.3.1

Sivodim::Drama::Model::Chapter	R0F8.7.7.1 R0F8.10 R0F8.11 R0F8.10.1 R0F8.10.1.1 R0F8.10.1.2 R1F8.11.11 R0F8.11.3 R0F8.11.4 R0F8.11.5 R0F8.11.2 R2F8.11.7 R2F8.11.8 R0F8.11.9 R1F8.11.3.2 R1F8.11.3.1 R1F8.11.3.1.2 R1F8.11.3.1.1 R1F8.11.3.3 R1F8.11.3.3.2 R0F8.11.4.1 R0F8.11.4.2 R0F8.11.5.3 R0F8.11.5.2 R0F8.11.1 R0F8.11.10 R0F8.11.5.2.6 R0F8.11.5.2.1 R0F8.11.5.2.4 R0F18
--------------------------------	---

Sivodim::Drama::Model::Character	R0F8.1 R0F8.1.1 R0F8.7.4 R1F8.7.5 R1F8.7.1 R0F8.7.7 R1F8.7.3 R1F8.7.2 R0F8.7.8 R1F8.7.6 R0F8.1.1.1 R0F8.1.1.2 R0F8.7 R1F8.8 R0F8.5 R0F8.11.5.1
Sivodim::Drama::Model::Screenplay	R0F1 R0F1.1 R1F8.9 R0F1.1.1 R0F1.1.2 R0F8.10 R0F8 R0F8.6 R1F8.12 R0F8.2 R0F8.4 R0F8.5 R0F8.11.6 R0F8.6.1 R2F8.6.3 R0F8.6.1.1 R0F8.6.1.2 R2F8.12.3 R1F8.12.1 R2F8.12.3.1 R2F8.12.3.2 R2F8.12.3.3 R1F8.12.1.1 R1F9.6 R0F8.11.4.1.1 R0F8.6.2 R0F12 R0F8.12.2

Tabella 2: Tracciamento Componenti - Requisiti

8.2 Tracciamento Requisiti - Componenti

Requisiti	Descrizione	Componenti
R0F1	L'utente deve poter creare uno sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F1.1	L'utente deve poter assegnare un titolo allo sceneggiato creato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R1F8.9	L'utente deve poter modificare il titolo di uno sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.1	L'utente deve poter creare un personaggio per lo sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F8.1.1	L'utente deve poter assegnare un nome al personaggio creato	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F8.7.4	L'utente deve poter modificare il nome di un personaggio	Sivodim::Drama::View Sivodim::Drama::Model::Character
R1F8.7.5	L'utente deve poter assegnare un <i>avatar_G</i> ad un personaggio creato scegliendo tra le immagini predefinite	Sivodim::Drama::View Sivodim::Drama::Model::Character
R1F8.7.1	L'utente deve poter assegnare un <i>avatar_G</i> ad un personaggio creato scegliendo tra le immagini personali	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Character
R0F8.7.7	L'utente deve poter assegnare una voce ad un personaggio creato	Sivodim::Drama::View Sivodim::Drama::Model::Character

R0F8.7.1	L'utente deve poter visualizzare le voci disponibili associabili ad un personaggio	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R1F8.7.3	L'utente deve poter inserire immagini personali come $avatar_G$ in formato .png	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Character
R1F8.7.2	L'utente deve poter inserire immagini personali come $avatar_G$ in formato .jpg	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Character
R0F8.7.8	L'utente deve poter cambiare la voce associata ad un personaggio con un'altra	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F2	L'utente deve poter creare una nuova voce personalizzata	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F5	L'utente deve poter selezionare una voce per i servizi Text-To-Speech _G del sistema	Sivodim::Libraries::Model Sivodim::Libraries::View
R1F8.7.6	L'utente deve poter cambiare l' $avatar_G$ di un personaggio	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F7	Un'applicazione Android _G esterna deve poter utilizzare per servizi di Text-To-Speech _G la voce selezionata per il sistema	Sivodim::Libraries::Model
R0F1.1.1	Il titolo dello sceneggiato non deve essere vuoto	Sivodim::Drama::Model::Screenplay
R0F1.1.2	Il titolo dello sceneggiato non deve contenere spazi né all'inizio né alla fine	Sivodim::Drama::Model::Screenplay
R0F8.1.1.1	Il nome di un personaggio non deve essere vuoto	Sivodim::Drama::Model::Character

R0F8.1.1.2	Il nome di un personaggio non deve contenere spazi né all'inizio né alla fine	Sivodim::Drama::Model::Character
R0F8.10	L'utente deve poter creare un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Chapter Sivodim::Drama::Model::Screenplay
R0F8	L'utente deve poter operare su uno sceneggiato esistente	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.11	L'utente deve poter operare su un capitolo esistente	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.7	L'utente deve poter operare su un personaggio esistente	Sivodim::Drama::View Sivodim::Drama::Model::Character
R1F8.8	L'utente deve poter cancellare un personaggio esistente	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F8.6	L'utente deve poter esportare uno sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R1F8.12	L'utente deve poter condividere uno sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.3	L'utente deve poter salvare uno sceneggiato	Sivodim::Drama::Model Sivodim::Drama::View
R0F8.2	L'utente deve poter riordinare i capitoli di uno sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.4	L'utente deve poter cancellare un capitolo esistente dello sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.5	L'utente deve poter importare nello sceneggiato i personaggi di un altro	Sivodim::Drama::View Sivodim::Drama::Model::Character Sivodim::Drama::Model::Screenplay

R0F2.4	L'utente deve poter assegnare un nome ad una voce personalizzata	Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F2.3	Il nome di una voce non può essere vuoto	Sivodim::Libraries::Model::Voice
R0F2.2	Il nome di una voce personalizzata non deve contenere spazi né all'inizio né alla fine	Sivodim::Libraries::Model::Voice
R0F2.1	L'utente deve poter selezionare la lingua della voce	Sivodim::Libraries::Model::Mivoq Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F2.5	L'utente deve poter selezionare il sesso della voce	Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9	L'utente deve poter operare su una voce esistente	Sivodim::Libraries::Model::Mivoq Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.5	L'utente deve poter impostare gli effetti vocali di una voce esistente	Sivodim::Libraries::Model::Mivoq Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.3	L'utente deve poter modificare la lingua relativa ad una voce esistente	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.4	L'utente deve poter cambiare il sesso di una voce esistente	Sivodim::Libraries::Model::Mivoq Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.1	L'utente deve poter ascoltare un'anteprima della voce su un testo predefinito	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.2	L'utente deve poter cancellare una voce personalizzata esistente	Sivodim::Libraries::Model Sivodim::Libraries::View

R0F5.1	L'utente deve poter selezionare la voce per il dispositivo dalle voci disponibili	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F8.10.1	L'utente deve poter assegnare un titolo ad un capitolo dello sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.10.1.1	Il titolo di un capitolo non deve essere vuoto	Sivodim::Drama::Model::Chapter
R0F8.10.1.2	Il titolo di un capitolo non può contenere spazi vuoti né all'inizio né alla fine	Sivodim::Drama::Model::Chapter
R1F8.11.11	L'utente deve poter modificare il titolo di un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.3	L'utente deve poter operare sullo sfondo (audio e video) di un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R0F8.11.4	L'utente deve poter creare una battuta all'interno di un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.5	L'utente deve poter operare su una battuta del capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.2	L'utente deve poter ascoltare un'anteprima audio di una battuta	Sivodim::Drama::Model Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.6	L'utente deve poter modificare l'ordine delle battute di un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R2F8.11.7	L'utente deve poter inserire un suono tra le battute	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R2F8.11.8	L'utente deve poter cancellare un suono inserito tra le battute	Sivodim::Drama::View Sivodim::Drama::Model::Chapter

R0F8.11.9	L'utente deve poter cancellare una battuta esistente	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R1F8.11.3.2	L'utente deve poter selezionare un'immagine di default come sfondo	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.1	L'utente deve poter caricare un'immagine personale come sfondo di un capitolo	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.1.2	L'utente deve poter caricare un'immagine personale come sfondo in formato .png _G	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.1.1	L'utente deve poter caricare un'immagine personale come sfondo in formato .jpg _G	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.3	L'utente deve poter inserire un sottofondo sonoro personale	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.3.2	L'utente deve poter caricare un file in formato .wav _G come sottofondo sonoro	Sivodim::Drama::View Sivodim::Drama::Model::Utilities Sivodim::Drama::Model::Chapter
R1F8.11.3.3.1	L'utente deve poter caricare un file in formato .mp3 _G come sottofondo sonoro	Sivodim::Drama::Model::Utilities
R0F8.11.4.1	L'utente deve poter scegliere il personaggio a cui assegnare una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.4.2	L'utente deve poter inserire il testo di una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.5.3	L'utente deve poter modificare il testo di una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter

R0F8.11.5.1	L'utente deve poter modificare il personaggio associato ad una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Character
R0F8.11.5.2	L'utente deve poter scegliere l'emozione da associare ad una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.1	L'utente deve poter avviare l'anteprima di una battuta	Sivodim::Drama::Model::Chapter
R0F8.11.10	L'utente deve poter interrompere l'anteprima avviata di una battuta	Sivodim::Drama::Model::Chapter
R0F8.6.1	L'utente deve poter esportare lo sceneggiato in formato audio	Sivodim::Drama::Model::Screenplay
R2F8.6.3	L'utente deve poter esportare uno sceneggiato in formato video .mp4 _G	Sivodim::Drama::Model::Screenplay
R0F8.6.1.1	L'utente deve poter esportare lo sceneggiato in formato audio .wav _G	Sivodim::Drama::Model::Screenplay
R0F8.6.1.2	L'utente deve poter esportare lo sceneggiato in formato audio .mp3 _G	Sivodim::Drama::Model::Screenplay
R2F8.12.3	L'utente deve poter condividere uno sceneggiato in formato video	Sivodim::Drama::Model::Screenplay
R1F8.12.1	L'utente deve poter condividere uno sceneggiato in formato audio	Sivodim::Drama::Model::Screenplay
R2F8.12.3.1	L'utente deve poter condividere uno sceneggiato in formato video su Facebook	Sivodim::Drama::Model::Screenplay
R2F8.12.3.2	L'utente deve poter condividere uno sceneggiato in formato video su Twitter	Sivodim::Drama::Model::Screenplay

R2F8.12.3.3	L'utente deve poter condividere in formato video uno sceneggiato su applicazioni di messaggistica	Sivodim::Drama::Model::Screenplay
R1F8.12.1.1	L'utente deve poter condividere in formato audio uno sceneggiato su Telegram _G	Sivodim::Drama::Model::Screenplay
R0F10	Il nome del file da salvare non deve essere vuoto	Sivodim::Drama::Model
R0F11	Il nome di un file da salvare non deve contenere spazi né all'inizio né alla fine	Sivodim::Drama::Model
R0F9.5.1	L'utente deve poter visualizzare l'elenco degli effetti con i relativi parametri	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.5.2	L'utente deve poter impostare i valori dei parametri degli effetti	Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.1.1	L'utente deve poter avviare l'anteprima di una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F5.2	L'utente deve poter visualizzare l'elenco delle voci disponibili nel dispositivo	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.1.5	L'utente deve poter selezionare la lingua inglese per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.1.3	L'utente deve poter selezionare la lingua italiana per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.1.4	L'utente deve poter selezionare la lingua francese per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.1.2	L'utente deve poter selezionare la lingua tedesca per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View

R0F2.1.6	L'utente deve poter selezionare la lingua inglese (americano) per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.5.4	L'utente deve poter selezionare il sesso maschile per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.5.3	L'utente deve poter selezionare il sesso femminile per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.5.5	L'utente deve poter selezionare il sesso neutro per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F2.5.1	L'utente deve poter selezionare il sesso indefinito per una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.5.4	L'utente deve poter modificare il parametro altezza di una voce	Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.5.5	L'utente deve poter modificare il parametro profondità di una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.5.6	L'utente deve poter modificare il parametro accento di una voce	Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F9.5.7	L'utente deve poter modificare il parametro velocità di una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R1F9.6	L'utente deve poter condividere in formato audio uno sceneggiato su Whatsapp _G	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.11.5.4	L'utente deve poter visualizzare le emozioni disponibili da poter associare ad una battuta	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F8.11.5.2.6	L'utente deve poter selezionare l'emozione felicità per una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter

R0F8.11.5.2.1	L'utente deve poter selezionare l'emozione tristezza per una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.5.2.5	L'utente deve poter selezionare l'emozione rabbia per una battuta	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F8.11.5.2.4	L'utente deve poter scegliere l'emozione disgusto per una battuta	Sivodim::Drama::View Sivodim::Drama::Model::Chapter
R0F8.11.5.2.3	L'utente deve poter scegliere l'emozione paura per una battuta	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F8.11.5.2.2	L'utente deve poter selezionare l'emozione sorpresa per una battuta	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F9.5.3	L'utente deve poter modificare il parametro distorsione di una voce	Sivodim::Libraries::Model Sivodim::Libraries::View
R0F8.11.4.1.1	L'utente deve poter visualizzare l'elenco dei personaggi dello sceneggiato	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R0F8.6.2	Quando uno sceneggiato viene esportato il nome del file sarà il nome dello sceneggiato seguito dal formato di esportazione scelto	Sivodim::Drama::Model::Screenplay
R0F12	L'utente deve poter visualizzare l'elenco degli sceneggiati salvati nel dispositivo all'avvio dell'applicazione	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R2F8.11.7.1	I suoni che l'utente deve poter inserire tra le battute sono esclusivamente forniti dall'applicazione (non possono essere caricati nuovi suoni dall'utente)	Sivodim::Drama::Model

R0F2.1.1	L'applicazione deve poter reperire dal server di MI-VOQ s.r.l. le lingue disponibili per la creazione di una voce personalizzata	Sivodim::Libraries::Model::Mivoq Sivodim::Libraries::Model::Voice Sivodim::Libraries::View
R0F2.5.2	L'applicazione deve poter reperire dal server di MI-VOQ s.r.l. i sessi disponibili per la creazione di una voce personalizzata	Sivodim::Libraries::Model::Mivoq
R2F13	Devono essere offerti meccanismi di annullamento delle operazioni svolte dall'utente	Sivodim::Libraries::Presenter Sivodim::Libraries::View Sivodim::Drama::Presenter Sivodim::Drama::View
R0F8.3.2	L'utente deve poter visualizzare un messaggio di errore nel caso in cui il salvataggio non andasse a buon fine	Sivodim::Drama::Model Sivodim::Drama::View
R0F8.12.2	L'applicazione deve poter visualizzare un messaggio di errore nel caso in cui la condivisione non andasse a buon fine	Sivodim::Drama::View Sivodim::Drama::Model::Screenplay
R1F8.3.1	L'applicazione per la creazione di sceneggiati deve effettuare salvataggi automatici ogni minuto	Sivodim::Drama::Model
R0F14	L'applicazione deve poter visualizzare un messaggio di avviso i caso di mancanza di connessione Internet	Sivodim::Libraries::Model Sivodim::Libraries::View Sivodim::Drama::View
R2F15	In caso di assenza di connessione l'applicazione deve poter funzionare con il motore di sintesi vocale interno al dispositivo	Sivodim::Libraries::Model

R0F16	Il modulo di sistema deve essere in grado di mantenere in memoria gli audio ricevuti dal server di MIVOQ s.r.l.	Sivodim::Libraries::Model
R0F18	Ogni battuta dovrà terminare con un punto (se non scritto dall'utente deve essere aggiunto dall'applicazione)	Sivodim::Drama::Model::Chapter
R0F19	L'applicazione deve salvare i <i>file</i> in una cartella di <i>default</i>	Sivodim::Drama::Model

Tabella 3: Tracciamento Requisiti - Componenti

A Descrizione design pattern

A.1 Design pattern architetturali

A.1.1 Model-View-Presenter

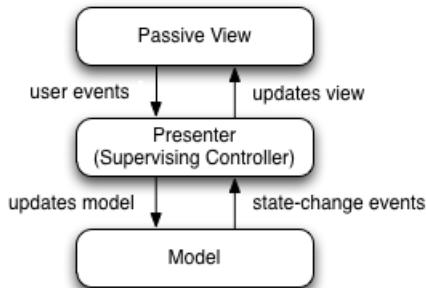


Figura 46: Struttura del design pattern MVP

Model-View-Presenter (MVP) è un *pattern* architettonale derivato dal MVC (Model-View-Controller), utilizzato per dividere il codice in funzionalità distinte. Viene utilizzato quando nelle applicazioni un insieme di informazioni deve essere rappresentato mediante un'interfaccia grafica. MVC è basato sul principio di disaccoppiamento di tre oggetti distinti, riducendo in questo modo le dipendenze reciproche. Inoltre permette di fornire una maggiore modularità, manutenibilità e robustezza al software. Le tre componenti che lo caratterizzano sono le seguenti:

- **Model:** Il *model* rappresenta il cuore dell'applicazione: esso definisce il modello dei dati definendo gli oggetti secondo la logica di utilizzo dell'applicazione, ossia la sua *business logic_G*. Inoltre, indica le possibili operazioni che si possono effettuare sui dati.
- **View:** Nel pattern MVP, la *view* è una componente prevalentemente passiva, ma si occupa anche di notificare al *presenter* eventuali modifiche del proprio stato. Nella struttura del pattern MVP, la *view* si occupa di prendere gli input dell'utente e passarli al *presenter*, affinché esso possa eseguire operazioni sul *model*.
- **Presenter:** Il *presenter* funge da intermediario tra il *model* e la *view*. Si occupa di implementare l'insieme di operazioni eseguibili sul modello dei dati attraverso una particolare vista, ossia l'*application logic*.

A.2 Design pattern creazionali

A.2.1 Builder

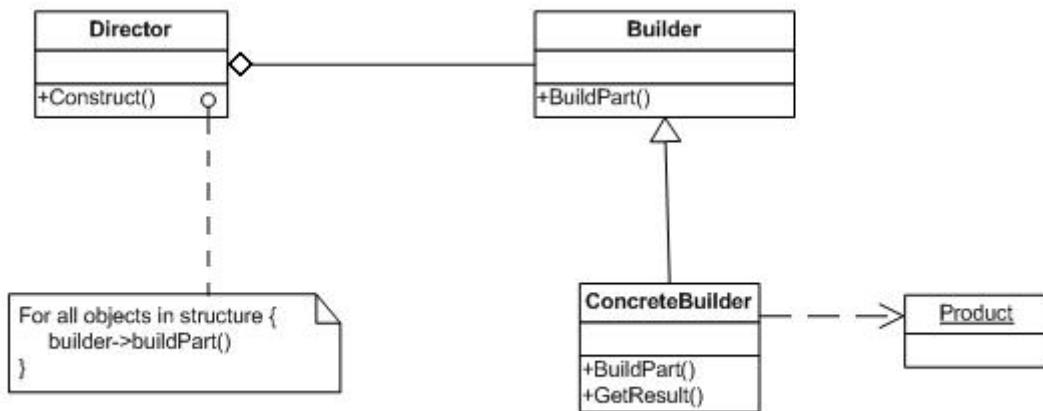


Figura 47: Struttura del design pattern Builder

Builder è un *design pattern_G* che ha lo scopo di separare la costruzione di un oggetto complesso dalla sua rappresentazione, cosicché il processo di costruzione stesso possa creare diverse rappresentazioni.

L'utilizzo di una classe *builder* separata permette di focalizzarsi sulla corretta costruzione di un'istanza, lasciando che la classe originale si concentri sul funzionamento degli oggetti.

Le parti che compongono il design pattern Builder sono le seguenti:

- **Builder:** specifica l'interfaccia astratta che crea le parti dell'oggetto Product;
- **ConcreteBuilder:** costruisce e assembla le parti del prodotto implementando l'interfaccia Builder, tiene traccia dell'istanza costruita e fornisce un'interfaccia per recuperare il prodotto;
- **Director:** costruisce un oggetto utilizzando l'interfaccia Builder;
- **Product:** rappresenta l'oggetto complesso da costruire, ovvero il prodotto.

A.2.2 Abstract Factory

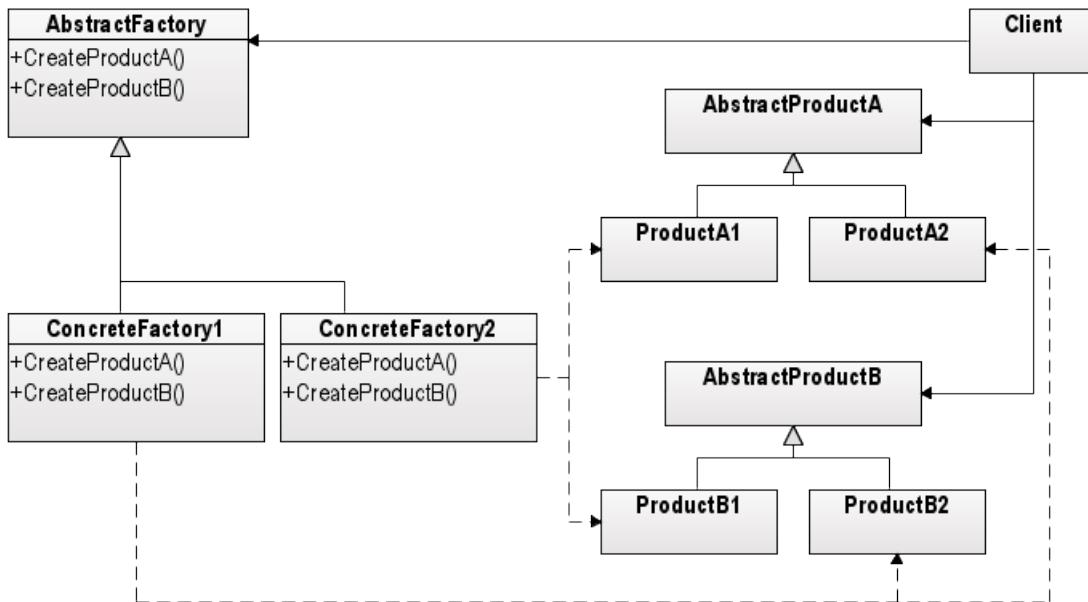


Figura 48: Struttura del design pattern Abstract Factory

L'Abstract Factory fornisce un'interfaccia per creare famiglie di oggetti connessi o dipendenti tra loro, in modo che non ci sia necessità da parte dei *client* di specificare i nomi delle classi concrete all'interno del proprio codice.

In questo modo un sistema può essere indipendente dall'implementazione degli oggetti concreti e si fa sì che il *client*, attraverso l'interfaccia, utilizzi diverse famiglie di prodotti. Il pattern Abstract Factory è composto dalle seguenti parti:

- **AbstractFactory**: dichiara l'interfaccia per le operazioni che creano prodotti astratti;
- **ConcreteFactory**: implementa le operazioni per creare i prodotti concreti;
- **AbstractProduct**: dichiara l'interfaccia per un particolare prodotto;
- **ConcreteProduct**: implementa l'interfaccia AbstractProduct e definisce l'oggetto prodotto che deve essere creato dalla factory concreta corrispondente;
- **Client**: utilizza solo le interfacce dichiarate da AbstractFactory e AbstractProduct.

A.2.3 Singleton

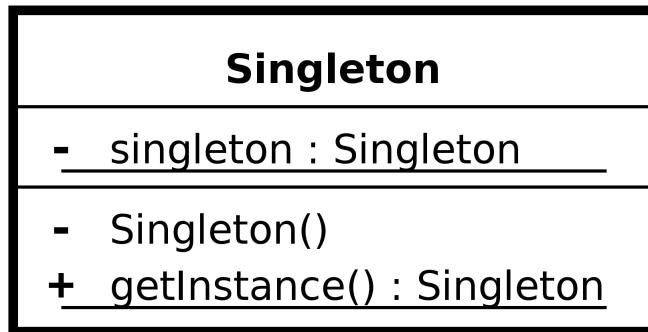


Figura 49: Struttura del design pattern Singleton

Viene usato il *pattern* Singleton per le classi che devono avere un'unica istanza durante l'intera esecuzione dell'applicazione. Il *pattern* Singleton consente di fornire un punto di accesso globale a tale istanza ed è applicabile ogni volta in cui debba esistere una sola istanza di una determinata classe in tutta l'applicazione, prestando attenzione al fatto che l'istanza sia estendibile tramite ereditarietà.

Il *pattern* Singleton può essere utilizzato nei casi seguenti:

- Quando deve esistere una sola istanza di una data classe resa accessibile ai *client* attraverso un punto di accesso noto a tutti gli utilizzatori;
- Quando l'unica istanza deve poter essere estesa attraverso la definizione di sottoclassi e i *client* devono essere in grado di utilizzarle senza necessità di modificare il proprio codice.

A.3 Design pattern strutturali

A.3.1 Adapter

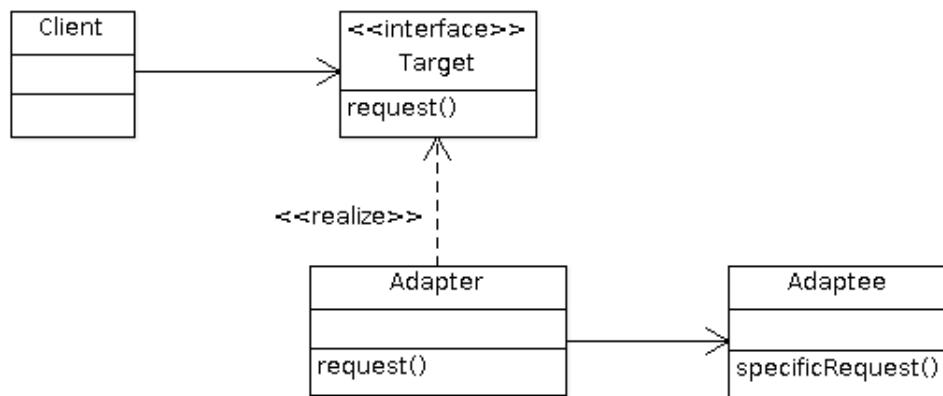


Figura 50: Struttura del design pattern Adapter

Il *design pattern* Adapter ha lo scopo di convertire l’interfaccia di una classe in un’altra interfaccia richiesta dal *client*. Il pattern Adapter consente a classi diverse di operare insieme quando ciò non sarebbe altrimenti possibile a causa di interfacce incompatibili.

I possibili casi d’utilizzo sono i seguenti:

- Quando si vuole usare una classe esistente, ma la sua interfaccia non è compatibile con quella desiderata;
- Quando si vuole creare una classe riusabile in grado di cooperare con classi non correlate o impreviste, cioè con classi che non necessariamente hanno interfacce compatibili;
- Per gli oggetti Adapter quando si devono utilizzare diverse sottoclassi esistenti, ma non è pratico adattare la loro interfaccia creando una sottoclasse per ciascuna di esse.

A.3.2 Facade

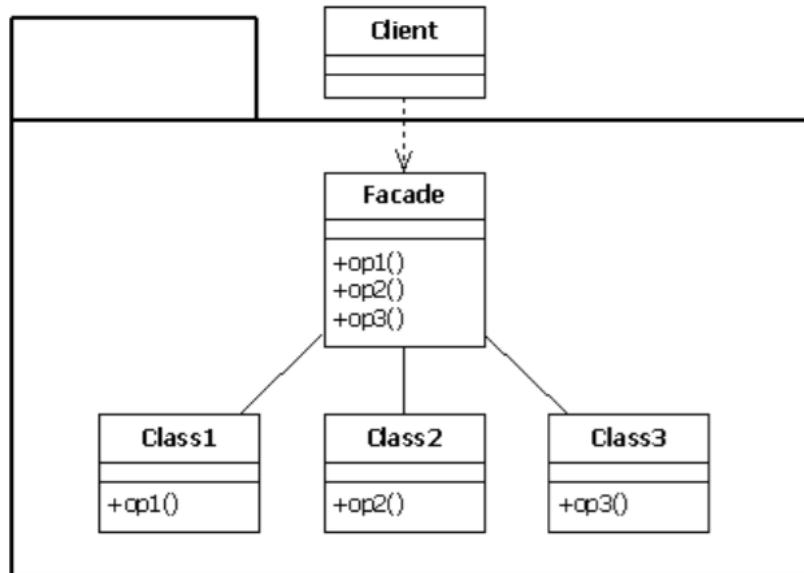


Figura 51: Struttura del design pattern Facade

Il *design pattern_G* Facade ha lo scopo di fornire un'unica interfaccia semplice per un sottosistema complesso, definendo un'interfaccia di livello più alto che rende il sottosistema più semplice da utilizzare.

Suddividendo un sistema in sottosistemi, si riesce a ridurne la complessità e si minimizzano le comunicazioni e le dipendenze fra i diversi sottosistemi.

Il pattern Facade si può utilizzare nei seguenti casi:

- Quando si vuole fornire un'interfaccia semplice a un sottosistema complesso poiché si fornisce un'interfaccia per l'interazione ad alto livello con la maggior parte dei client;
- Nei casi in cui ci siano molte dipendenze fra i *client* e le classi che implementano un'astrazione, dato che si disaccoppia il sottosistema dai *client* e dagli altri sistemi, promuovendo portabilità e indipendenza delle parti sottostanti del sistema;
- Quando si vogliono organizzare i sottosistemi in una struttura a livelli.

A.4 Design pattern comportamentali

A.4.1 Strategy

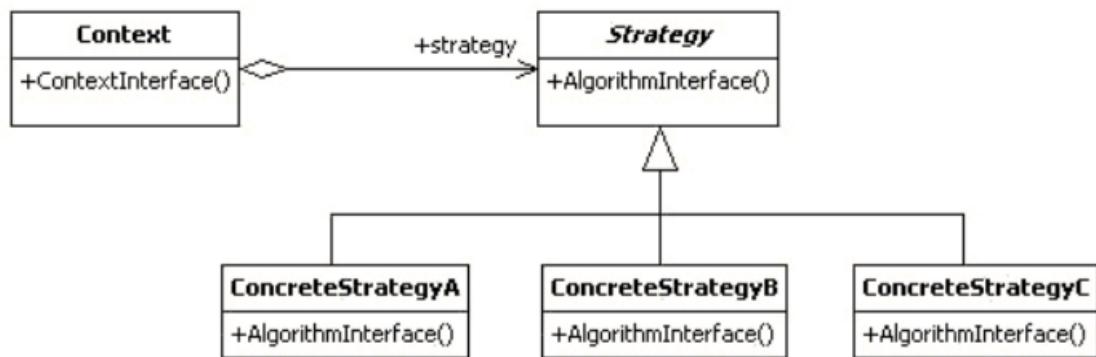


Figura 52: Struttura del design pattern Strategy

L'obiettivo del *design pattern_G* Strategy è di definire una famiglia di algoritmi, incapsularli e renderli interscambiabili. Questo permette agli algoritmi di variare indipendentemente dai *client* che ne fanno uso. Il *pattern* Strategy è utile in quelle situazioni dove sia necessario modificare dinamicamente gli algoritmi utilizzati da un'applicazione.

I possibili motivi di utilizzo sono i seguenti:

- Sono necessarie più varianti di un algoritmo;
- Molte classi correlate differiscono fra loro solo per il comportamento;
- Un algoritmo usa una struttura dati che non dovrebbe essere resa nota ai *client*;
- Una classe definisce molti comportamenti che compaiono all'interno di scelte condizionali multiple.