

**Progetto Centri Vaccinali**

Laboratorio Interdisciplinare B

**Portale Cittadini**

**Centro Vaccinale**

**Manuale Tecnico**

Preparato da

**Daniel Satriano Mat. 745232**

**Claudio Menegotto Mat. 745394**

**Cristian De Nicola Mat. 744954**

**Francesco Cavallini Mat. 746933**

**Sommario**

[**INTRODUZIONE** 3](#_Toc96888163)

[Librerie esterne utilizzate 3](#_Toc96888164)

[Struttura generale delle classi 4](#_Toc96888165)

[**CORE CLASSES** 5](#_Toc96888167)

[**Classi enumerative** 8](#_Toc96888168)

# **INTRODUZIONE**

1. **Portale cittadini** è un progetto iniziato nell’ambito del progetto di Laboratorio A e continuato a sviluppare per il progetto di Laboratorio B, per il corso di laurea in Informatica dell’Università degli Studi dell’Insubria.

Il progetto è sviluppato in Java 12, usa un’interfaccia grafica costruita con OpenJFx 12 ed è stato sviluppato e testato sul sistema operativo Windows 10 e Windows 11.

**NOTA**: se si vuole cambiare l’indirizzo IP della connessione al server (tramite il .bat), bisognerà modificare il .bat con un semplice editor di testo e aggiungere come argomenti:

* **1**: indirizzo IP nuovo;
* **2**: numero di porta.

## **Librerie esterne utilizzate**

1.1 L’applicazione fa uso di 3 librerie:

* **OpenJFx 12:** La libreria OpenJFx 12 contiene tutti gli elementi per lo sviluppo dell’interfaccia grafica. L’utilizzo di questa libreria si è reso necessario poiché JavaFx non è più incluso nel Java Development Kit di Oracle dalla release di Java 9.
* **Jfoenix 9.0.10:** La libreria Jfoenix è una open source di java che permette di implementare (graficamente) “Google Material Design”. Usato assieme a scene builder.



## **Struttura generale delle classi**

1.2 Il progetto è strutturato fondamentalmente in 2 rami: le core classes + classi enumerative e le classi adibite alla gestione dell’interfaccia grafica [Figura 1]

Figura 0‑1

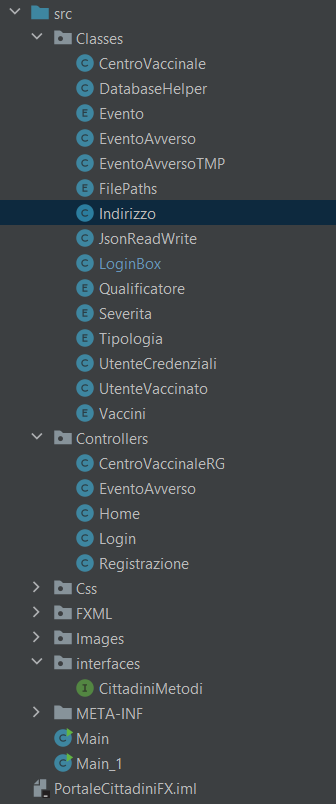
* **Classi “core”:**

Figura 1

* + CentroVaccinale
  + DatabaseHelper
  + EventoAvverso
  + Indirizzo
  + LoginBox
  + UtenteVaccinato
  + UtenteCredenziali
* **Classi enumerative:**
  + Evento
  + FilePaths
  + Qualificatore
  + Severita
  + Tipologia
  + Vaccini
* **Controllers:**
  + CentroVaccinaleRG
  + EventoAvverso
  + Home
  + Login
  + Registrazione
* **Rispettivi FXML:**
  + CentroVaccinaleRG
  + EventoAvversoForm
  + Home
  + Login
  + Registrazione
* **Interfaccia:**
  + CittadiniMetodi

# **CORE CLASSES**

* 1. Verranno presentate ora le core classes nel dettaglio.

(N.B -> la maggior parte della classi presenta un override del toString e un metodo costruttore che non andremo a citare)

**CentroVaccinale:**

* 1. La classe **Centro Vaccinale** identifica i centri vaccinali e le loro funzioni.

Più in particolare presenta al suo interno quattro tag per l’identificazione:

1. LinkedList<Short> **IDVaccinazioni** : è una lista di ID delle vaccinazioni effettuate nel centro, in modo da poterne tenere traccia.
2. String **Nome** : indica il nome del centro vaccinale.
3. [Indirizzo](#Indirizzo) **Indirizzo** : va ad indicare l’indirizzo stradale del centro.
4. [Tipologia](#Tipologia) **Tipologia** : indica la tipologia del centro.

La classe inoltre presenta inoltre un metodo:

* + **getNome**() : che restituisce il nome del centro vaccinale.

**DatabaseHelper:**

* 1. La classe **DatabaseHelper** viene utilizzata per la comunicazione con il server RMI.

Implementa i metodi dell’interfaccia CittadiniMetodi. La classe presenta al suo interno due tipi di costruttore:

* + DatabaseHelper(): utilizza parametri default (messi come variabili globali) impostati nella classe; verranno utilizzati per creare la connessione con il server RMI, rispettivamente **8080** per la porta e “**localhost**” per l’indirizzo.
  + DatabaseHelper(int port, String address): costruttore che permette di scegliere una porta e un indirizzo diversi da quelli dati di default.

**EventoAvverso:**

* 1. La classe **Evento Avverso**  identifica e classifica gli eventi avversi che sono capitati agli utenti a cui è stato sottoministrato il vaccino.

Presenta al suo interno quattro tag, i quali verranno adesso elencati:

1. [Evento](#Evento) **evento** : va ad identificare il tipo di evento che è capitato all’utente.
2. [Severita](#Severità) **severità** : indica, come descritto dal nome, la severità dell’evento subito dall’utente.
3. Short **IDVaccinazione** : serve per tenere traccia della vaccinazione somministrata all’utente che sta riportando l’evento avverso.
4. String **noteOpzionali** : come da nome, sono note opzionali che l’utente può aggiungere prima di pubblicare l’evento avverso.

**Indirizzo:**

* 1. La classe **indirizzo** serve come classe complementare alla classe [CentroVaccinale](#CentroVaccinale), in quanto contiene al suo interno tutte le informazioni riguardanti l’indirizzo stradale del centro. Al suo interno possiamo trovare sei tag che la caratterizzano:

1. [Qualificatore](#Qualificatore) **qualificatore** : va ad indicare il qualificatore dell’indirizzo stradale.
2. String **nome** : nome della strada a cui fa riferimento l’indirizzo.
3. Int **numeroCivico** : numero civico della strada ove situato il centro.
4. String **comune** : comune dove è situato il centro.
5. String **provincia** : provincia dove è situato il centro.
6. Int **cap** : codice di avviamento postale del comune.

**LoginBox:**

1. La classe **LoginBox** è adibita al login. Questa classe gestisce il login e logout tramite due metodi: rispettivamente login() e logout():
2. login (String email, String psw, String nomeCentro): vengono passati come parametri:

* email: la mail con cui l’utente si è registrato al centro vaccinale;
* psw: la password con cui l’utente si è registrato al centro vaccinale;
* nomeCentro: il nome del centro a cui l’utente si è registrato.

1. Logout (): metodo usato per effettuare il logout dal sistema.

**UtenteVaccinato:**

1. La classe **Utente Vaccinato** serve a salvare le informazioni degli utenti vaccinati di tutti i centri registrati nell’applicativo. Al suo interno presenta 8 tag:
2. String **nomeCentroVaccinale** : nome del centro.
3. String **nome** : nome dell’utente vaccinato.
4. String **cognome** : cognome dell’utente vaccinato.
5. String **codiceFiscale**: codice fiscale dell’utente vaccinato.
6. String **dataSomministrazione** : data somministrazione del vaccino.
7. [Vaccini](#Vaccini) **vaccino** : tipo di vaccino effettuato.
8. Short **idVaccinazione** : id della vaccinazione.
9. [EventoAvverso](#EventoAvverso) **evento** : \*\*

La classe presenta tre metodi:

* + **getIdVaccinazione**() : restituisce l’idVaccinazione dell’utente.
  + **getInformation**() : restituisce tutte le informazioni dell’oggetto sotto forma di stringa in UPPER case.
  + **getDataSomministrazione**() : restituisce la data di somministrazione.

**UtenteCredenziali:**

1. La classe **Utente Credenziali** serve a salvare le informazioni degli utenti che effettuano la registrazione. Al suo interno presenta 4 tag:
2. **UserID:**
3. **IDvaccinazione:** id usato per legare l’utente registrato con i resto del database.
4. **IndirizzoEmail:** indirizzo mail dell’utente usato in fase di login per accedere al sistema.
5. **Password:** password scelta dall’utente per accedere al sistema.

**Classi enumerative**

Vengono presentate ora le classi enumerative usate nel progetto.

* **Evento:**

Classe enumerativa utilizzata nella gestione degli eventi avversi, in particolare possiamo trovare al suo interno i seguenti tag:

1. Mal di testa
2. febbre,
3. dolori\_muscolari\_e\_articolari
4. infodonopatia
5. tachicardia
6. crisi\_ipertensiva
7. altro

* **Qualificatore:**

Classe enumarativa utilizzata per indicare i diversi tipi di qualificatori di un indirizzo. Possiamo trovare all’interno della classe i seguenti tag:

1. Via
2. Viale
3. Piazza
4. Corso.

* **Severita:**

Enum utilizzata per la gestione della severità. Al suo interno troviamo i seguenti tag:

1. Molto\_bassa\_1
2. bassa\_2
3. fastidiosa\_3
4. sopportabile\_4
5. insopportabile\_

* **Tipologia:**

Enum utilizzato per la definizione della tipologia di un centro vaccinale. Al suo interno troviamo i seguenti tag:

1. Aziendale
2. Ospedaliero
3. Hub.

* **Vaccini:**

Enum utilizzato per la definizione dei diversi vaccini disponibili in commercio. Al suo interno abbiamo i seguenti tag:

1. Pfizer
2. AstraZeneca
3. Moderna
4. JeJ