



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Laboratorio Interdisciplinare B

Specifiche di Progetto

Davide Tosi

Dipartimento di Scienze Teoriche e Applicate

davide.tosi@uninsubria.it



Laboratorio Interdisciplinare - Cos'è:

- **Obiettivi:** mettere a fattor comune e in pratica le conoscenze acquisite nei corsi di Progettazione, Sviluppo Concorrente e Distribuito, e Basi Dati
- **Come:** progettando e sviluppando un progetto software reale

Il Progetto

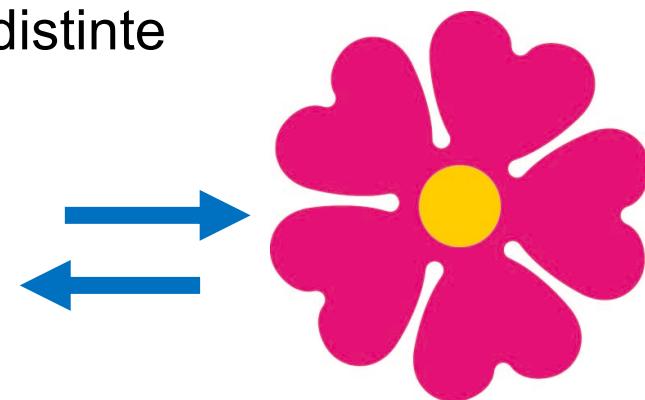
- Un sistema di segnalazione eventi avversi da vaccinazione covid19
 - Eventi Avversi numerici (scala da 1 a 5 stelle con 5 = severità massima)
 - Note opzionali testuali (max 256 caratteri liberi)

Evento	Severità (da 1 a 5)	Note opzionali (256 caratteri)
mal di testa	<<n>>	
febbre	<<n>>	
dolori muscolari e articolari	<<n>>	
linfoadenopatia	<<n>>	
tachicardia	<<n>>	
crisi ipertensiva	<<n>>	

- Il sistema è costituito da due applicazioni distinte
 - Centri Vaccinali
 - Cittadini



Cittadini



Centri vaccinali



Il Progetto – Centri vaccinali



- L'applicazione permette di:
 1. registrare a sistema un centro vaccinale, tramite la funzione `registraCentroVaccinale()`
 2. registrare a sistema, per un dato centro vaccinale, il cittadino vaccinato tramite la funzione `registraVaccinato()`
- Per registrare un centro vaccinale è necessario inserire:
 - nome del centro vaccinale
 - indirizzo del centro (qualificatore via/v.le/pizza, nome, numero civico, comune, sigla provincia, cap)
 - tipologia (ospedaliero, aziendale, hub)
- I dati di ogni centro vaccinale sono memorizzati su DB in una Tabella denominata `CentriVaccinali`



Il Progetto – Centri vaccinali



- Per registrare un cittadino dopo la vaccinazione, tramite la funzione `registraVaccinato()`, inserire:
 - nome centro vaccinale
 - nome e cognome del cittadino
 - codice fiscale
 - data somministrazione vaccino (formato: gg/mm/aaaa)
 - vaccino somministrato (Pfizer, AstraZeneca, Moderna, J&J)
 - id univoco vaccinazione (id numerico su 16 bit)
- I dati di ogni cittadino vaccinato sono memorizzati su DB in una Tabella denominata `Vaccinati_NomeCentroVaccinale`

dove
`NomeCentroVaccinale` deve essere sostituito dinamicamente dal nome del centro vaccinale

Il Progetto – Cittadini



- L'applicazione permette di:
 1. consultare le informazioni di ogni centro vaccinale (accesso libero)
 2. registrarsi al centro vaccinale di propria competenza
 3. inserire eventuali effetti collaterali post vaccinazione (dopo login)
- Per consultare le informazioni di ogni centro vaccinale
(non è necessario login o registrazione cittadino)
 - a. funzionalità di ricerca `cercaCentroVaccinale()`
 - ricerca per nome del centro (prende in input una stringa di caratteri e restituisce i centri vaccinali nel cui nome compare la stringa di caratteri)
 - ricerca per comune e tipologia (prende in input un comune e una tipologia e restituisce il nome del/i centro/i vaccinale/i nel comune per ogni tipologia)
 - b. funzionalità di selezione e visualizzazione `visualizzaInfoCentroVaccinale()`
 - una volta ricercato il centro, deve essere possibile selezionare il centro di interesse e visualizzare tutte le informazioni relative al centro selezionato.
Inoltre, deve essere riportato un prospetto riassuntivo degli eventi avversi segnalati dai cittadini (in forma anonima) per quel centro (riportando il numero di segnalazioni e la severità media)

Il Progetto – Cittadini



- Per registrarsi presso un centro vaccinale tramite la funzione `registraCittadino()` è necessario inserire:
 - nome e cognome
 - codice fiscale
 - indirizzo di posta elettronica
 - userid
 - password per accedere al sistema
 - id univoco vaccinazione (fornito al momento della vaccinazione)
- I dati della registrazione devono essere salvati su DB in una Tabella denominata `Cittadini_Registrati`

Il Progetto – Cittadini



- Per inserire eventuali eventi avversi post vaccinazione, il cittadino:
 1. deve autenticarsi tramite userid e password fornite al momento della registrazione
 2. ricercare e selezionare il proprio centro vaccinale dove è stata eseguita vaccinazione
 3. può ora usare la funzione `inserisciEventiAvversi()`
(gli eventi avversi sono memorizzati su DB)

Evento	Severità (da 1 a 5)	Note opzionali (256 caratteri)
mal di testa	<<n>>	
febbre	<<n>>	
dolori muscolari e articolari	<<n>>	
linfoadenopatia	<<n>>	
tachicardia	<<n>>	
crisi ipertensiva	<<n>>	
.... .	<<n>>	



Il Progetto – Home



All'avvio l'applicazione mostra un menu iniziale dove:

- gli operatori vaccinali possono
 - registrare un nuovo centro vaccinale
 - registrare un nuovo vaccinato
- i cittadini possono invece
 - cercare e visualizzare le informazioni di un centro vaccinale
 - registrarsi presso un centro vaccinale
 - inserire segnalazioni di eventi avversi alla vaccinazione



Il Progetto – Piattaforma CV



- La Piattaforma CentriVaccinali (CV) consiste di:
 1. un modulo `serverCV`, che interfacciandosi con un DBMS relazionale (PostgreSQL) fornisce servizi di back-end;
 2. un modulo `clientCV`, che fornisce tutti i servizi e funzionalità designate per gli utilizzatori dell'applicazione.

Al lancio di `serverCV` deve essere richiesto di specificare:

1. le credenziali per accedere al `dbCV` (database di supporto all'esecuzione dei servizi della piattaforma CV)
2. l'host del DB

Una volta lanciato `serverCV`, questo dovrà rimanere in attesa di richieste di connessione da parte di client `clientCV`

L'applicazione CV dovrà essere realizzata in modo tale da supportare l'interazione in parallelo con più utenti connessi alla piattaforma da postazioni (client) differenti.



Il Progetto – Aspetti Chiave



- Strutture dati per i Centri Vaccinali
 - Info Centri Vaccinali
 - Eventi Avversi
- Strutture dati per i Cittadini
- Cosa e come memorizzare su DB (e relativa progettazione)
- Interfaccia utente con interfaccia grafica
- Gestione della Concorrenza
(i servizi dell'applicazione vengono erogati in parallelo a più utenti, e possono verificarsi accessi concorrenti a risorse condivise)



Il Progetto – ToDo



1. Progettazione della Soluzione
2. Sviluppo della Soluzione Software
3. Documentazione di Progetto (2 documenti distinti)*
 - Manuale Utente
 - Manuale Tecnico

* Ulteriori informazioni su come scrivere la documentazione le vedremo la prossima lezione



Il Progetto – ToDo



1. Progettazione della Soluzione

Le attività di analisi e progettazione devono essere adeguatamente documentate facendo uso del **linguaggio UML** per l'applicazione software e del modello **Entity-Relationship (ER)** per il database.

È richiesto di progettare l'applicazione avvalendosi dove possibile dell'uso di **design patterns**, e di realizzare l'applicazione con un'opportuna **interfaccia grafica**, usando il linguaggio Java e gli strumenti utili

Progettare e realizzare un database utilizzando **PostgreSQL** per la sua implementazione (<http://www.postgresql.org>) e **JDBC** per l'accesso alla base di dati da programma Java (<http://jdbc.postgresql.org/download.html>)



Il Progetto – ToDo



2. Sviluppo della Soluzione – Database

- Si ristrutturi, se necessario, secondo le metodologie di progettazione i requisiti descritti. Si scelgano le metodologie per la costruzione dello schema ER, motivando le scelte fatte.
- Si definisca lo schema concettuale ER per il database, evidenziando le entità e le associazioni di interesse, nonché i vincoli di cardinalità e di identificazione, motivando le scelte effettuate. Altri eventuali vincoli devono essere espressi in linguaggio naturale.
- Si effettui la ristrutturazione dello schema ER motivando le scelte effettuate.
- E' richiesto di produrre un documento di analisi dei requisiti ristrutturato e documentazione associata allo schema ER (ristrutturato e non), con eventuale specifica di vincoli in linguaggio naturale.
- Si effettui la traduzione dello schema ER ristrutturato in un equivalente schema relazionale. E' richiesto di produrre la documentazione associata allo schema relazionale derivato dallo schema concettuale.
- Si realizzi il database utilizzando PostgreSQL, e SQL per la definizione dei dati, l'implementazione dei vincoli identificati, e la manipolazione dei dati, secondo le operazioni previste dall'applicazione.
- Documentare gli script SQL necessari alla creazione della base di dati e dei vincoli definiti sui dati e le query SQL a supporto dei servizi erogati da IP.



Il Progetto – ToDo



2. Sviluppo della Soluzione – Codice Sorgente

- Il progetto deve essere sviluppato in linguaggio **Java**
- Il codice deve essere opportunamente commentato in formato **javadoc**
- I package `centrivaccinali` e `cittadini` devono essere definiti e devono contenere le relative classi (ulteriori package sono ammessi)
- Il `main` per l'esecuzione dell'applicazione deve essere contenuto nella classe di nome `CentriVaccinali` del package `centrivaccinali`
- L'intestazione di tutti i file `*.java` devono contenere *nome, cognome, num. matricola, sede (VA o CO)* degli autori del progetto



Il Progetto – ToDo



3. Documentazione di Progetto

(Seguirà lezione ad hoc su come scrivere un buon documento utente e tecnico)





Il Progetto – Consegnna



1. Il progetto deve essere consegnato in formato compresso denominato *cognome_matricola* del project manager del team

2. La cartella compressa dovrà contenere:
 - un file dal nome **autori.txt** contenente cognome, nome, numero di matricola e sede (Va o CO) di ogni membro del team
 - la directory **doc** contenente il manuale utente, il manuale tecnico in formato .pdf, e tutti gli artefatti (diagrammi ER, UML) prodotti
 - la directory **src** contenente il codice sorgente del progetto
 - file di build Apache Ant (<http://ant.apache.org/>) o Maven (<https://maven.apache.org/>) per compilare il progetto, lanciare il server e i client, creare il database, creare la documentazione javadoc, etc.
 - eventuali librerie necessarie alla compilazione e/o all'esecuzione
 - file README con indicazioni precise sull'installazione e sulla compilazione, specificando i comandi Ant/Maven da utilizzare, ed indicazioni di particolari librerie, usate in modo non standard.



Il Progetto – Consegnna



3. La consegna viene fatta tramite invio cartella compressa all'indirizzo davide.tosi@uninsubria.it con oggetto email: **Progetto Lab B**
4. Responsabile della consegna sarà il project manager tramite email istituzionale di ateneo
5. Le date di consegna verranno comunicate di volta in volta (indicativamente una decina di giorni prima della data di appello d'esame)



Il Progetto – Valutazione



- In fase di discussione orale verrà:
 - richiesto allo studente di saper argomentare in modo opportuno le scelte progettuali, algoritmiche, e implementative adottate
 - verificata l'effettiva padronanza delle tecniche utilizzate attraverso una serie di domande
- La valutazione terrà conto dei seguenti fattori:
 - l'aderenza del sistema realizzato ai requisiti proposti
 - la qualità dei documenti di analisi e progettazione prodotti sia per la realizzazione del software che per il database (correttezza sintattica, semantica, completezza e leggibilità, minimalità dello schema logico)
 - le scelte algoritmiche e di progettazione effettuate (design pattern)
 - la qualità del codice sorgente prodotto (funzionalità, correttezza, facilità d'uso).