

# **APPLICAZIONE SISTEMA VACCINI**

## **Manuale tecnico**

**Università degli Studi dell'Insubria – Laurea Triennale in Informatica**

**Progetto Laboratorio B: Applicazione per prenotazione vaccini**

**Sviluppato da: Lorenzo Bruni, Francesco Clary, Anna Lutsyshyna, Alessio Panarese**

## **SOMMARIO**

|  |        |
|--|--------|
| INTRODUZIONE                             | PAG. 3 |
| LIBRERIE USATE                           | PAG. 3 |
| PACKAGE                                  | PAG.3  |
| DESCRIZIONE CLASSI                       | PAG. 4 |
| PACKAGE COM.EXAMPLE                      | PAG. 4 |
| PACKAGE COM.EXAMPLE.MODELS               | PAG. 5 |
| LIMITI SOFTWARE E IMPLEMENTAZIONI FUTURE | PAG.7  |

## INTRODUZIONE

La nostra applicazione cerca di simulare, realisticamente, un sistema complesso di prenotazione dei vaccini per il Covid-19, di ricerca degli hub vaccinali e di inserimento degli eventi avversi relativi al vaccino. Il progetto è sviluppato in Java 16, usa un'interfaccia grafica costruita con JavaFX ed è stato sviluppato e testato sui sistemi operativi Windows 10 e 11.

Il tutto è gestito da un server, all'interno di esso è presente un database dove sono salvati tutti i dati.

## LIBRERIE USATE

- **Java SQL**  
Fornisce metodi per implementare un database con tutti i vantaggi del linguaggio SQL su un programma Java.
- **Java RMI**  
Permette di avere metodi utili per creare un'applicazione distribuita in Java.
- **Java Util**  
Contiene vari metodi come, ad esempio, la generazione casuale di una stringa.
- **Java IO/NIO**  
Forniscono tutti i metodi necessari input e output di dati.
- **Java FX**  
Contiene tutti gli elementi per lo sviluppo dell'interfaccia grafica come ad esempio slider, bottoni e frame collegandosi ad un file fxml.
- **Java Net**  
Presenta metodi che permettono di creare un collegamento fra server e client.

## PACKAGE

### **com.example**

Server

IstanzaServer

InterfacciaServer

App

HubIniziale

SceltaCittadino

RicercaCentri

RegistrationFormCittadino

EventiAvversi

LoginOperatori

SceltaOperatori

RegistrationFormCentroVaccinale

RegistrationFormVaccinato

### **com.example.models**

Cittadino

CittadinoRegistrato

CittadinoVaccinato

EventoAvverso

LoginCentro

CentroVaccinale

CentroVaccinaleSimple

Indirizzo

Qualificatore

SigleProvince

TipoCentro

TipoVaccino

All'interno del package **com.example** sono presenti tutti i file fxml per far sì che la grafica creata con la libreria Java con l'ausilio del programma SceneBuilder, sia collegata alle classi.

### DESCRIZIONE CLASSI

#### PACKAGE COM.EXAMPLE

##### **Server**

Tale classe mette a disposizione dei client le operazioni da effettuare sul database.

I metodi principali sono *ApriConnessioneDB()* e *ChiudiConnessioneDB()* che permettono di aprire e chiudere la connessione con il database.

I metodi *getCentriVaccinaliByName()*, *getCentriVaccinaliByType()*, *getCittadiniVaccinati()*, *getCittadiniVaccinatild()*, *getDatiLogin()*, *getCittadiniRegistrati()*, *getCentriVaccinali()* e *getEventiAvversi()* servono a recuperare i dati registrati al sistema dal database.

Invece i metodi *setCentroVaccinale()*, *setVaccinato()*, *registraCittadino()* e *registraEventiAvversi()* inseriscono nella tabella dinamica, creata dal metodo *createTableDinamica()*, i dati inseriti dall'utente.

### **InterfacciaServer**

Interfaccia del Server.

### **IstanzaServer**

Istanza del server collegata all'interfaccia.

### **App**

Applicazione client vera e propria che una volta lanciata permetterà di avviare il programma facendo partire la schermata d'inizio.

### **HubIniziale**

Classe che ci permette di collegare le schermate di SceltaCittadino e LoginOperatori.

### **SceltaCittadino**

Classe che permette al cittadino di effettuare il login o fare il redirect sulle pagine di ricerca centri o registrazione

### **RicercaCentri**

Classe che permette al cittadino di effettuare una ricerca dei centri vaccinali sia per nome del centro che per tipologia ma anche per comune o provincia.

I metodi principali sono *cercaCentri()* che legge i dati opportuni in base al valore dello slider ed effettua la ricerca dei centri tramite chiamata al server infine popola l'ObservableList per permettere la visualizzazione sulla tabella e *initialize()* che è subito eseguito all'inizializzazione della classe che imposta un listener sullo slider cambiando modalità di ricerca (per nome o comune/tipologia) all'occorrenza.

### **RegistrationFormCittadino**

Classe che permette al cittadino di registrarsi al sistema in modo da poter segnalare eventi avversi dovuti al vaccino.

### **EventiAvversi**

Classe che permette al cittadino di effettuare l'inserimento di eventi avversi.

### **LoginOperatori**

Classe per operazioni di login da parte dell'operatore sanitario.

## SceltaOperatore

Classe per scegliere operazione di registrazione centro o registrazione vaccinato.

## RegistrationFormCentroVaccinale

Classe che specifica il modulo di registrazione del centro vaccinale per inserirlo nel sistema.

## RegistrationFormVaccinato

Classe che specifica il modulo di registrazione del vaccinato ad un dato centro.

**N.B.** Sono presenti anche dei metodi *backToHub()* che permettono all'utente di tornare indietro di schermata premendo un semplice bottone.

```
PACKAGE COM.EXAMPLE.MODELS
```

## Cittadino

Class che restituisce i metodi get and set per il Cittadino, inoltre all'interno è presente il costruttore della classe ovvero *Cittadino(String nome, String cognome, String codiceFiscale, String idVaccinazione)*.

## CittadinoRegistrato

Classe che restituisce i metodi get and set per il Cittadino Registrato ed è presente anche il costruttore della classe ovvero *CittadinoRegistrato(String nome, String cognome, String codiceFiscale, String idVaccinazione)*.

## CittadinoVaccinato

Classe che restituisce i metodi get and set per il Cittadino Vaccinato. Oltre a ciò, è presente il costruttore *CittadinoVaccinato(String nome, String cognome, String codiceFiscale, String idVaccinazione)*.

## EventoAvverso

Classe che restituisce i metodi get and set per gli eventi avversi inoltre è presente il costruttore *EventoAvverso(String evento, int severità, String noteOpzionali)*.

## LoginCentro

Classe che restituisce i metodi get and set per la parte dei login Operatori. Inoltre è presente il costruttore *loginCentro(String user, String password)*.

## **CentroVaccinale**

Class che restituisce i metodi get and set per il CentroVaccinale inoltre è presente il costruttore *CentroVaccinale(String nome, Indirizzo indirizzo, TipoCentro tipoCentro)*.

## **CentroVaccinaleSimple**

Classe che restituisce i metodi get and set per la visualizzazione dei dati dei centri nella table dedicata. È presente anche il costruttore *CentroVaccinaleSimple(String nome, String qualificatore, String indirizzo, Integer numCivico, String comune, String provincia, Integer cap, String tipologia, Integer eventi, Double media)*.

## **Indirizzo**

Class che restituisce i metodi get and set per la parte dell'Indirizzo è presente, inoltre, il costruttore della classe ovvero *Indirizzo(Qualificatore qualificatore, String nome, String numeroCivico, String comune, String provincia, String cap)*.

## **Qualificatore**

Enum che restituisce nella UI il qualificatore dell'indirizzo.

## **SigleProvince**

Enumerativo che restituisce nella UI la sigla della Provincia

## **TipoCentro**

Enumerativo che restituisce nella UI il tipo di Centro.

## **TipoVaccino**

Enumerativo che restituisce nella UI il tipo di Vaccino.

## **LIMITI SOFTWARE E IMPLEMENTAZIONI FUTURE**

L'applicazione presenta alcune problematiche che verranno sistemate nelle prossime versioni, tali problemi sono:

- Logo dell'applicazione non presente
- Controlli sulle date
- Tabella migliorata
- Cancellazione della prenotazione
- Aumento capacità memoria del server