

# PROGETTO DI INGEGNERIA DEL SOFTWARE

# RELAZIONE

ANDREA BERETTA 932874
DAVIDE BETTERO 923212

A.A. 2021/2022

## **INDICE**

1. Descrizione del problema	2
1.1 Glossario	
1.2 Analisi e specifica dei requisiti	3
1.2.1 Requisiti utente	
1.2.2 Requisiti di sistema	
1.3 Requisiti non funzionali	
1.3.1 Requisiti di affidabilità	
1.3.2 Requisiti di semplicità	
1.3.3 requisiti di manutenibilità	
1.3.4 requisiti di portabilità	
1.3.5 requisiti di usabilità	
1.3.6 Requisiti legali	
1.3.7 Requisiti di sicurezza	6
2. Progettazione del sistema	7
2.1 Diagramma dei casi d'uso	7
2.2 Diagrammi di sequenza	10
2.3 Diagrammi delle attività	15
2.4 Macchine di stato	18
2.5 Diagramma dei componenti	18
2.6 Diagramma di deployment	19
3. Implementazione del sistema	20
3.1 Diagramma delle classi di progetto	20
3.2 Discussione dei Design Pattern utilizzati	20
3.3 Gestione dei dati persistenti	21
3.4 Descrizione dell'Interfaccia Grafica	25
3.5 Specifica e verifica dei vincoli	32
3.6 Descrizione del testing	33
3.7 Note per l'installazione e l'utilizzo	34

## 1. Descrizione del problema

Il sistema permette di effettuare una votazione elettronica e si pone l'obbiettivo di garantire le stesse caratteristiche e funzionalità di una votazione "tradizionale" cartacea, ma in modalità digitale. In particolare, vengono aggiunte le funzionalità necessarie a rendere tale votazione conforme alle norme di legge varate per garantire lo svolgimento di questo nuovo tipo di elezioni.

È ideato per svolgere quattro tipi di votazione, voto ordinale, voto categorico, voto categorico con preferenze e referendum, e prevede due possibili tipi di utilizzatori, l'elettore e l'impiegato/gestore del sistema.

### 1.1 Glossario

Terminologia specifica utilizzata all'interno del documento:

#### Diritto di voto

Diritto che assicura a un individuo la possibilità di manifestare la propria volontà durante un'elezione.

#### Voto ordinale

Modalità di voto in cui all'elettore è richiesto di ordinare i candidati (o gruppi/partiti) presenti nella scheda in base alle proprie preferenze.

#### Voto categorico

Modalità di voto in cui l'elettore inserisce una preferenza per un candidato (o gruppo/partito).

#### • Voto categorico con preferenze

Modalità di voto in cui l'elettore inserisce una preferenza per un gruppo/partito e ha la possibilità di indicare una o più preferenze tra i candidati del gruppo/partito selezionato (niente voto disgiunto).

#### • Referendum

Modalità di voto che consiste in una domanda fatta all'elettorato con la quale si chiede se si sia favorevoli o contrari a un determinato quesito.

#### Maggioranza

Modalità di definizione del vincitore mediante la quale quest'ultimo è il candidato che ha ottenuto il maggior numero di voti.

## Maggioranza assoluta

Modalità di definizione del vincitore mediante la quale quest'ultimo è il candidato che ha ottenuto la maggioranza assoluta dei voti, cioè il 50% + 1 dei voti espressi.

#### • Referendum senza quorum

Referendum in cui si procede al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.

#### Referendum con quorum

Referendum in cui si procede al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto.

#### • Elettore

Colui che ha la facoltà di eleggere e in particolare che ha diritto di voto nelle elezioni.

• Impiegato/gestore del sistema

Colui che ha la facoltà di amministrare la sessione di voto nella sua completezza.

#### 1.2 Analisi e specifica dei requisiti

#### 1.2.1 Requisiti Utente

- 1. All'avvio del sistema deve essere possibile scegliere il tipo d'utente.
- 2. L'utente deve avere la possibilità, prima di inserire le credenziali d'accesso, di modificare la tipologia d'utente selezionata in caso di scelta errata.
- 3. L'utente amministratore deve potersi autenticare prima di poter amministrare la sessione di voto.
- 4. L'utente amministratore deve poter selezionare la modalità di voto.
- 5. L'utente elettore deve potersi identificare se la votazione avviene a distanza.
- 6. L'amministratore deve poter configurare una sessione di voto, specificando le modalità di voto e le modalità di calcolo del vincitore.
- 7. L'amministratore del sistema deve poter far terminare la votazione e visualizzare l'esito del voto.
- 8. L'utente elettore deve aver la possibilità di effettuare il voto.
- 9. L'utente deve avere la possibilità di non confermare il voto e correggerlo.
- 10. Gli elettori dei comuni con più di quindicimila abitanti possono esprimere anche una doppia preferenza per i candidati al consiglio comunale della lista selezionata, purché di genere diverso, ovvero un maschio e una femmina.

#### 1.2.2 Requisiti di sistema

- 1. Il sistema deve prevedere una schermata iniziale che permette di selezionare se l'utente è un elettore o un amministratore.
- 2. Se l'utente sbaglia a selezionare la tipologia il sistema deve consentirgli, attraverso un apposito pulsante, di tornare all'interfaccia iniziale per permettergli di selezionare nuovamente la tipologia d'utente.
- 3. Il sistema deve garantire che l'utente amministratore effettui, mediante un'apposita schermata, la fase di autenticazione al sistema prima di effettuare qualunque operazione inerente alla sessione di voto.
- 4. Se l'utente è l'impiegato/gestore del sistema ci deve essere una schermata che gli permette di scegliere tra possibili diverse modalità di voto che sono:
  - voto ordinale;
  - voto categorico;
  - voto categorico con preferenze;
  - referendum.
- 5. L'applicazione permette se l'utente è un elettore, nel caso di votazione a distanza, di identificarsi inserendo in una apposita schermata le credenziali fornitegli.

- 6.
- 6.1. Il sistema prevede un'interfaccia grazie alla quale l'impiegato/gestore del sistema può configurare la tipologia di votazione scegliendo tra le diverse modalità. Le diverse modalità di voto sono:
  - voto ordinale;
  - voto categorico;
  - voto categorico con preferenze;
  - referendum.
- 6.2. Il sistema prevede solo all'impiegato/gestore del sistema un menù con un'opzione per fargli specificare la modalità di calcolo del vincitore.

7.

- 7.1. Il sistema deve permettere all'impiegato/gestore del sistema mediante un pulsante di far terminare la sessione di voto.
- 7.2. Il sistema deve richiedere all'impiegato/gestore del sistema la conferma riguardo la volontà di far chiudere la sessione di voto.
- 7.3. Il sistema deve poter mostrare all'impiegato/gestore del sistema l'esito della votazione. Il calcolo del vincitore può essere effettuato tramite le seguenti modalità:
  - maggioranza;
  - maggioranza assoluta;
  - referendum senza quorum;
  - referendum con quorum.

8.

- 8.1. Il sistema garantisce che ogni scheda contenga:
  - o un voto valido;
  - o sia bianca.
- 8.2. Il sistema, dopo che l'utente ha effettuato la sua scelta, mostra attraverso un'interfaccia il riassunto del voto effettuato e chiede attraverso un apposito pulsante la conferma all'elettore.
- 8.3. Il sistema, una volta che l'utente elettore ha effettuato e confermato la sua votazione, prevede che venga effettuato il logout automaticamente.
- 9. Il sistema deve permettere all'utente elettore attraverso un pulsante di non confermare la sua scelta di voto e di ritornare all'interfaccia di votazione. Deve quindi a questo punto permettergli di selezionare nuovamente le sue preferenze.

10.

10.1. Il sistema deve verificare mediante il processo di autenticazione se l'utente elettore abita in un comune con più di quindicimila abitanti ed entrare in una configurazione dedicata che gli permette di esprimere anche una doppia preferenza.

- 10.2. Il sistema deve verificare che la doppia preferenza che è stata indicata abbia rispettato i termini fissati.
- 10.3. Se l'utente esprime due preferenze a candidati dello stesso genere, il sistema deve annullare la seconda preferenza.

## 1.3 Requisiti non funzionali

### 1.3.1 Requisiti di affidabilità

- 1. Il sistema prevede una chiara procedura da seguire in caso di errore/fallimento del software.
- 2. Il sistema viene testato in modo da verificare eventuali rallentamenti dovuti al collegamento contemporaneo di un numero particolarmente elevato di utenti.
- 3. Il corretto funzionamento del sistema, prima della sua messa in funzione e, in ogni caso, prima di ogni tornata elettorale in cui è necessario il suo impiego, viene verificato dall'amministrazione attraverso la collaborazione tra un Organismo indipendente appositamente nominato e le Autorità preposte alla sicurezza cibernetica nazionale.

## 1.3.2 Requisiti di semplicità

- 1. Il sistema deve essere di agevole comprensione e utilizzo da parte di tutti gli elettori, con informazioni chiare e trasparenti, che vengono presentate in modo sostanzialmente equivalente rispetto alle altre modalità di votazione.
- 2. Il sistema deve consentire alle persone diversamente abili e alle persone con esigenze speciali di votare in modo indipendente, tenuto conto delle particolari situazioni individuali.

## 1.3.3 requisiti di manutenibilità

- 1. Il sistema è decentralizzato così da permettere che, in caso di guasto di un server, continui a funzionare e sia necessario solo sostituire la macchina guasta.
- 2. Il sistema deve essere creato sfruttando i principi dell'astrazione procedurale in modo da facilitare la manutenibilità per eventuali aggiornamenti futuri.

### 1.3.4 requisiti di portabilità

1. Si deve avere la possibilità di effettuare la votazione da qualsiasi tipo di dispositivo dotato di una connessione ad Internet.

#### 1.3.5 requisiti di usabilità

- 1. L'elettore deve essere in grado di verificare, tramite lo schermo del proprio dispositivo, che il voto espresso è quello corrispondente alla sua volontà.
- 2. Una volta che l'elettore ha effettuato la sua scelta, il sistema deve richiedere la conferma di voto.

## 1.3.6 requisiti legali

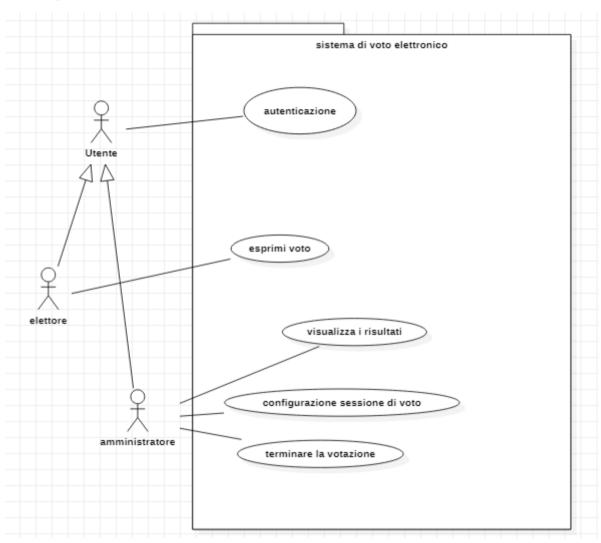
- 1. Ogni elettore deve poter effettuare il voto una sola volta.
- 2. Il sistema non deve in alcun modo, nemmeno indirettamente, influenzare l'elettore nella votazione.
- 3. Il sistema può mostrare i risultati di voto solo a votazione conclusa.
- 4. L'interfaccia utente del sistema di voto elettronico non dove contenere più informazioni rispetto a quelle presenti nelle schede cartacee e nei manifesti con i nominativi dei candidati ammessi.

## 1.3.7 requisiti di sicurezza

- 1. Il sistema deve garantire la segretezza del voto e l'anonimato dell'elettore.
  - 1.1. Il sistema deve garantire che le informazioni personali sui votanti vengano separate da quelle sui voti espressi.
  - 1.2. I voti sono e rimangono anonimi, non devono essere quindi riconducibili all'elettore.

# 2. Progettazione del Sistema

## 2.1 Diagramma dei casi d'uso



## Descrizione dei casi d'uso

Nome		Autenticazione			
Scopo		Permettere di accedere al sistema di voto elettronico e di		Permettere di accedere al sistema di voto elettronico e di	
		votare/gestire la sessione di voto			
Attore/i		Utente			
Pre-condizioni		Nessuna		Nessuna	
Trigger		Il sistema chiede alla persona di autenticarsi per votare.			
Descrizione	sequenza	1. Richiede all'utente di identificarsi inserendo le			
eventi		proprie credenziali d'accesso;			
		2. L'utente in questione deve specificare se è un			
		amministratore o un elettore attraverso l'apposito			
		menu a tendina;			
		3. Premere il tasto "accedi".			

Alternativa/e	1'. se la persona vota in presenza la fase di			
	autenticazione avviene manualmente e l'utente si			
	trova già davanti l'interfaccia per votare.			
Post-condizioni	Se tutto è andato a buon fine permette all'utente di			
	accedere alla votazione elettronica, altrimenti gli nega			
	l'accesso.			

L'utente, che sia un amministratore o un elettore, prima di poter accedere a qualsiasi funzionalità del sistema di voto elettronico deve autenticarsi. Per farlo si devono inserire le proprie credenziali nel form e premere sul tasto apposito per accedere al sistema.

Riferendosi al RASD, questo use case "Autenticazione" è in relazione con i requisiti utente: 2, 3, 5.

Nome	Esprimi voto		
Scopo	Far esprimere il voto		
Attore/i	Elettore		
Pre-condizioni	Avere diritto di voto, ossia essere maggiorenne e non aver		
	ancora votato.		
Trigger	La persona richiede di votare.		
Descrizione sequenza eventi	<ol> <li>La persona seleziona la propria intenzione di voto;</li> <li>Il sistema mostra una schermata riassuntiva contenente la scelta della persona;</li> <li>La persona preme "conferma" per completare il voto oppure "cancella" per fare una scelta diversa, eliminando quella precedente (in questo caso si riparte dal punto 1).</li> </ol>		
Alternativa/e	<ol> <li>1'. Come sopra;</li> <li>2'. Se la votazione è ordinale o categorica e l'elettore è di un comune con più di quindicimila abitanti, può esprimere anche una doppia preferenza per i candidati al consiglio comunale della lista selezionata, purché di genere diverso, ovvero un maschio e una femmina;</li> <li>3'. Come 2 sopra;</li> <li>4'. Come 3 sopra;</li> </ol>		
Post-condizioni	Viene registrato che l'elettore in questione ha votato e viene registrata anche la sua scelta.		

L'utente elettore, se è una persona avente diritto al voto, deve poter richiedere di votare. Per farlo la persona seleziona la propria preferenza di voto e, se la votazione è ordinale o categorica e l'elettore è di un comune con più di quindicimila abitanti, può esprimere anche una doppia preferenza per i candidati al consiglio comunale della lista selezionata, purché di genere diverso, ovvero un maschio e una femmina. Dopo che il sistema gli ha mostrato una schermata riassuntiva l'elettore, conferma o annulla la propria scelta per poterla modificare.

Riferendosi al RASD, questo use case "Esprimi voto" è in relazione con i requisiti utente: 8, 9, 10.

Nome		Configurazione sessione di voto		
Scopo		Configurare la sessione di voto digitale in vista dell'elezione		
Attore/i		amministratore		
Pre-condizioni		La fase di autenticazione per amministratore deve essere		
		completata correttamente.		
Trigger		L'impiegato richiede di configurare la sessione di voto.		
Descrizione s	equenza	<ol> <li>Configurare la modalità di voto scegliendo tra:</li> </ol>		
eventi		<ul> <li>voto ordinale</li> </ul>		
		<ul> <li>voto categorico</li> </ul>		
		<ul> <li>voto categorico con preferenze</li> </ul>		
		<ul> <li>referendum</li> </ul>		
		2. Configurare la modalità di calcolo del vincitore		
		scegliendo tra:		
		<ul> <li>Maggioranza</li> </ul>		
		Maggioranza assoluta		
		Referendum senza quorum		
		Referendum con quorum		
		3. Premere il tasto "termina configurazione" per		
		applicare la configurazione selezionata, o "annulla"		
		per cambiare e modificare uno o più passi della		
		configurazione (in questo caso si riparte dal punto 1).		
Alternativa/e		La configurazione per la votazione a distanza è già stata		
		eseguita.		
Post-condizioni		Configura la sessione di voto conformemente alla scelta		
		dell'impiegato.		

Come amministratore del sistema, la persona vuole configurare la sessione di voto. Per farlo seleziona una tra le modalità di voto disponibili e successivamente la modalità di calcolo dei risultati. A questo punto preme il tasto "termina configurazione" per salvare la configurazione, oppure "annulla" per modificarla ed effettuare una nuova configurazione.

Riferendosi al RASD, questo use case "configurazione sessione di voto" è in relazione con i requisiti utente: 4, 6.

Nome	Visualizza i risultati	
Scopo	Permettere all'impiegato di vedere i risultati dell'elezione	
Attore/i	amministratore	
Pre-condizioni	La votazione deve essere terminata e l'utente in questione	
	deve essere un amministratore	
Trigger	L'amministratore vuole visualizzare i risultati delle elezioni	
	per poterli comunicare	

Descrizione	sequenza	<ol> <li>L'impiegato preme su "visualizza risultati".</li> </ol>		
eventi		2. Il sistema gli mostra l'esito della votazione.		
Alternativa/e		Nessuna		
Post-condizioni		Mostra i risultati dell'elezione		

Come amministratore del sistema, la persona può visualizzare i risultati. Per farlo preme sul tasto "visualizza risultati" e il sistema gli mostra una schermata con l'esito della votazione.

Riferendosi al RASD, questo use case "visualizza i risultati" è in relazione con i requisiti utente: 7.

Nome	Termina votazione		
Scopo	Chiudere la votazione e non permettere più a nessun		
	elettore di votare.		
Attore/i	Impiegato/gestore del sistema		
Pre-condizioni	Deve esserci una sessione di voto aperta e l'utente deve		
	essere un impiegato/gestore del sistema.		
Trigger	L'impiegato vuole terminare la votazione.		
Descrizione sequenza	1. Fa terminare la votazione cliccando su "termina		
eventi	votazione"		
	2. Il sistema mostra una schermata per confermare di		
	voler terminare la sessione di votazione.		
	3. Se l'impiegato riconferma cliccando sul tasto		
	"conferma" il sistema termina la votazione,		
	altrimenti, se preme sul tasto "annulla" il sistema		
	torna al punto 1		
Alternativa/e	La sessione di votazione a distanza viene terminata		
	automaticamente una volta raggiunto il tempo limite.		
Post-condizioni	La votazione viene terminata e non è più possibile votare.		

Come impiegato, la persona può far terminare la votazione. Per farlo clicca su "termina votazione" e a questo punto il sistema chiede conferma dell'azione; in caso positivo la votazione viene chiusa, altrimenti rimane aperta.

Riferendosi al RASD, questo use case "termina votazione" è in relazione con i requisiti utente: 7.

## 2.2 Diagrammi di sequenza

Sono stati implementati i diagrammi di sequenza relativi ai casi d'uso per i quali è stata fornita la descrizione.

Al fine di garantire la leggibilità dei diagrammi è stato deciso di utilizzare in essi una classe "fittizia" Sistema, per sostituire le numerose classi controller dell'interfaccia utente, la stessa soluzione ovviamente è stata adottata per il nome dei metodi appartenenti alla classe Sistema.

checkInDB è l'operazione che rappresenta una query al database.

(la request e la reply associata ad essa sono rispettivamente indicate tramite un LostMessage ed un FoundMessage).

Di seguito vengono presentate le immagini dei diagrammi.

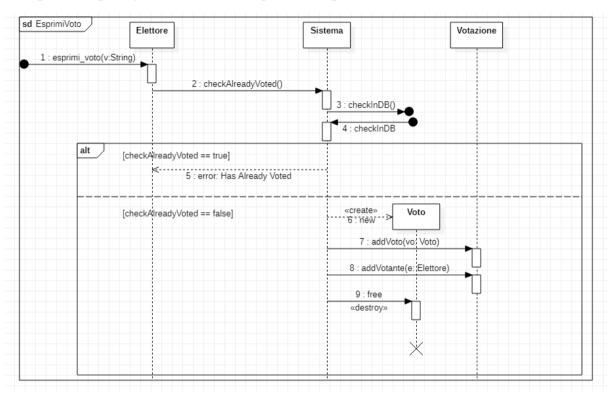


Figura 1: diagramma di sequenza del metodo esprimi\_voto

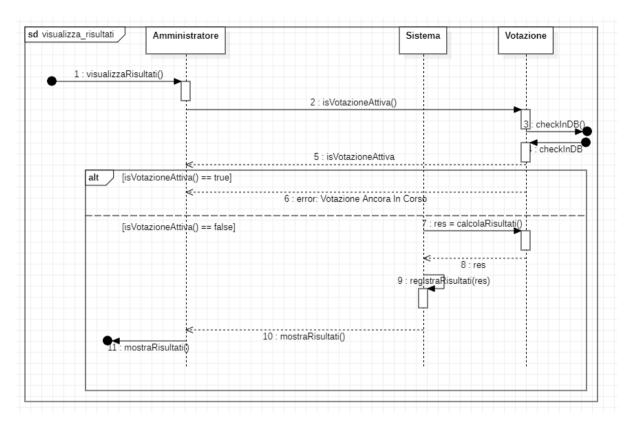


Figura 2: diagramma di sequenza del metodo VisualizzaRisultati

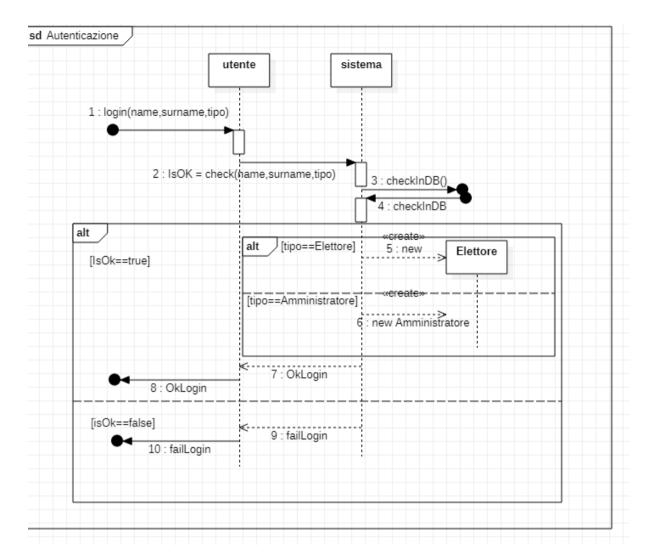


Figura 3: diagramma di sequenza del metodo Autenticazione

L'alt fragment innestato è stato creato per rappresentare i due casi relativi al tipo dell'utente, nel primo caso viene creato un elemento di tipo elettore, altrimenti uno di tipo Amministratore (dal diagramma la creazione dell'elemento Amministratore non risulta chiarissima per via di problemi tecnici relativi a starUML).

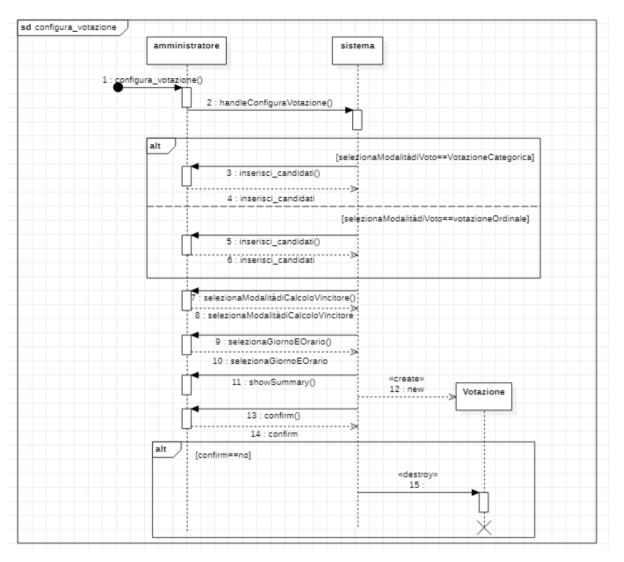


Figura 4: diagramma di sequenza del metodo configura\_votazione

## 2.1 Diagrammi delle attività

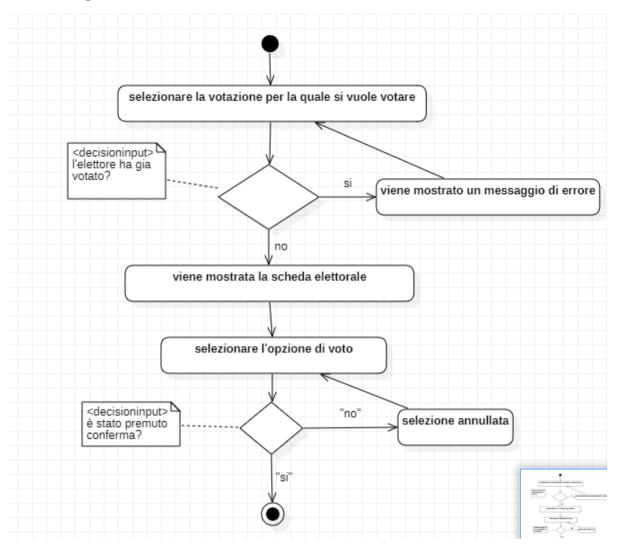


Figura 5diagramma di attività di esprimi\_voto

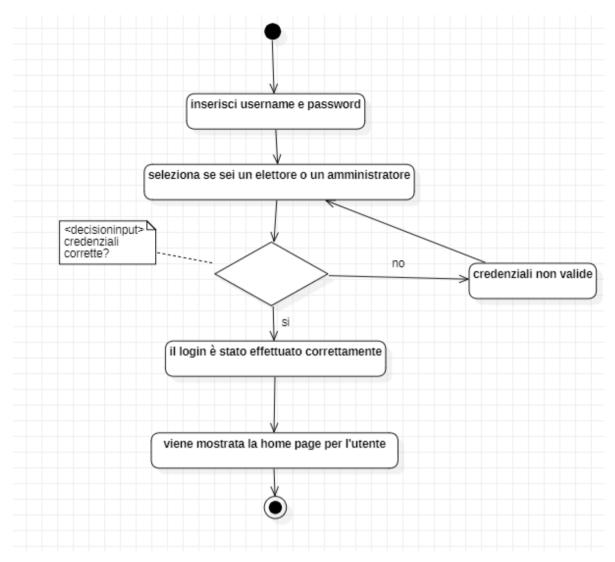


Figura 6:diagramma di attività per login

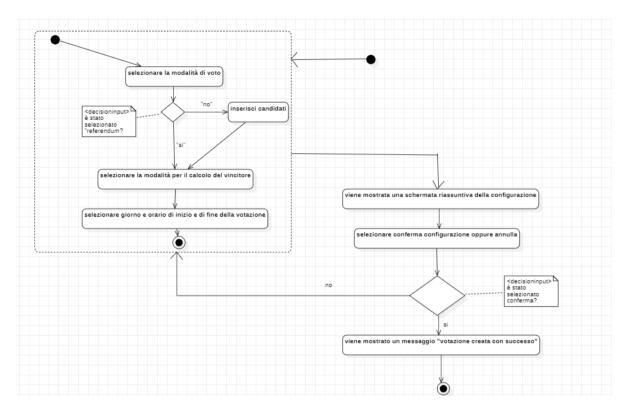


Figura 7:diagramma di attività per configura votazione

## 2.4 Macchine di stato

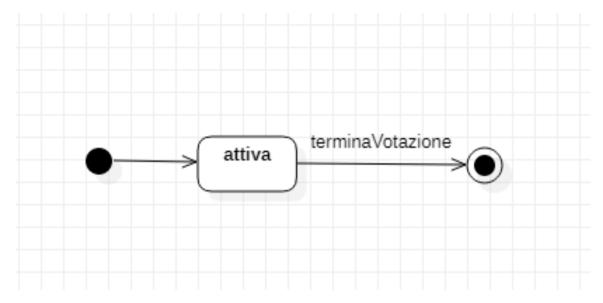
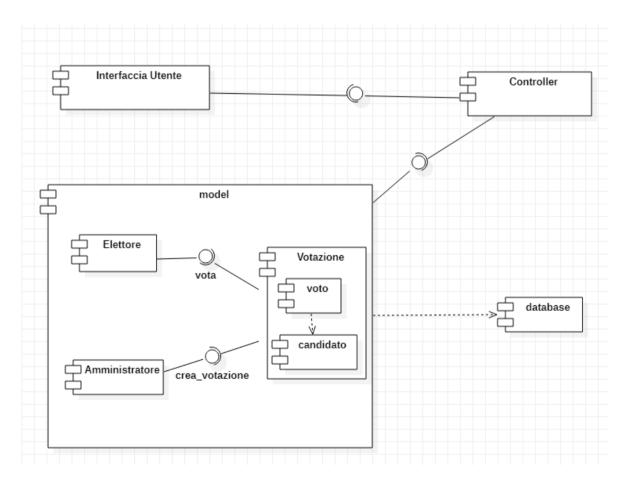
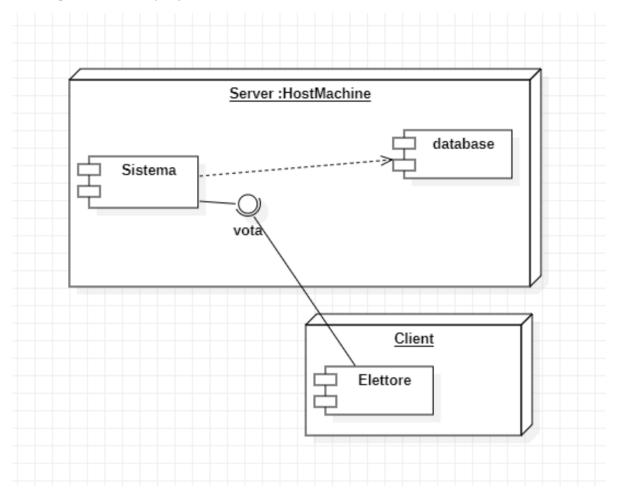


Figura 8: macchina di stato della classe Votazione

## 2.5 Diagramma dei componenti

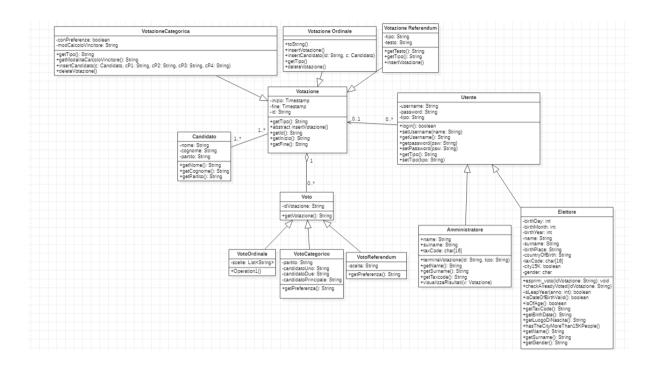


## 2.6 Diagramma di deployment



## 3. Implementazione del sistema

## 3.1 Diagramma delle classi di progetto



## 3.2 Discussione dei Design Pattern utilizzati

I design pattern utilizzati sono il model-view-controller, il singleton ed il Dao.

Il primo è stato fondamentale per il progetto in quanto ha semplificato notevolmente la logica dietro alla comunicazione tra modello e vista; in particolare questo design pattern è stato utilizzato per ciascuna pagina dell'interfaccia utente e prevede che in seguito all'utilizzo di un componente dell'interfaccia utente, venga chiamato il metodo della classe controller con lo scopo di gestire lo specifico evento, aggiornando quindi il modello.

Questa logica è stata molto utile per la configurazione di una votazione, in quanto tramite un'interfaccia utente siamo riusciti a modificare, in seguito ai vari input dell'amministratore, lo stato di un oggetto della classe Votazione (tipo di votazione, data e orario di inizio, ecc...), il costo dell'utilizzo di questa soluzione è naturalmente la creazione di classi controller.

Il pattern DAO è stato utile per separare la logica di acceso ai dati dal resto del progetto.

In particolare, la classe DaoPatternDemo rappresenta il modello, successivamente abbiamo creato un'interfaccia contente tutti i metodi che richiedono di interrogare il database (DigitalVotingDao).

Successivamente abbiamo creato DigitalVotingDaoImpl, l'implementazione dell'interfaccia, racchiudendo appunto in questa classe la parte di progetto che richiede l'interazione con il db.

Durante lo sviluppo dell'interfaccia grafica ci siamo chiesti come fare a garantire che tutta la sessione di voto fosse sempre associata allo specifico utente. Per ottenere questo risultato è necessario avere la possibilità che ogni classe coinvolta nella GUI (classi controller) si possa sempre reperire l'informazione riguardante chi è l'utente che sta eseguendo la procedura per esprimere il voto.

Pertanto, abbiamo implementato la classe SessioneSingleton che ha all'interno un campo privato user, di tipo Utente. La classe è strutturata in modo che possa essere istanziata una sola volta e contiene il metodo statico getSessioneSingleton() che restituisce questa istanza. In questo modo è sufficiente eseguire SessionSingleton().getSessioneSingleton().getUser() per ottenere il nome dell'utente.

### 3.3 Gestione dei dati persistenti

Il dbms utilizzato è PostgreSQL.

L'applicazione utilizza un database contenente i dati anagrafici di votanti e amministratori, ed anche un'altra base di dati contente le votazioni, attive e concluse, con i relativi voti espressi dagli elettori.

Di seguito viene descritto lo schema relativo alla base di dati contente i dati anagrafici.

	Tabella "digital	voting.elettore	2"		
Colonna	Tipo	Ordinamento	Può essere null		
username password taxcode name surname gender birthdate birthplace countryofbirth city15k Indici:	character varying(100) character varying(100) character varying(50) character varying character varying character varying(1) date character varying character varying boolean		not null not null not null		
<pre>"elettore_pkey" PRIMARY KEY, btree (password) "elettore_taxcode_key" UNIQUE CONSTRAINT, btree (taxcode)</pre>					

I dati nella figura sopra sono richiesti durante la fase di registrazione che avviene in un momento precedente alla votazione, pertanto, quando la votazione è attiva, i dati sono già all'interno del database.

All'interno dello schema è presente una tabella "elettore" contente le informazioni relative a ciascun votante:

Username e password sono le credenziali con le quali il votante accederà alla piattaforma, proprio per questo motivo costituiscono una chiave primaria e ciascuno di essi è contrassegnato da un vincolo NOT NULL.

Successivamente troviamo I dati anagrafici relativi alla persona fisica, quali nome, cognome, codice fiscale, genere; giorno, mese e anno di nascita, luogo e Stato in cui l'elettore è nato.

Infine, è presente un attributo di tipo booleano che indica se l'utente appartiene ad una città con più di 15 mila abitanti, al fine di stabilire se l'elettore ha diritto, secondo la legge italiana, in alcuni tipi di votazione, ad esprimere una doppia preferenza.

Per quanto riguarda l'amministratore, è presente una tabella con nome omonimo, che contiene i dati relativi all'impiegato. In particolare, sono presenti username e password, che costituiscono una chiave primaria.

```
Tabella "digitalvoting.amministratore'
Colonna
                     Tipo
                                     Ordinamento
                                                   Può essere null
           character varying(100)
                                                    not null
username
password
           character varying(100)
                                                    not null
           character varying
nome
           character varying
cognome
taxcode
           character varying(16)
ndici:
   "amministratore_pkey" PRIMARY KEY, btree (password)
```

La piattaforma supporta 3 tipi di votazione: ordinale, referendum, categorica.

Per ciascun tipo esiste una tabella all'interno del db.

In ognuna di esse è presente un codice univoco (id), inizio e fine votazione in formato Timestamp. Il codice univoco rappresenta una chiave primaria.

#### **REFERENDUM**

Per il referendum, abbiamo deciso di usare anche due attributi in formato varchar che rappresentano rispettivamente il tipo di referendum (con o senza quorum) ed il testo della votazione.

#### **ORDINALE**

```
Colonna | Tipo | Ordinamento | Può essere null | Default

id | character varying | not null |
inizio | timestamp without time zone | not null |
fine | timestamp without time zone | not null |
Indici:
"ordinale_pkey" PRIMARY KEY, btree (id)
Referenziato da:
TABLE "candidatiordinale" CONSTRAINT "candidatiordinale_id_fkey" FOREIGN KEY (id) REFERENCES ordinale(id)
```

Per la votazione ordinale non abbiamo ritenuto necessario inserire attributi aggiuntivi oltre a quelli comuni alle altre tabelle per le votazioni.

#### **CATEGORICO**

	Tabella "digitalvoting	.categorico"		
Colonna	Tipo	Ordinamento	Può essere null	Default
id	character varying	+ 	not null	<del> </del>
inizio	timestamp without time zone	i		
fine	timestamp without time zone	İ		
conpreferenze	boolean	İ		
modcalcolovincitore	character varying	l		
Indici:				
	PRIMARY KEY, btree (id)			
Vincoli di controllo:				
	lcolovincitore_check" CHECK (m	odcalcolovinci	tore::text = ANY (	ARRAY['maggioranza'::character varying::text, 'maggioranza assoluta'::character varyi
ng::text]))				
Referenziato da:				
TABLE "candidatica	ategorico" CONSTRAINT "candida	ticategorico_i	d_fkey" FOREIGN KE	(id) REFERENCES categorico(id)

Riguardo alla votazione categorica, abbiamo deciso di inserire un attributo booleano che indica se la votazione è con preferenze(true) o meno(false). Abbiamo poi deciso, data la complessità maggiore del tipo di votazione, di inserire un attributo per rappresentare la modalità di calcolo del vincitore, in formato varchar.

Per lo store dei voti espressi durante ciascuna votazione sono presenti 3 tabelle, una per ogni tipo di votazione, il cui nome è composto dalla parola "voti"+tipo di referendum.

In questo tipo di tabelle sono presenti l'identificatore della votazione, che è una chiave esterna che fa riferimento all' id della tabella il cui nome è il tipo di votazione, ed un attributo di tipo varchar che indica il voto, in particolare il tipo di scelta effettuata dall'utente.

```
Tabella "digitalvoting.votireferendum"

Colonna | Tipo | Ordinamento | Può essere null | Default

idvotazione | character varying | | not null |
scelta | character varying | | |
Vincoli di integrità referenziale
"votireferendum_idvotazione_fkey" FOREIGN KEY (idvotazione) REFERENCES referendum(id)
```

Per tenere traccia degli elettori che hanno votato in una specifica votazione è presente una tabella "votanti", nella quale sono presenti 3 attributi:

l'id della votazione ed il codice fiscale dell'elettore che ha votato costituiscono una chiave primaria.

Inoltre, è presente lo username dell'utente che ha votato.

_		italvoting.vota	
Colonna	Tipo	Ordinamento	Può essere null
idvotazione codicefiscale username Indici:	character varying character varying character varying		not null not null
"votanti_pk	ey" PRIMARY KEY, btre	ee (idvotazione	e, codicefiscale)

Infine, abbiamo aggiunto una tabella che funge da log del sistema, in modo tale che vengano registrate le azioni eseguite nel sistema.

Per ciascuna azione vengono registrate data e l'ora in formato <u>Timestamp</u>, mentre l'utente e l'azione compiuta in formato varchar.

Colonna	Tabella "digita   Tipo	0 0	le"   Può essere null
dataeora utente azione Indici: "logtal	timestamp without time zone   character varying   character varying   ble_pkey" PRIMARY KEY, btree (	dataeora)	not null

#### 3.4 Descrizione dell'Interfaccia Grafica

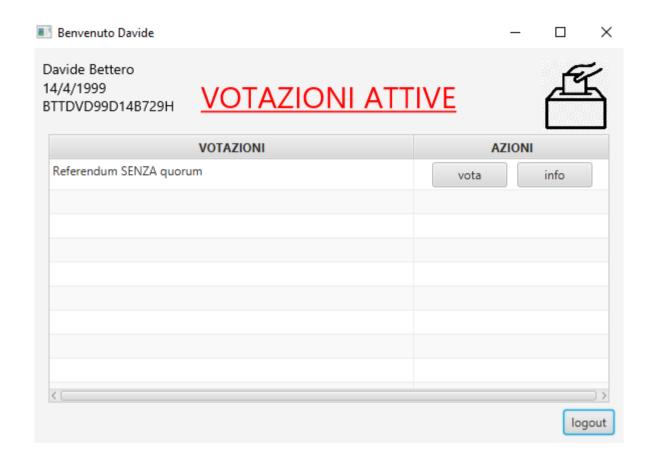
Quando viene eseguita l'applicazione la prima schermata che compare è quella del login.

Nella pagina sono presenti da due container all'interno dei quali è necessario inserire username e password, ed un menu a tendina nel quale si può selezionare il tipo di utente con cui si intende loggarsi. Infine, è presente il button "login" per tentare l'autenticazione.

Se essa fallisce, nella stessa schermata viene mostrato un messaggio di errore e si può ritentare il login, altrimenti viene mostrata la home del sistema.

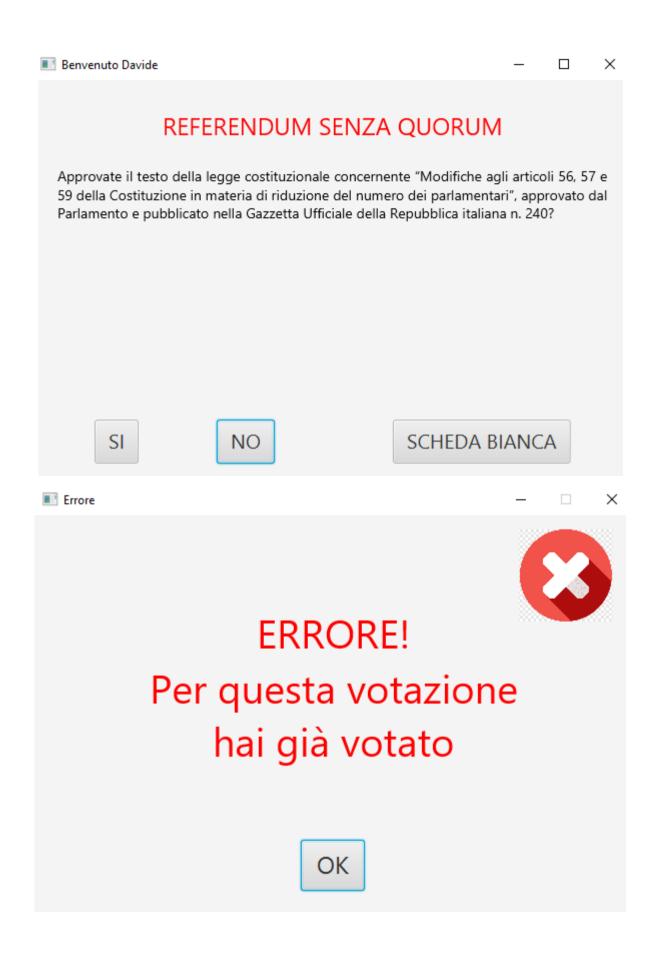


Nel caso ci si autentichi come elettore, questa è la pagina che viene mostrata:

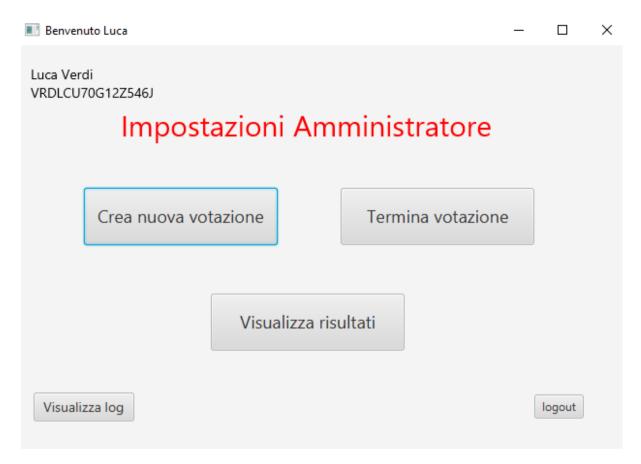


Sono presenti in alto le informazioni dell'utente, mentre nel centro vengono mostrate le votazioni attive al momento. Di fianco a ciascuna sono presenti due button etichettati rispettivamente con "vota", per votare, ed "info", per avere una schermata riepilogativa della votazione. Inoltre, è presente un tasto per il logout.

Premendo su "vota", viene mostrata la scheda elettorale, nella quale le opzioni di voto sono presenti sottoforma di tasti. Se l'elettore aveva già votato per quella votazione, viene mostrata una schermata con un messaggio che dice all'utente che non può più votare.

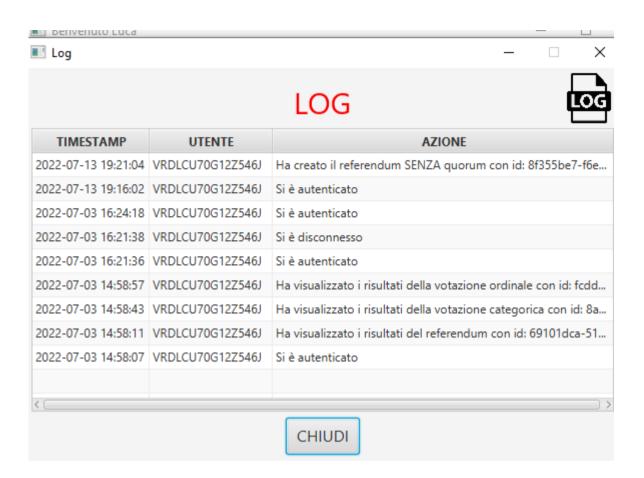


Se si decide di autenticarsi come amministratore, la pagina che si presenta è questa:



Sono presenti dei button per selezionare l'azione che si intende compiere.

Inoltre, è presente un tasto "visualizza log" per avere accesso ad una schermata con la cronologia delle azioni compiute dall'amministratore.



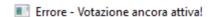
Se si intende creare una votazione viene mostrata la schermata per configurarne una.



Se si intende terminare una votazione, viene mostrata una pagina con l'elenco delle votazioni da terminare.



Infine, se si vogliono visualizzare i risultati di una votazione, viene mostrata una pagina con l'elenco delle votazioni. Premendo "visualizza", se la votazione è terminata si vedono i risultati, altrimenti viene mostrato un messaggio di errore.







×

# Impossibile visualizzare i risultati!

La votazione corrente è ancora attiva. Per visualizzarne i risultati è necessario che sia terminata.

OK

Risultati referendum

## - 🗆



# **REFERENDUM**

TIPO: Referendum CON quorum

TESTO:

abababababababababa

SI: 0 NO: 1

SCHEDA BIANCA: 0



## 3.5 Specifica e verifica dei vincoli

Sono stati specificati i seguenti vincoli OCL, ciascuno dei quali è stato mappato nel codice, tramite un commento JML.

#### Per Elettore:

```
//nome elettore diverso da null
{context Elettore::getName () : string
       post: result != null and result.size() > 0
}
//cognome elettore diverso da null
{context Elettore::getSurname() : string
       post: result != null and result.size() > 0
}
//Codice fiscale deve essere di 16 cifre e diverso da null
{context Elettore::getTaxCode() : string
       post: result != null and result.size() = 16
}
//Età deve essere valida
{context Elettore::getEta() : Integer
       pre: self.birthDay > 0 and self.birthDay <= 31 and self.birthMonth > 0 and
       self.birthMonth <= 12 and self.birthyear>0
}
Per amministratore:
// nome amministratore deve essere diverso da null
{context Amministratore::getName(): string
       post: result != null and result.size() > 0
}
//cognome amministratore deve essere diverso da null
{context Amministratore::getSurname() : String
       post: result != null and result.size() > 0
}
```

## 3.6 Descrizione del testing

## Test classe Utente, mappatura con requisito utente 5:

testLogInTrue() : dato un utente all'interno del database testa se effettivamente il login per quell'utente va a buon fine.

testLogInFalse(): testa che per un utente non presente nel database effettivamente il login non vada a buon fine.

testUsernameNotNull(): verifica che lo username di un utente non sia null.

testPasswordNotNull(): verifica che la password di un utente non sia null.

#### **Test classe Amministratore:**

testTaxCodeLengthTrue(): testa che un codice fiscale appartenente ad un amministratore sia effettivamente non sia diverso da

testTaxCodeLengthFalse(): testa che un codice fiscale appartenente ad un amministratore sia effettivamente di 16 cifre

testTaxCodeNotNull(): testa che il codice fiscale dell'amministratore non sia null.

testNameNotNull(): verifica che il nome di un amministratore non sia null.

testSurnameNotNull(): verifica che il cognome di un amministratore non sia null.

#### **Test classe Elettore:**

testDateOfBirthValid(): verifica che la data di nascita sia valida.

testIsOfAge(): verifica che l'età dell'elettore sia almeno 18 anni.

testNameNotNull(): testa che il nome dell'elettore non sia null.

testSurnameNotNull(): testa che il cognome dell'elettore non sia null.

testLeapYear2020(): verifica che il 2020 sia bisestile.

testLeapYear1999(): verifica che il 1999 non sia bisestile.

#### **Test classe Candidato:**

testPartitoNull(): verifica che il partito di un candidato non sia null.

testEqualsFalse(): testa che due candidati con lo stesso nome e cognome, ma diversi partiti siano effettivamente due candidati diversi.

### 3.7 Note per l'installazione e l'utilizzo

Il primo passo è importare il dump del database chiamato "sweng.sql".

Per eseguire l'applicazione è sufficiente poi importare il progetto in un qualsiasi ide.

Se si usa intellij cliccare con tasto destro sul file pom.xml e cliccare su "aggiungi come maven project", assicurarsi di avere una versione di java che sia almeno la 11.

Infine, eseguire il metodo main della classe "DaoPatternDemo".

Per loggarsi come elettore si hanno due opzioni: usare username "Davide" e password "db" oppure username "Andrea" con password "ab"; mentre per loggarsi come amministratore usare username: "Luca" e password: "lv".