



RASD - Sistema voto e scrutinio elettronico

Mauro Tellaroli - 922738 Marco Pasini - 922742

Settembre 2022

Indice

0 Prefazione	2
1 Descrizione del Problema	2
1.1 Analisi e specifica dei requisiti	2
1.1.1 Requisiti funzionali	3
1.1.2 Requisiti non funzionali	4
1.2 Glossario	5
2 Progettazione del Sistema	6
2.1 Diagramma dei casi d'uso	6
2.2 Descrizione degli scenari	8
2.3 Diagramma delle classi di progetto	11
2.3.1 Prima versione	12
2.3.2 Versione definitiva	13
2.4 Diagrammi di sequenza	14
2.4.1 Accesso	14
2.4.2 Creazione della sessione	15
2.4.3 Votazione ordinale	16
2.4.4 Votazione categorica	16
2.4.5 Votazione referendum	17
2.5 Diagrammi delle attività	18
2.5.1 Accesso	18
2.5.2 Votazione	18
2.5.3 Creazione della sessione	18
2.6 Diagramma dei componenti	19
3 Implementazione del Sistema	20
3.1 Design Pattern	20
3.2 Gestione dei dati persistenti	20
3.2.1 Schema logico	22
3.3 Descrizione dell'interfaccia grafica	22
3.3.1 Login	23
3.3.2 Finestra Elettore	24
3.3.3 Finestra Gestore	26
3.4 Diagramma di Deployment	29
3.5 Specifica dei vincoli	29
3.5.1 Sessione	29
3.5.2 Persona	30
3.5.3 Votazione	30
3.6 Testing	30
3.7 Note per l'installazione e l'utilizzo	31

0 Prefazione

Il seguente progetto è stato realizzato dagli studenti:

- Mauro Tellaroli con matricola 922738
- Marco Pasini con matricola 922742

Durante le lezioni sono stati consegnati, con valutazione positiva, tutti gli assignment proposti dalla docente. Rispetto agli assignment consegnati precedentemente sono stati modificati:

- Diagramma dei casi d'uso: sono stati aggiunti diversi casi d'uso;
- Diagramma delle classi: sono state aggiunte tutte le classi legate al controllo dei dati e alla interfaccia grafica, in aggiunta a delle migliorie delle classi principali;
- Diagrammi di sequenza: sono stati modificati conseguentemente alla modifica delle classi

1 Descrizione del Problema

Il progetto riguarda la realizzazione di un sistema di votazione e scrutinio elettronico in sostituzione dell'attuale sistema di votazione cartaceo.

1.1 Analisi e specifica dei requisiti

Questo nuovo sistema ha come scopo quello di portare maggiore sicurezza e permettere una votazione a distanza, tramite un apposito applicativo web, esclusivamente agli *elettori* che sono impossibilitati nel recarsi ai *seggi* per motivi convalidati. Gli *elettori* che non presenteranno una valida motivazione per poter effettuare la votazione a distanza, dovranno recarsi ai vari *seggi* che metteranno a disposizione dei *totem*.

L'accesso al sistema da parte di *gestori* e *elettori* avviene tramite credeziali. Si assume che la *pubblica amministrazione* si occupi della distribuzione di tali credenziali.

Le credenziali dei *gestori* non permettono la votazione. Ogni persona che ricopre il ruolo di gestore avrà delle credenziali da *gestore* e da *elettore*.

Le votazioni dovranno avvenire durante una *sessione di voto* che potrà essere aperta da un *gestore*.

Una *sessione* può essere gestita esclusivamente dal *gestore* che l'ha creata.

La creazione della votazione potrà avvenire dopo che il *gestore* inserirà le seguenti informazioni:

- Tipologia di votazione
- Tipologia di scrutinio
- Titolo della votazione
- Data di inizio della sessione
- Data di fine della sessione
- Lista di candidati/partiti/domanda del referendum

Le modalità di voto sono:

Modalità di voto	Descrizione
Voto ordinale	L' <i>elettore</i> dovrà ordinare una lista di candidati a seconda della propria preferenza.
Voto categorico	L' <i>elettore</i> dovrà selezionare da una lista di candidati un unico candidato secondo la sua preferenza.
Voto categorico con preferenze	L' <i>elettore</i> dovrà selezionare da una lista di partiti un unico partito. Avrà inoltre la possibilità di ordinare i candidati del partito scelto in base alle proprie preferenze.
Referendum	L' <i>elettore</i> dovrà dare il consenso o il dissenso al quesito del referendum.

Le modalità di scrutinio sono:

Modalità di scrutinio	Descrizione
Maggioranza	Il vincitore è il candidato che ha ottenuto il maggior numero di voti
Maggioranza assoluta	Il vincitore è il candidato che ha ottenuto la maggioranza assoluta dei voti, cioè il 50%+1 dei voti espressi.
Referendum senza quorum	Si procede al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.
Referendum con quorum	Si procede al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione il 50%+1 degli aventi diritto al voto

I risultati saranno visibili solo dopo la chiusura della *sessione di voto* su richiesta di un *gestore* del sistema.

1.1.1 Requisiti funzionali

Requisiti utente

1. Il sistema deve permettere ai *gestori* del sistema, previo login con apposite credenziali, la selezione delle modalità di voto e di scrutinio;
2. Il sistema deve permettere ai *gestori* del sistema, previo login con apposite credenziali, di chiudere la *sessione di voto* solo se creata dal *gestore* stesso;
3. Il sistema deve fornire, una volta chiusa la *sessione di voto* e su decisione del *gestore* creatore della *sessione*, i vincitori in base alle modalità di scrutinio scelto all'avvio;
4. Il sistema deve permettere agli *elettori* il voto, in base alla modalità di voto scelta precedentemente;
5. Il sistema deve chiedere un'ulteriore conferma del voto espresso dall'*elettore* prima di salvare il voto;
6. Il sistema deve supportare le seguenti modalità di voto: voto ordinale, voto categorico, voto categorico con preferenze e referendum;
7. Il sistema deve supportare le seguenti modalità di scrutinio: maggioranza, maggioranza assoluta, referendum senza quorum e referendum con quorum.

User story

1. In quanto *gestore* del sistema, devo poter selezionare la modalità di voto e di scrutinio;
2. In quanto *gestore* del sistema, devo poter chiudere la sessione di voto creata da me;
3. In quanto *gestore* del sistema, devo poter visualizzare i vincitori di una *sessione* di voto creata da me, solo dopo aver chiuso la sessione di voto;
4. In quanto *elettore*, devo poter votare in base alle modalità impostate.

Requisiti di sistema

1. All'avvio del sistema, un apposita pagina deve permettere sia ai *gestori* del sistema che agli *elettori* il login tramite credenziali;
2. Il sistema deve permettere ai *gestori* del sistema autenticati di creare una sessione di voto selezionando modalità di voto e di scrutinio, titolo della votazione, data di inizio e fine della sessione e la lista dei candidati/partiti/domanda del referendum;
3. Una volta avviata una *sessione di voto* non sarà possibile in alcun modo modificarne le modalità di voto o scrutinio;
4. Una volta che l'*elettore* entra nella scheda di votazione non potrà annullare la votazione;
5. Il sistema fornirà all'*elettore* autenticato l'elenco di tutte le varie *sessioni di voto* in cui ha diritto di voto e la possibilità di votare in quelle *sessioni di voto*;
6. Il sistema dovrà fornire per ogni *sessione di voto* un titolo della votazione;
7. Dopo che l'*elettore* effettua il voto, prima del salvataggio, deve esserci una schermata di riepilogo del voto e di ulteriore conferma;
8. Il sistema sovrà fornire come possibilità di voto, in tutte le modalità di voto, la possibilità di lasciare la *scheda bianca*;
9. Nel caso si debba esprimere nel voto una preferenza doppia, i candidati dovranno essere di genere diverso (uomo e donna), in caso contrario la seconda preferenza non verrà conteggiata;
10. I risultati della votazione potranno essere visualizzati, su richiesta del *gestore* del sistema che ha creato la *sessione*, solo dopo la chiusura della *sessione di voto*.

1.1.2 Requisiti non funzionali

Requisiti di prodotto

1. Il sistema potrà essere raggiungibile anche tramite un *applicativo web* da un qualsiasi dispositivo.

Requisiti esterni

1. Un *elettore* può votare una volta sola in una sessione di voto;
2. Solo gli *elettori* che hanno fornito una valida motivazione alla *pubblica amministrazione* per non potersi recare ai *seggi* potranno effettuare la votazione a distanza;
3. Il sistema non deve influenzare il voto.

Requisiti di sicurezza

1. Non sarà possibile in nessun modo, risalire all'identità di un *elettore*;
2. Il sistema deve fornire un sistema di *auditing*.

1.2 Glossario

Termine	Definizione
Applicativo web	Programma sviluppato per risiedere sulla rete, ed essere accessibile ed utilizzabile da qualsiasi computer connesso alla medesima rete che può essere internet.
Auditing	Sistema di revisione atto a verificare il corretto funzionamento del sistema.
Elettore	Persona che intende esercitare il proprio diritto di voto.
Gestore	Addetto all'apertura e alla chiusura delle sessioni di voto.
Pubblica amministrazione	Insieme degli enti pubblici che concorrono all'esercizio e alle funzioni della gestione, direzione e coordinazione di uno Stato nelle materie di sua competenza.
Scheda bianca	Esercitazione del diritto di astensione dal voto.
Seggio	Luogo dove gli elettori si recano per votare. Ogni comune disporrà di propri seggi.
Sessione di voto	Periodo di tempo in cui sarà possibile votare.
Totem	Dispositivo fisso che permetterà agli elettori di effettuare la votazione.

2 Progettazione del Sistema

2.1 Diagramma dei casi d'uso

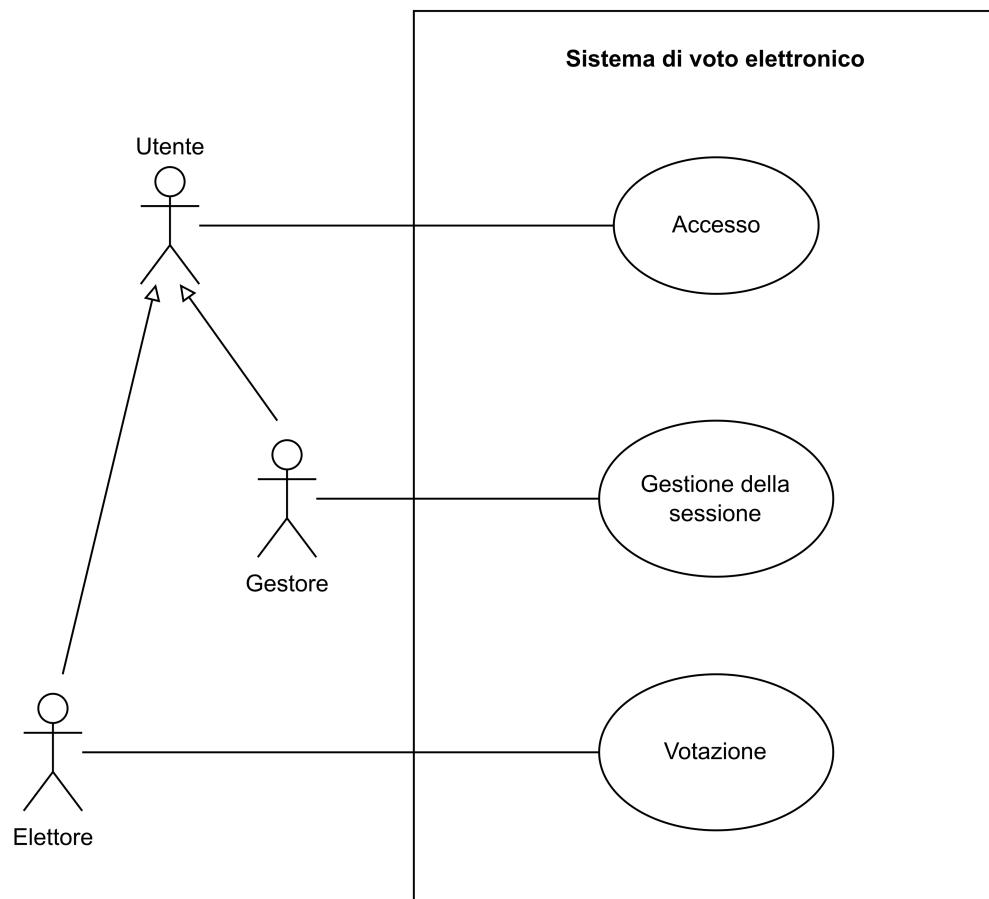


Figura 1: Diagramma dei casi d'uso

Viene riportato in figura 1 il diagramma dei casi d'uso in maniera molto astratta. Di seguito degli approfondimenti dei vari casi d'uso.

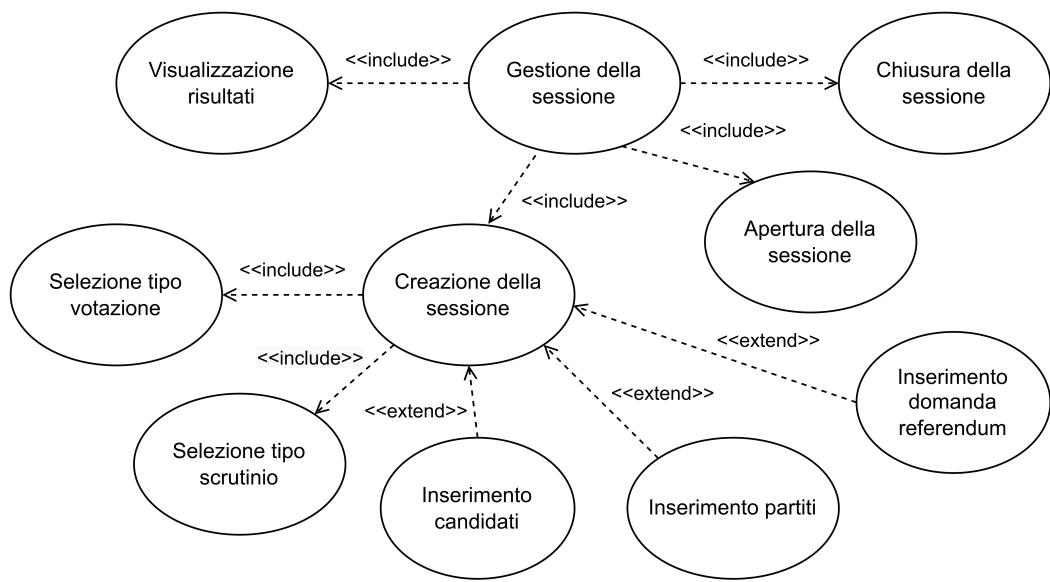


Figura 2: Approfondimento del caso d'uso "Gestione della sessione"

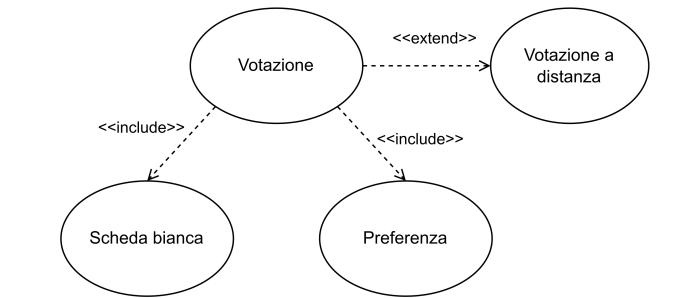


Figura 3: Approfondimento del caso d'uso "Votazione"

• Accesso:

caso d'uso utilizzato dagli utenti (sia elettori che gestori) per poter accedere tramite credenziali al sistema.

• Visualizzazione risultati:

caso d'uso utilizzato dai gestori per visualizzare gli esiti di una sessione di voto.

• Creazione della sessione:

caso d'uso utilizzato dai gestori per creare una sessione di voto in cui gli elettori potranno votare una volta aperta.

• Selezione tipo votazione:

caso d'uso utilizzato dai gestori per selezionare il tipo di votazione della sessione.

• Selezione tipo scrutinio:

caso d'uso utilizzato dai gestori per selezionare il tipo di scrutinio della sessione.

- **Inserimento candidati:**
caso d'uso utilizzato dai gestori per inserire gli eventuali candidati coinvolti in una sessione.
- **Inserimento partiti:**
caso d'uso utilizzato dai gestori per inserire gli eventuali partiti coinvolti in una sessione.
- **Inserimento domanda referendum:**
caso d'uso utilizzato dai gestori per inserire l'eventuale domanda referendaria in una sessione.
- **Apertura della sessione:**
caso d'uso utilizzato dai gestori per aprire una sessione di voto.
- **Chiusura della sessione:**
caso d'uso utilizzato dai gestori per chiudere una sessione di voto, non permettendo agli elettori di votare.
- **Scheda bianca:**
caso d'uso utilizzato dagli elettori per esprimere l'astensione dal voto.
- **Preferenza:**
caso d'uso utilizzato dagli elettori per poter esprimere la propria preferenza alla votazione e quindi votare.
- **Votazione a distanza:**
caso d'uso utilizzato dagli elettori che hanno diritto di voto a distanza per effettuare il voto.

2.2 Descrizione degli scenari

Nome	Accesso
Scopo	Verificare l'identità dell'elettore o del gestore indirizzandolo alla pagina relativa al proprio ruolo
Attore/i	Utente
Pre-condizioni	L'utente deve possedere delle credenziali d'accesso
Trigger	L'utente vuole accedere al sistema
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. L'utente inserisce le proprie credenziali 2. Le credenziali inserite sono corrette 3. Avviene l'accesso alla piattaforma di votazione
Alternativa/e	<ol style="list-style-type: none"> 2'. Le credenziali sono errate 3'. Non avviene l'accesso alla piattaforma di votazione
Post-condizioni	

Nome	Creazione della sessione
Scopo	Permettere a un gestore del sistema di creare una sessione di voto
Attore/i	Gestore
Pre-condizioni	
Trigger	Il gestore vuole creare una nuova sessione di voto
Descrizione sequenza eventi	<ul style="list-style-type: none"> 1. Il gestore vuole creare una nuova sessione di voto 2. Il gestore sceglie la modalità di voto 3. Il gestore sceglie la modalità di scrutinio 4. Il gestore inserisce un titolo della sessione di voto 5. Il gestore inserisce una descrizione della votazione 6. Il gestore inserisce la lista dei candidati e/o partiti o la domanda del referendum 7. Il gestore sceglie una data di inizio della sessione di voto 8. Il gestore sceglie una data di fine della sessione di voto 9. Il gestore conferma le scelte
Alternativa/e	<ul style="list-style-type: none"> 8'. Il gestore non conferma le scelte 9'. Il gestore modifica le scelte 10'. Il gestore conferma le scelte
Post-condizioni	<ul style="list-style-type: none"> 1. La data di inizio della sessione di voto deve essere successiva o uguale alla data in cui il gestore crea la sessione di voto 2. La data di fine della sessione di voto deve essere successiva alla data di inizio

Nome	Visualizzazione risultati
Attore/i	Gestore
Scopo	Visualizzare i risultati della sessione di voto
Pre-condizioni	<ol style="list-style-type: none"> 1. La sessione di voto relativa ai risultati deve essere chiusa 2. La sessione deve essere stata creata dal quello specifico gestore
Trigger	Il gestore desidera visualizzare i risultati della sessione di voto
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. Il gestore richiede i risultati 2. Il sistema comunica i risultati
Alternativa/e	<ol style="list-style-type: none"> 2' Errore di sistema
Post-condizioni	

Nome	Votazione
Attore/i	Elettore
Scopo	Far votare l'elettore
Pre-condizioni	<ol style="list-style-type: none"> 1. L'elettore deve avere il diritto di voto 2. L'elettore non deve aver già votato nella stessa sessione di voto
Trigger	L'elettore vuole votare
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. L'elettore seleziona la sessione di voto in cui votare 2. L'elettore esprime la propria preferenza 3. L'elettore conferma la sua scelta
Alternativa/e	<ol style="list-style-type: none"> 2' L'elettore si astiene dal voto tramite la scheda bianca 3' L'elettore non conferma la sua scelta 4' L'elettore modifica la propria preferenza 5' L'elettore conferma la sua scelta
Post-condizioni	L'elettore non potrà più votare in quella sessione di voto

Nome	Votazione a distanza
Attore/i	Elettore
Scopo	Far votare l'elettore che ne ha diritto a distanza
Pre-condizioni	<ol style="list-style-type: none"> 1. L'elettore deve avere il diritto di voto a distanza 2. L'elettore non deve aver già votato nella stessa sessione di voto
Trigger	L'elettore vuole votare
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. L'elettore seleziona la sessione di voto in cui votare 2. L'elettore esprime la propria preferenza 3. L'elettore conferma la sua scelta
Alternativa/e	<ol style="list-style-type: none"> 2' L'elettore si astiene dal voto tramite la scheda bianca 3' L'elettore non conferma la sua scelta 4' L'elettore modifica la propria preferenza 5' L'elettore conferma la sua scelta
Post-condizioni	L'elettore non potrà più votare in quella sessione di voto

2.3 Diagramma delle classi di progetto

I seguenti diagrammi sono stati realizzati utilizzando [PlantUML](#); i file sorgente sono disponibili nella directory [Diagrammi](#).

2.3.1 Prima versione

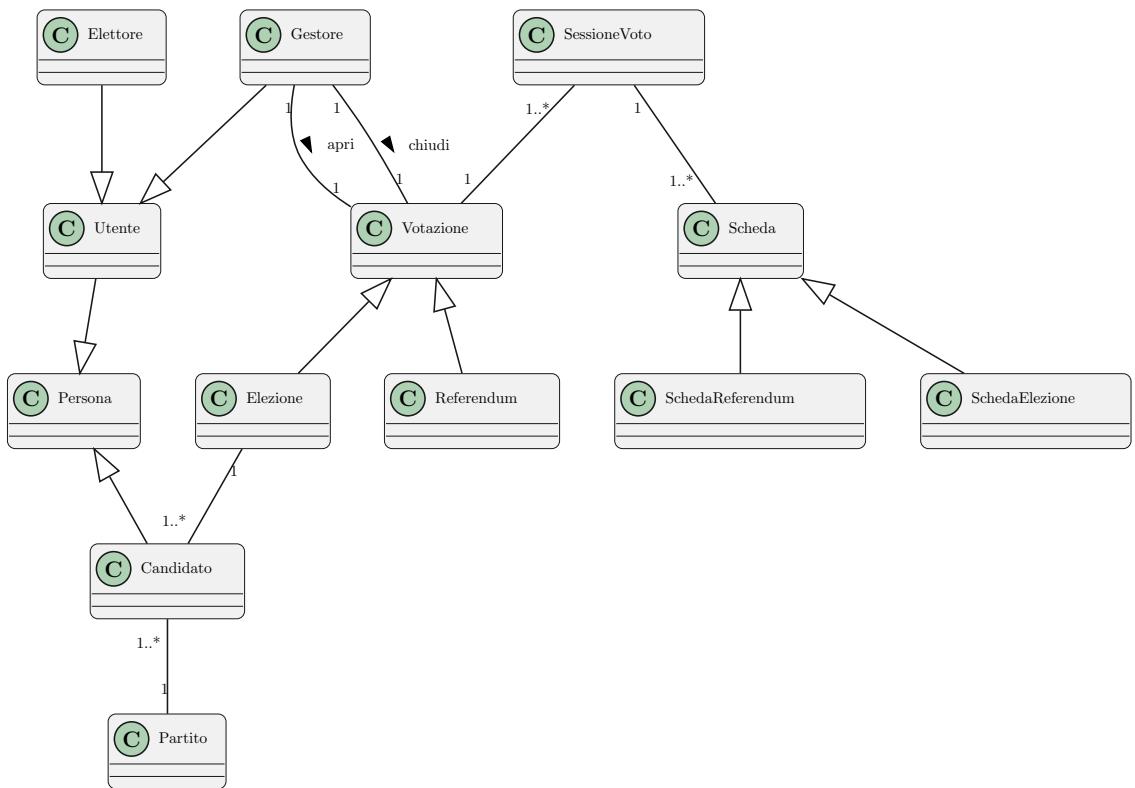


Figura 4: Prima versione del diagramma delle classi

È sopra riproposto il diagramma delle classi iniziale, proposto negli assignment.

2.3.2 Versione definitiva



Figura 5: Versione definitiva del diagramma delle classi

È sopra riportato la versione, poi implementata nel progetto, del diagramma delle classi. Le due classi *PersonaBuilder* e *SessioneBuilder* non sono state completamente rappresentate per poter migliorare la leggibilità del grafico. Una versione che presenta una risoluzione maggiore è presente nella cartella "Diagrammi\Classi \ClassDiagram.svg"

2.4 Diagrammi di sequenza

I seguenti diagrammi sono stati realizzati utilizzando [PlantUML](#); i file sorgente sono disponibili nella directory [Diagrammi](#).

2.4.1 Accesso

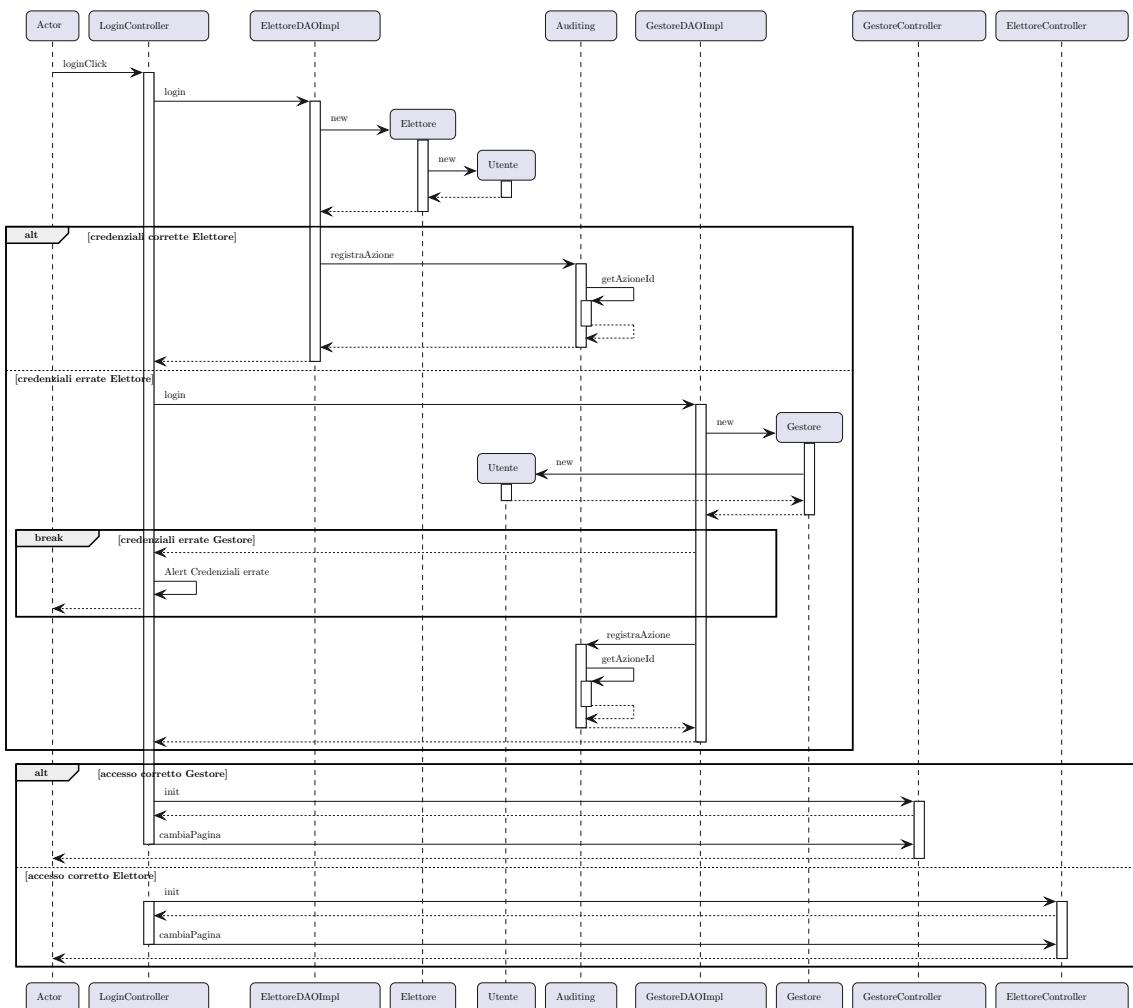


Figura 6: Diagramma di sequenza che mostra i passi per l'accesso al sistema

2.4.2 Creazione della sessione

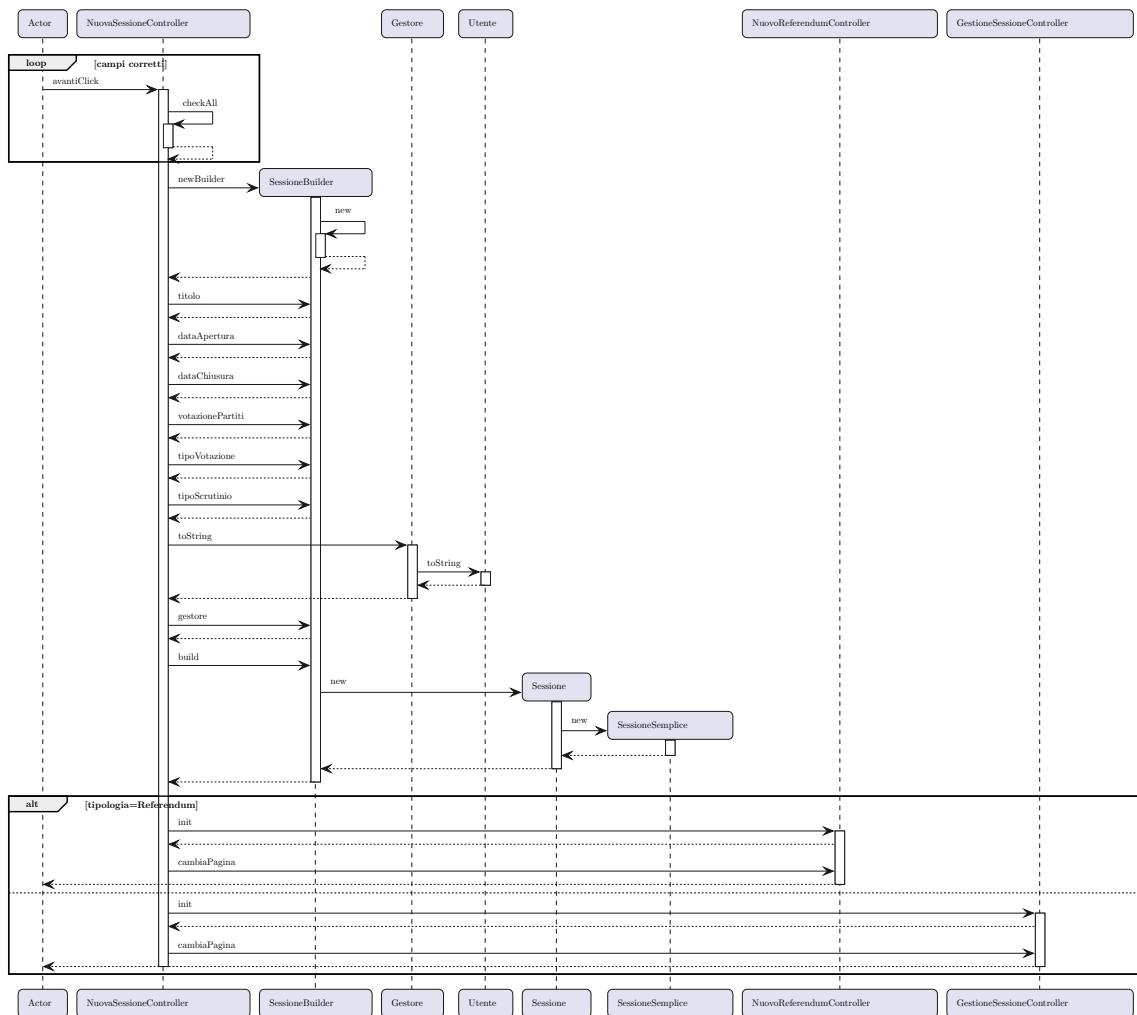


Figura 7: Diagramma di sequenza che mostra i passi per la creazione di una sessione

2.4.3 Votazione ordinale

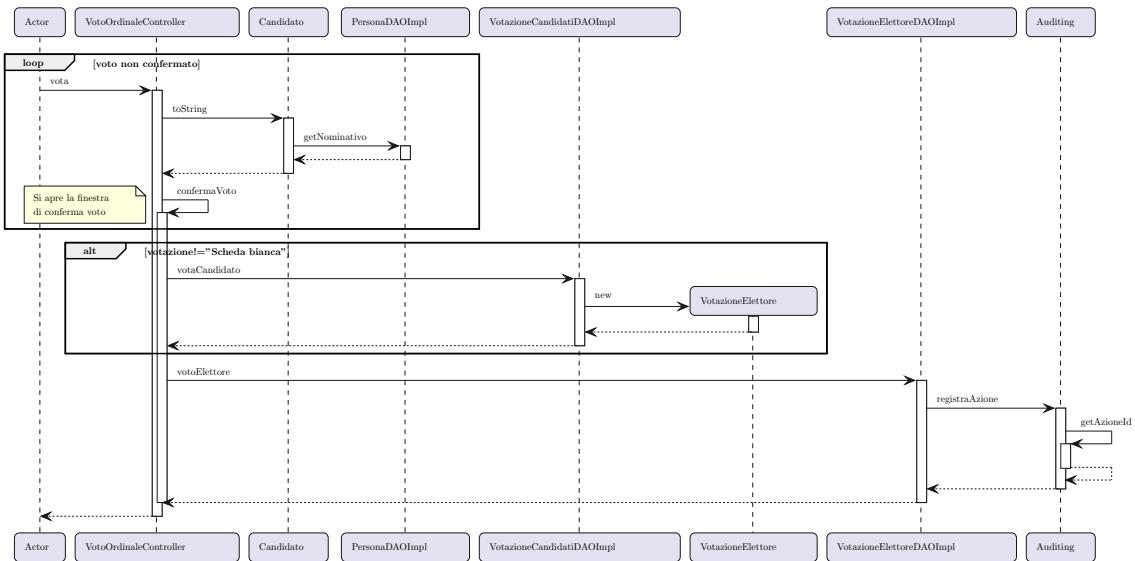


Figura 8: Diagramma di sequenza che mostra i passi una votazione di tipo ordinale

2.4.4 Votazione categorica

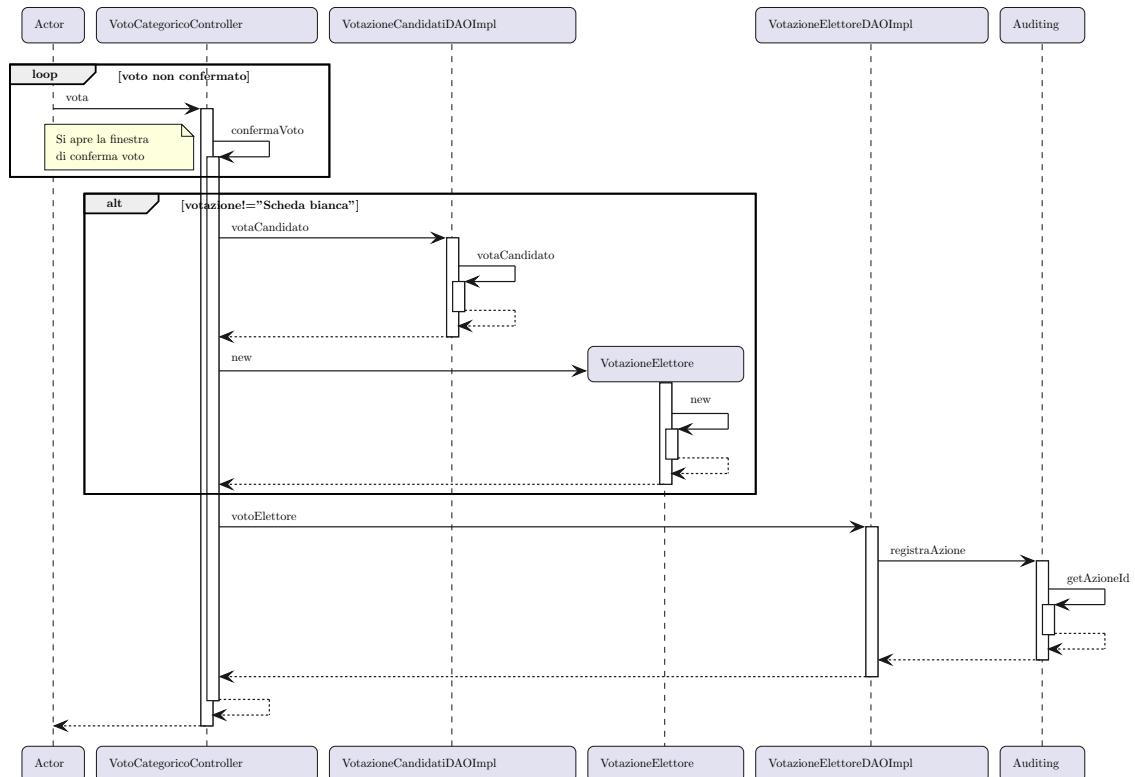


Figura 9: Diagramma di sequenza che mostra i passi per una votazione di tipo categorica

2.4.5 Votazione referendum

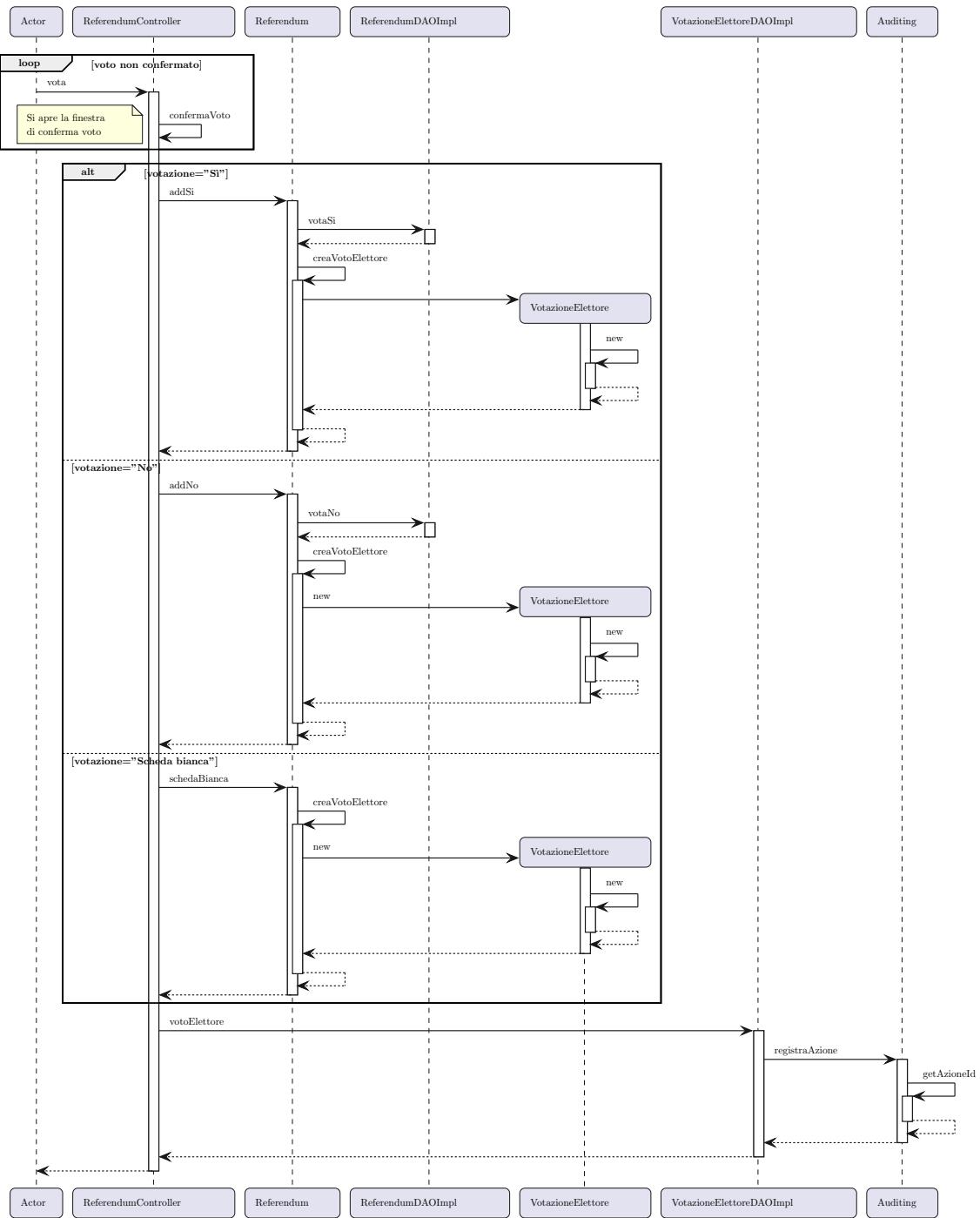


Figura 10: Diagramma di sequenza che mostra i passi per la creazione di una sessione

2.5 Diagrammi delle attività

2.5.1 Accesso

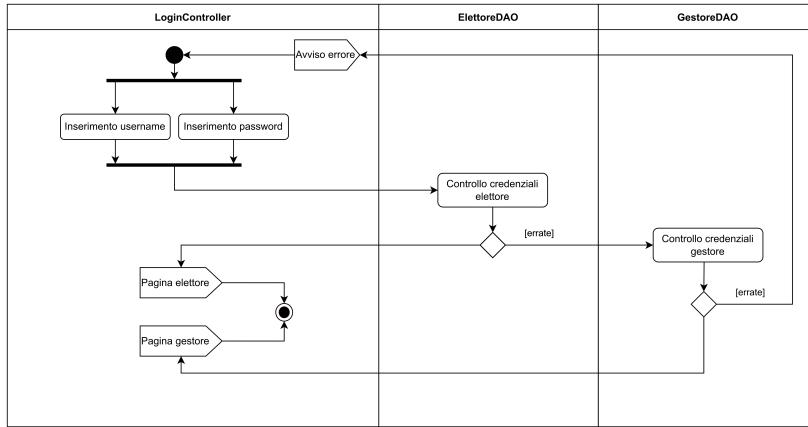


Figura 11: Diagramma di attività dell'accesso

2.5.2 Votazione

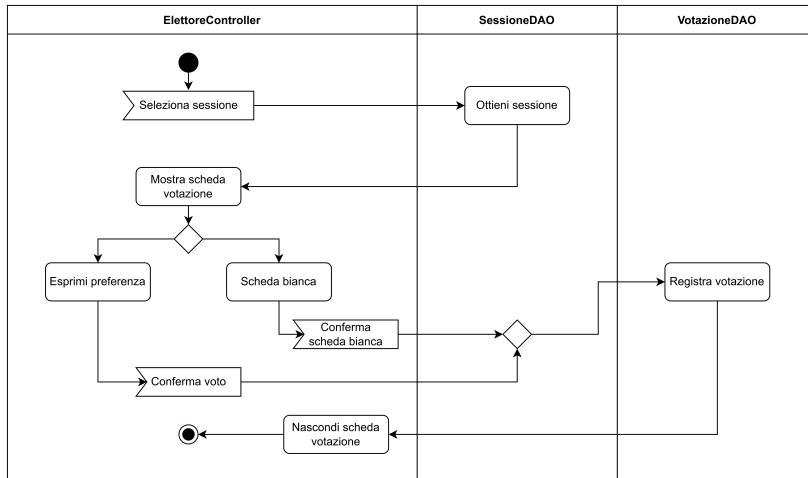


Figura 12: Diagramma di attività della votazione

2.5.3 Creazione della sessione

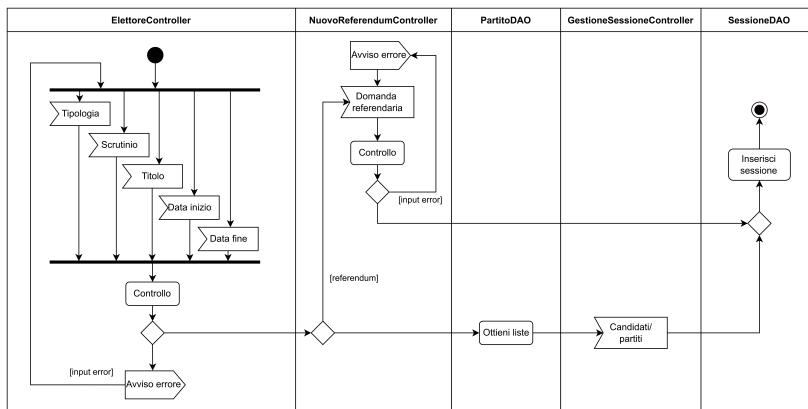


Figura 13: Diagramma di attività della creazione di una sessione

2.6 Diagramma dei componenti

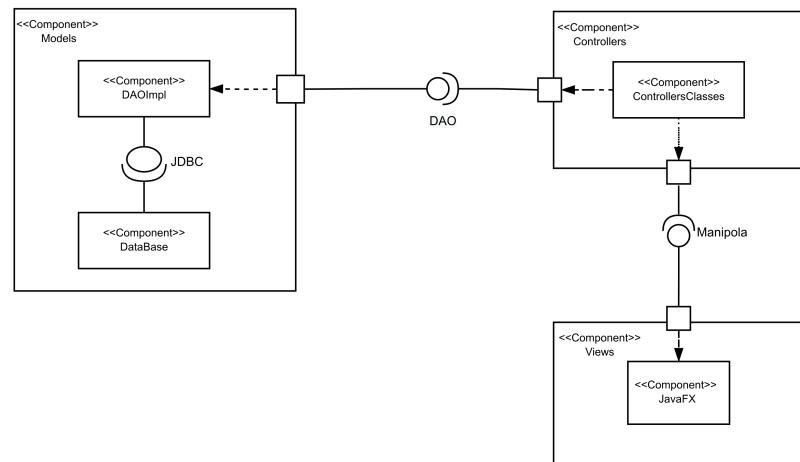


Figura 14: Diagramma dei Componenti

3 Implementazione del Sistema

3.1 Design Pattern

Nel sistema sono stati utilizzati diversi pattern:

- Model View Controller (MVC): Utilizzato per separare l'interfaccia grafica dalla sua manipolazione. L'intera applicazione è stata divisa in tre packages, uno che contiene i modelli, uno che contiene le view di JavaFX e uno che contiene tutti i controller relativi alle view.
- Data Abstract Object (DAO): Utilizzato per garantire una manutenibilità maggiore, separando la parte di accesso e di interazione al Database dalla parte logica.
- Singleton: Utilizzato per garantire che venga creata esattamente una sola istanza, in particolar modo per tutte le classi che implementano il pattern DAO.
- Builder: Utilizzato per facilitare la creazioni di oggetti il cui costruttore presenta molti argomenti come *Sessione*.
- Observer/Observable: Utilizzato per una gestione ottimale e reattiva delle componenti dell'interfaccia grafica.

3.2 Gestione dei dati persistenti

La base di dati utilizzata è stata realizzata con il DBMS open-source PostgreSQL ed è composta dalle seguenti tabelle:

1. *Auditing*, contiene i log del sistema; è formata dai seguenti campi:
 - id (integer): l'identificativo del log
 - orario (timestamp): l'orario in cui l'azione è stata compiuta
 - ruolo_chi (varchar): stringa che indica se l'azione è stata eseguita da un Elettore o da un Gestore
 - chi (varchar): codice fiscale di chi ha effettuato l'azione
2. *Azioni Auditing*, contiene la lista delle azioni che possono essere memorizzate nel sistema di auditing, composto dai seguenti campi:
 - id (integer): identificativo del log
 - azione (carchar): azione eseguita (*Login*, *Votazione*, *Apertura Sessione*, *Chiusura Sessione*, *Logout*)
3. *Candidati*, composto dai seguenti campi:
 - id (integer): identificativo del candidato
 - persona (varchar): codice fiscale del candidato
 - partito (integer): identificativo del partito di appartenenza del candidato
 - ruolo (varchar): ruolo ricoperto dal candidato nel partito di appartenenza
4. *Elettori*, composto dai seguenti campi:
 - cf (varchar): codice fiscale dell'Elettore
 - password (varchar): password dell'Elettore
 - diritto_voto (boolean): indica se l'Elettore possiede il diritto di voto
 - diritto_voto_distanza (boolean): indica se l'Elettore possiede il diritto di voto a distanza
5. *Gestori*, composto dai seguenti campi:
 - cf (varchar): codice fiscale del Gestore

- `password` (`varchar`): password (criptata) del Gestore

6. *Partiti*, composto dai seguenti campi:

- `id` (`integer`): identificativo del partito
- `nome` (`varchar`): nome del partito

7. *Persone*, Contiene i dati anagrafici relativi alle persone, composto dai seguenti campi:

- `cf` (`varchar`): codice fiscale della persona
- `nome` (`varchar`): nome della persona
- `cognome` (`varchar`): cognome della persona
- `nazionalita` (`varchar`): ISO 3166-1 del paese di nascita
- `luogo_nascita` (`varchar`): provincia di nascita della persona
- `data_nascita` (`date`): data di nascita della persona

8. *Referendum*, composto dai seguenti campi:

- `sessione` (`integer`): identificativo della Sessione
- `quesito` (`varchar`): quesito da sottoporre agli Elettori
- `si` (`integer`): numero di votazioni favorevoli al referendum
- `no` (`integer`): numero di votazioni negative al referendum

9. *Sessioni*, contiene le informazioni relative alle sessioni di votazione, composto dai seguenti campi:

- `id` (`integer`): identificativo della Sessione
- `titolo` (`varchar`): titolo della votazione
- `data_apertura` (`date`): data di apertura della Sessione
- `data_chiusura` (`date`): data di chiusura della Sessione
- `tipo_votazione` (`varchar`): tipo di votazione
- `tipo_scrutinio` (`varchar`): tipo di scrutinio
- `chiuse` (`boolean`): indica se la Sessione è aperta o chiusa
- `gestore` (`varchar`): codice fiscale del Gestore
- `votazioniPartiti` (`boolean`): indica se la votazione riguarda i partiti oppure i candidati

10. *VotiCandidati*, contiene le votazioni favorevoli verso i candidati

- `candidato` (`integer`): identificativo del Candidato
- `sessione` (`integer`): identificativo della Sessione
- `voti` (`integer`): numero di voti espressi per la Sessione

11. *VotiElettori*, contiene le votazioni espresse dagli elettori

- `elettore` (`varchar`): identificativo dell'Elettore
- `sessione` (`integer`): identificativo della Sessione
- `orario` (`timestamp`): orario in cui viene espresso il voto

12. *VotiPartiti*, contiene le votazioni favorevoli verso i partiti

- `partito` (`integer`): identificativo del Partito
- `sessione` (`integer`): identificativo della Sessione
- `voti` (`integer`): numero di voti espressi per il Partito

3.2.1 Schema logico

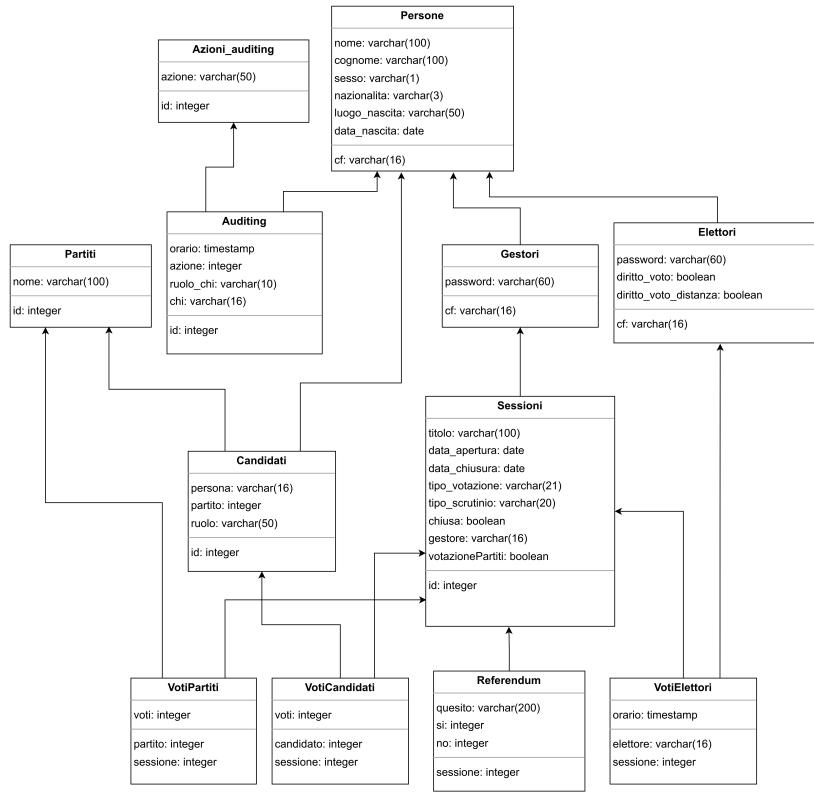


Figura 15: Schema logico della base di dati

3.3 Descrizione dell’interfaccia grafica

L’interfaccia grafica del sistema è stata implementata attraverso l’utilizzo di JavaFX.

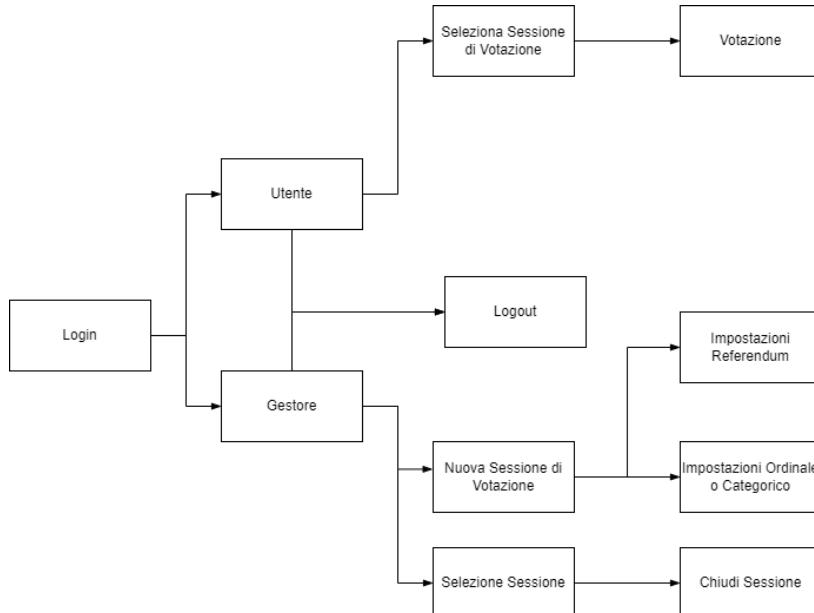


Figura 16: Mappa dell’interfaccia

3.3.1 Login

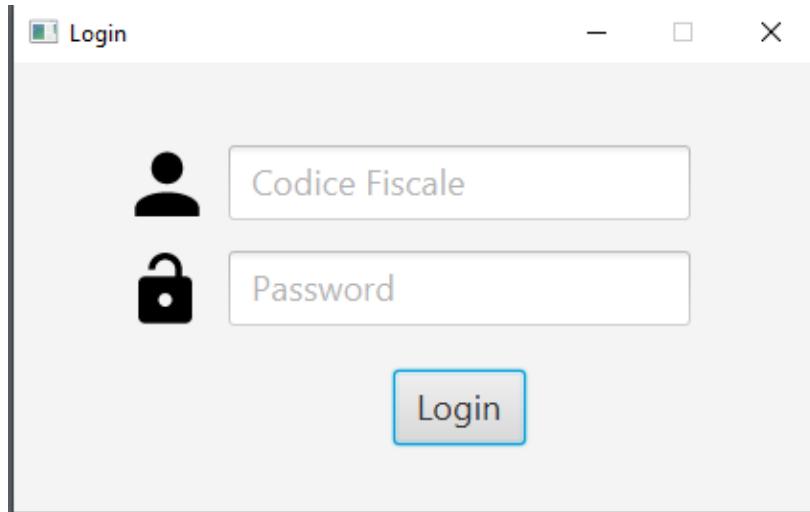


Figura 17: Login

La finestra per effettuare il login permette ad un Utente di identificarsi sia come Elettore ed accedere alle votazioni sia come Gestore ed accedere alla gestione delle proprie sessioni.
Questa finestra è composta dalle seguenti componenti:

- *Input codice fiscale*: L'Utente inserisce il proprio codice fiscale per identificarsi
- *Input password*: L'Utente deve inserire la password corretta per il codice fiscale inserito. Il testo viene oscurato per garantire una maggiore sicurezza.
- *Pulsante "Login"*: L'utente preme il tasto per completare l'identificazione. In questo momento vengono fatti i controlli sulle stringhe inserite nei text field sopracitati, controllando che non siano nulle o vuote (ed eventualmente mostrando un alert) e che le credenziali inserite corrispondano effettivamente ad un Utente nel database.

Nel caso in cui le credenziali siano corrette e corrispondano a quelle di un Elettore verrà mostrata la finestra elettore, mentre nel caso in cui corrispondano a quelle di un Gestore verrà mostrata la finestra per la gestione delle Sessioni (entrambe descritte in seguito).

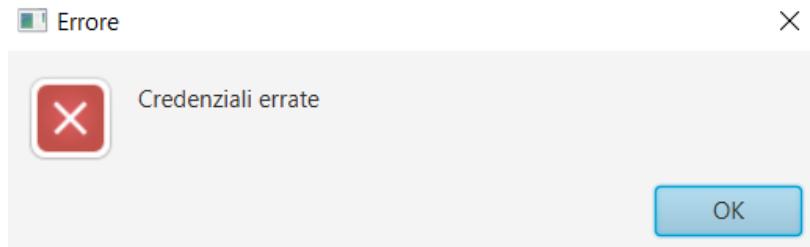


Figura 18: Alert credenziali errate

Nel caso in cui le credenziali siano invece errate viene mostrato un avviso di errore.

3.3.2 Finestra Elettore

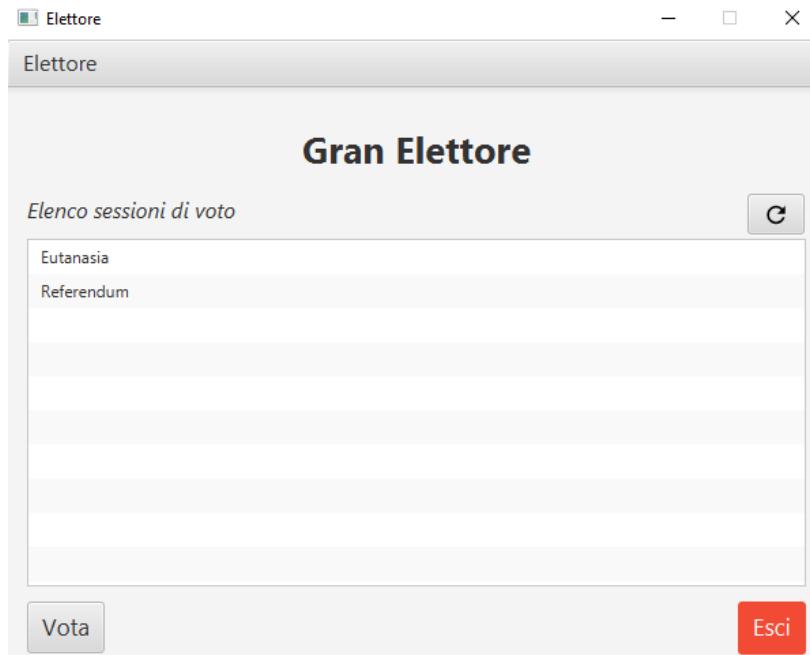


Figura 19: Finestra Elettore

Effettuato il login come Elettore è possibile accedere ad una sessione di voto selezionando una tra le sessioni disponibili.

In questa finestra sono presenti 3 pulsanti:

- Pulsante Refresh: Permette di eseguire il refresh delle sessioni attualmente disponibili
- Pulsante "Esci": Permette di eseguire il logout dell'Elettore corrente
- Pulsante "Vota": Permette di accedere alla votazione della sessione selezionata

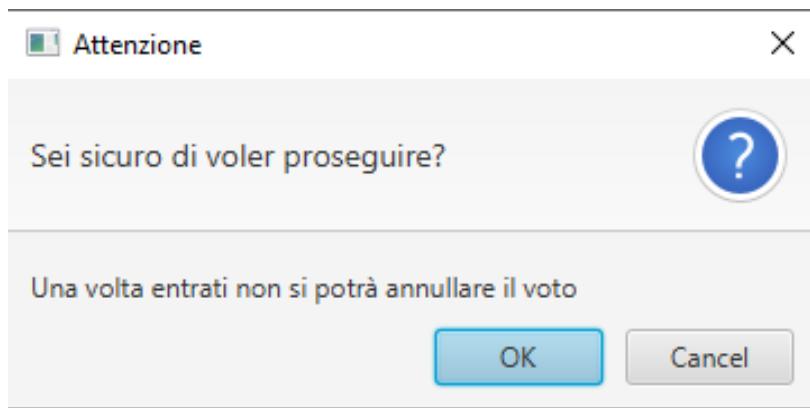


Figura 20: Alert Elettore

Una volta premuto il pulsante "Vota" verrà mostrato un alert che informa l'Elettore che non sarà possibile tornare indietro e che quindi dovrà necessariamente effettuare la votazione.



Figura 21: Votazione Elettore (caso voto ordinale)

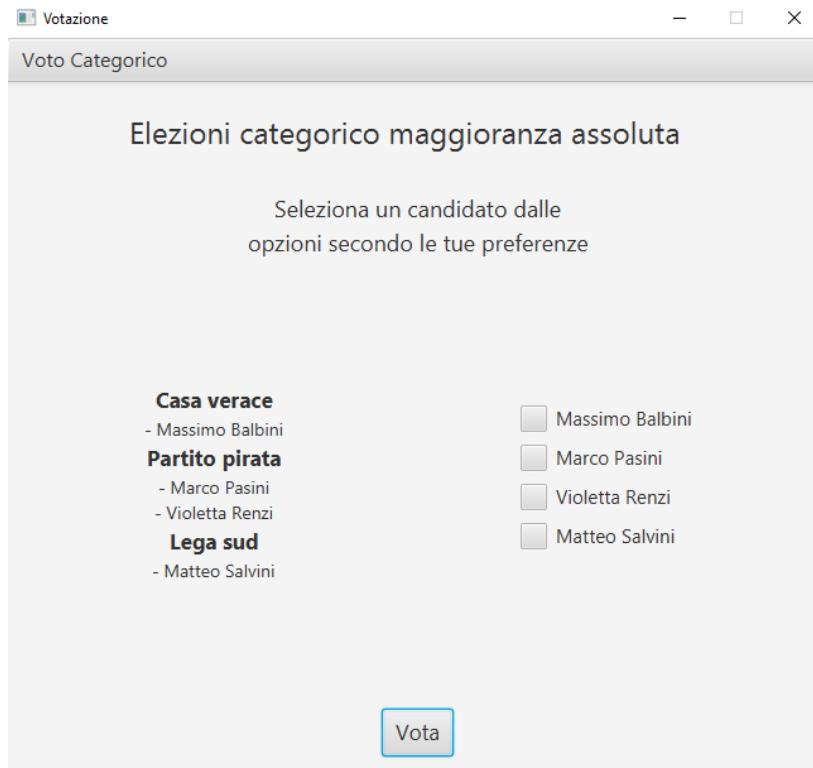


Figura 22: Votazione Elettore (caso voto categorico)



Figura 23: Votazione Elettore (caso voto referendum)

La finestra di votazione è differente a seconda del tipo di votazione:

- *Referendum*: Viene mostrata la domanda con due checkbox per il SI e per il NO
- *Voto Ordinale*: Vengono mostrati i partiti e l'Elettore deve ordinarli secondo le proprie preferenze
- *Voto Categorico*: Vengono mostrati i partiti e l'Elettore deve scegliere il candidato del partito desiderato

3.3.3 Finestra Gestore



Figura 24: Finestra principale Gestore

Effettuando il login come Gestore è possibile accedere alle Sessioni create dallo stesso Gestore (sia aperte che chiuse), visualizzare i dettagli (*titolo*, *id*, *data di apertura e chiusura*, *tipo di votazione*, *tipo di scrutinio*) e chiudere la sessione aperta selezionata.

In questa finestra sono presenti 4 pulsanti:

- Pulsante "Chiudi Sessione": Permette di chiudere la sessione selezionata nelle Sessioni attive
- Pulsante "Nuova Sessione di Voto": Apre la finestra "*Nuova Sessione di Votazione*" che permette al Gestore di creare una nuova Sessione
- Pulsante "Visualizza esiti": Apre la finestra "*Esiti*" che permette di visualizzare i risultati della Sessione di votazione selezionata
- Pulsante "Logout": effettua il logout dell'attuale gestore

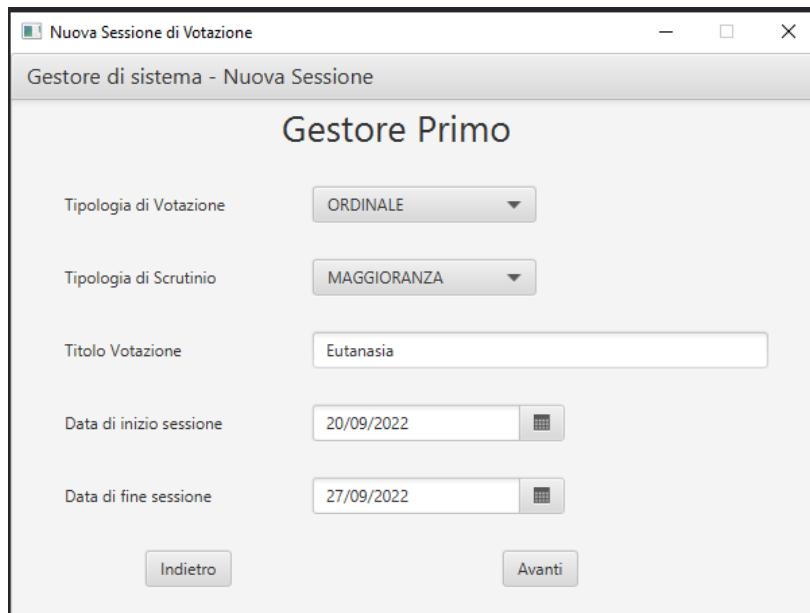


Figura 25: Finestra creazione nuova Sessione

In questa finestra il Gestore può creare una nuova Sessione di votazione, andando a specificare la tipologia di votazione e di scrutinio, il titolo, la data di apertura e di chiusura della sessione. Come per il caso dell'Elettore, premendo il pulsante "Avanti", è possibile andare a finalizzare la Sessione, selezionando i partiti nel caso si sia selezionato un Voto ordinale o categorico, oppure inserendo la domanda nel caso si sia selezionato un Referendum come tipo di Votazione.

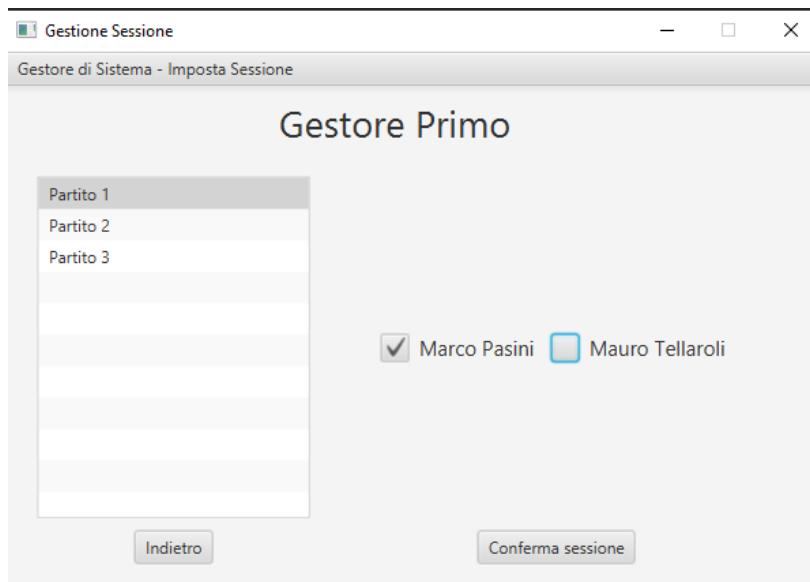


Figura 26: Finestra impostazioni (voto Ordinale o Categorico)

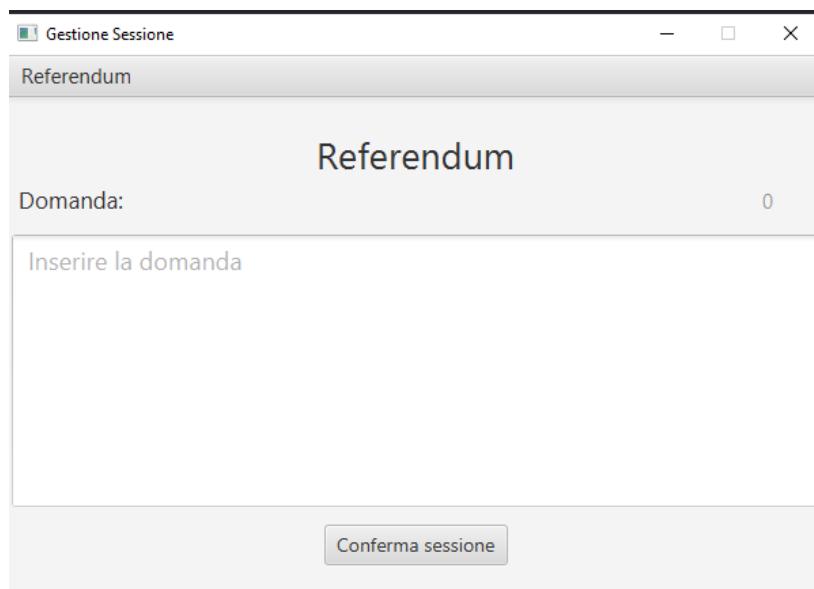


Figura 27: Finestra impostazioni (Referendum)

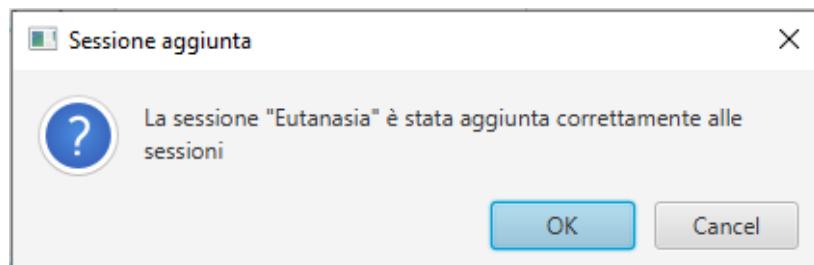


Figura 28: Alert di avvenuta creazione della Sessione

Premendo il pulsante "*Conferma Sessione*" un messaggio conferma l'avvenuto inserimento della Sessione.

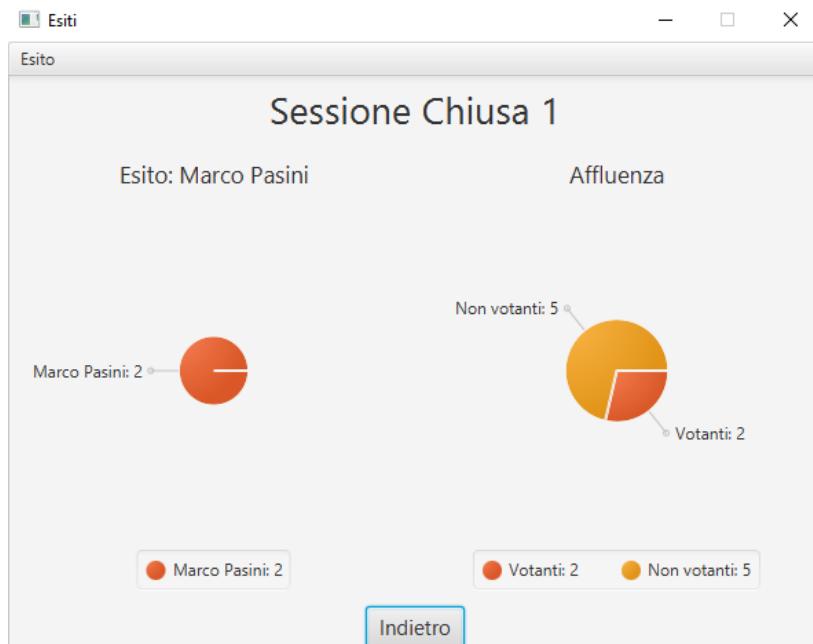


Figura 29: Visualizzazione esiti Sessione

Una volta che la Sessione è stata chiusa dal Gestore è possibile visualizzare i risultati e l'affluenza al voto.

3.4 Diagramma di Deployment

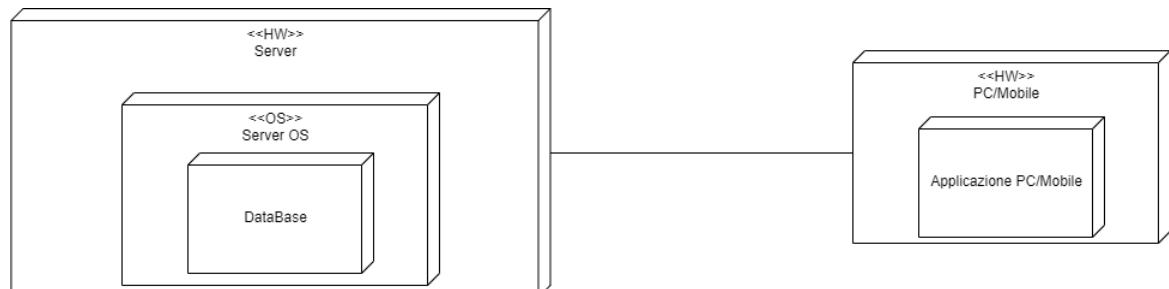


Figura 30: Diagramma di Deployment

3.5 Specifica dei vincoli

Di seguito vengono riportati alcuni vincoli OCL, implementati con JML, dei casi più di rilievo nel sistema, tralasciando i frequenti e banali controlli *notNull*.

3.5.1 Sessione

```

1 public class SessioneSemplice {
2
3     private final int id;
4     //@ invariant titolo!=null && titolo.length()>0;
5     private final String titolo;
6     ...
7 }
```

```

1 public class Sessione extends SessioneSemplice {
2
3     //@ invariant dataApertura!=null && dataApertura.compareTo(dataChiusura)<=0;
4     private final LocalDate dataApertura;
5     //@ invariant dataChiusura!=null && dataChiusura.compareTo(dataApertura)>0;
6     private final LocalDate dataChiusura;
7     //@ invariant tipoVotazione!=null;
8     private final TipoVotazione tipoVotazione;
9     private final boolean votazionePartiti;
10    //@ invariant tipoScrutinio!=null;
11    private final TipoScrutinio tipoScrutinio;
12    private boolean chiusa;
13    //@ invariant gestore!=null && gestore.length()>0;
14    private final String gestore;
15    ...
16 }
```

3.5.2 Persona

```

1 public class Persona {
2
3     //@ invariant nome!=null && nome.length()>0;
4     private final String nome;
5     //@ invariant cognome!=null && cognome.length()>0;
6     private final String cognome;
7     // Codice fiscale
8     //@ invariant CF!=null && CF.length()>0 && checkCF(nome,cognome,dataNascita,
9         sesso,nazionalita,CF);
10    private final String CF;
11    //@ invariant sesso=='M' || sesso=='F';
12    private final char sesso;
13    // nazionalita in formato iso (3 lettere)
14    //@ invariant nazionalita!=null && nazionalita.length()==3;
15    private final String nazionalita;
16    //@ invariant luogoNascita!=null && luogoNascita.length()>0;
17    private final String luogoNascita;
18    //@ invariant dataNascita!=null && dataNascita.compareTo(LocalDate.now())<0;
19    private final LocalDate dataNascita;
20    ...
}
```

3.5.3 Votazione

```

1 public abstract class Votazione {
2
3     // sessione relativa ai voti
4     //@ invariant sessione!=null;
5     private Sessione sessione;
6     // numero di votazioni effettuate dagli elettori
7     //@ invariant numeroVoti>=0;
8     private int numeroVoti;
9     ...
10 }
```

3.6 Testing

Il testing è stato effettuato utilizzando l'ultima versione di JUnit. Sono state testate le seguenti interazioni con la base di dati:

- test di login di un elettore;
- test di login di un gestore;
- test di creazione di una sessione;
- test di chiusura di una sessione;
- test di creazione di un referendum legato alla sessione;
- test di verifica sullo stato di chiusura della sessione;
- test di recupero degli esiti del referendum legato alla sessione;
- test di recupero del numero di votanti al referendum.

Mentre per i modelli sono stati testate le seguenti interazioni:

- test sessione con data chiusura precedente a data apertura;
- test sessione con tipo di votazione non compatibile con tipo di scrutinio;
- test persona con sesso diverso da 'M' o 'F';
- test persona con codice fiscale non compatibile con i dati anagrafici;
- test persona con nazionalità non in formato ISO a 3 lettere;
- test votazione con numero voti totali negativo.

3.7 Note per l'installazione e l'utilizzo

Il progetto è stato sviluppato con:

- **SDK 18.0.2** come Software Development Kit;
- **IntelliJ IDEA** come IDE per lo sviluppo del codice Java;
- **diagrams.net** per la creazione di alcuni diagrammi;
- **Puml** per la creazione di alcuni diagrammi;
- **Github** per lo sviluppo in contemporanea con più persone (2);
- **DataGrip** per la gestione della base di dati.

Le altre dipendenze sono indicate nel file "pom.xml".

Per poter avviare l'applicativo, eseguire i comandi:

```
1 $ mvn clean compile -f pom.xml
2 $ mvn exec:java -f pom.xml
```

Nella cartella "src/data" è presente il dump del database utilizzato "dump.sql". Al suo interno sono presenti 7 persone, 5 elettori, 3 gestori, 4 candidati, 3 partiti e 6 sessioni. Come credenziali di test si può usare:

- Elettore:
 - **CF:** "E"
 - **Password:** "E"
- Gestore:
 - **CF:** "E"
 - **Password:** "G"

Chiaramente per queste credenziali di test è stato bypassato il controllo sul codice fiscale.