**ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA SCUOLA DI INGEGNERIA E ARCHITETTURA (SEDE DI BOLOGNA)**

**CORSO DI LAUREA IN INGEGNERIA INFORMATICA(T)**

**RELAZIONE DI FINE TIROCINIO CURRICULARE DI VITTORIO CHIUSELLI**

**------------------------------------------------------------------------**

Svolto presso

**NSI, via Bonazzi, 2**

**Java/JavaFX, test automatici e continuos integration**

Studente:

Vittorio

Chiuselli

Tutore Universitario: Prof. Marco Patella

Referente Struttura Ospitante:

Stefano Borghi

***Indice***

***-Introduzione***

1. ***JavaFX***
2. ***Database***

***2.1 ORM***

***2.2 Hibernate***

1. ***Maven***
2. ***Conclusioni***
3. **Introduzione**

Il tirocinio è stato svolto presso l’azienda NSI (Castelmaggiore in via Bonazzi, 2), la quale si occupa di progettare software, realizzare soluzioni informatiche e consulenza informatica.

Sotto la supervisione di Alessandra Cerioni sono stato inserito nel progetto di realizzazione di un programma aziendale interno di rimborsi spese, come sviluppatore. Durante il tirocinio ho

sviluppato l’applicazione con l’IDE IntelliJ, dovendo studiare il linguaggio di programmazione JavaFX, ,studiare Hibernate, Maven, JUnit, ORM e HSQLDB.

1. **Java FX**

E’ stato usato JavaFX , come linguaggio di programmazione, perché abbiamo ritenuto che sia un ottimo strumento per poter sfruttare il cambiamento istantaneo degli oggetti a livello di interfaccia,

così che al momento dell’utilizzo dell’applicazione, ogni oggetto che viene cambiato viene subito mostrato.

JavaFX è linguaggio JAVA che garantisce una flessibile creazione di applicazioni web e Rich Internet Application (applicazioni con caratteristiche e funzionalità di applicazioni desktop).

Ho studiato sul libro alcuni argomenti per approfondire la conoscenza dell’utilizzo di questo linguaggio.

Quando si lancia una applicazione JavaFX, ha un ciclo di vita che esegue, in ordine:

-costruttori senza argomenti

-metodo init()

-metodo start()

-metodo stop()

Le proprietà fondamentali di javaFX sono:

1. binding

2. proprietà

3. collezioni osservabili

4. changer notification

**1.** Il binding lega due oggetti per sincronizzarli tra loro al momento di una modifica, occorre chiaramente creare una notifica al cambiamento.

Può essere unidirezionale, per cui il cambiamento si propaga solo in una direzione, o bidirezionale, in cui entrambi dipendono tra loro, per cui abbiamo che x=y e y=x.

**2.** Le proprietà ci permettono di osservare i valori e vedere che cambiano (e ci cambiano il pojo tradizionale):



Mentre il pojo tradizionale era cosi:



Binding può essere Lazy (al momento della computazione controlla se le proprietà sono cambiate) o Eager (in ogni momento si notifica il cambiamento delle proprietà).

A default tutte le proprietà sono lazy.

**3.** L’osservabilità, ovvero, quando un oggetto cambia, viene notificato a tutti gli ascoltatori.

**4.** Changer notification ci dice quando una proprietà cambia, sta a noi a identificare quali avvertire e cosa fare al momento della notifica di cambiamento

1. **Database**

Come database per l’applicazione, abbiamo scelto HSQLDB perché si poteva integrare in maniera semplice dentro l’applicazione, e perché non deve gestire una grande quantità di dati.

HyperSQL DataBase (embedded) è scritto interamente in Java, e ci sono tre modalità di memorizzazione:

-In memory: persistenza assente, database allocato in RAM.

Il formato dell’URL è jdbc:hsqldb:mem:NODENAME

-Standalone engine: utilizza i file senza consumo di risorse di rete. Programma ha un accesso diretto al database. Impedisce l’interferimento di tool esterni quando viene eseguito.

-Server Engine e Webserver engine: massimo dell’accessibilità.

Si può dire che questa applicazione è standalone, ovvero è capace di funzionare da sola e senza essere installata in un sistema operativo.

***2.1 ORM***

Durante la fase di progettazione del programma, abbiamo scelto di usare un ORM perché, in coscienza dell’utilizzo di molte operazioni CRUD e di vari campi non banali da riempire e creare come tabelle, un ORM avrebbe facilitato la parte di database.

In informatica, l’ORM, è una tecnica di programmazione per convertire dati tra sistemi di dati incompatibili.

Il principio dell’ORM è quello di delegare a degli strumenti la gestione della persistenza e di lavorare a livello di codice con oggetti rappresentanti il modello (MVC Model).

Tali strumenti instaurano un collegamento bidirezionale tra il database e gli oggetti dell’applicazione.

Esso garantisce che ci sia persistenza dei dati e garantisce uno strato sopra al database, ciò garantisce una solidità di progetto, perché si può cambiare il database sotto e l’ORM non ne risente del cambiamento (occorre cambiare un paio di righe di codice).

Un buon ORM deve offrire:

-Flessibilità (deve essere adattabile a diversi ambienti).

-Facilità di utilizzo.

-Supporto per idiomi OOP (ereditarietà, polimorfismo,etc..).

L’ORM si colloca sotto il livello logico di business.

***2.2 Hibernate***

Hibernate è un software open source e distribuito sotto licenza LGPL. Hibernate fornisce un framework semplice da usare mappa un modello di dominio orientato agli oggetti in un classico database relazionale. Tramite annotazioni (metodo bottom up), ci permette di trasformare un oggetto in una tabella nel proprio

database. A seguire, un frammento di codice dove si può vedere le annotazioni che permetteranno al framework di creare la tabella attraverso l’oggetto:



Hibernate non si occupa solo di mappare dalle classi Java in tabelle del database ma fornisce anche degli aiuti per le query di dati, il recupero di informazioni e riduce significativamente il tempo di scrittura che altrimenti sarebbe speso lavorando manualmente in SQL e con JDBC.

Infatti si pone l’obiettivo di snellire significativamente la scrittura del codice relativo allo strato di persistenza. Elimina la necessità di scrivere codice SQL per effettuare le operazioni CRUD (Create, Retrieve, Update, elete). Fornisce soluzione ai problemi ricorrenti di programmazione (generazione id, gestione delle versioni).

Hibernate si colloca uno strato sotto l’ORM, ma sempre sopra al database, come si può vedere nella figura 1.

Per utilizzare hibernate occorre avere delle librerie.

(E’ possibile utilizzare anche un metodo top down)

E' stata usata questa tecnologia per tutte le ragioni che ho sopra citato, perchè rispetto gli altri servizi di ORM, è produttivo, mantenibile, portabile, gratuito, facile da imparare,

popolare (per ogni problema potevo trovare aiuto con una ricerca di Google) e sviluppo dell'applicazione veramente facile e rapido.

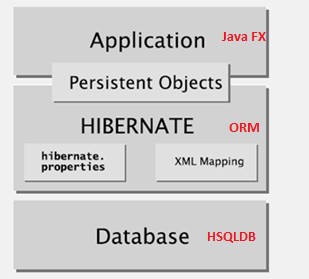


Figura 1

1. **Maven**

È un framework che ho installato nel mio IDE che aiuta a gestire meglio il progetto, in termini di documentazione, distribuzione, compilazione del codice e ci da ulteriori funzionalità:

-Creare uno standard di compilazione e di struttura del progetto

-Test automatizzati

-Gestione e download delle librerie necessarie al progetto

-Ampliare le funzionalità installando plugin

I principali componenti di Maven sono:

-pom.xml: contiene tutte le informazioni sul progetto (dipendenze, test, documentazione)

-Goal: singola funzione che può essere eseguita sul progetto

-Plugin: goal riutilizzabili in più progetti

-Plugin built-in tra cui: clean, compiler,deploy, instal…

-Repository è una directory strutturata destinata alla gestione delle librerie.

L'azienda mi ha chiesto di utilizzare questo framework perchè rispetto ad Ant, le dipendenze vengono scaricate automaticamente, effettua lavori di compilazione, impacchettamento, test,

pulizia, in maniera automatica, e non manuale (come in Ant e MakeFile)

1. **Conclusioni**

Il lavoro svolto nel corso del tirocinio è stato molto istruttivo, sia da un punto di vista conoscitivo, che lavorativo.

Grazie alla supervisione di Alessandra Cerioni sono venuto a conoscenza di un modo di lavorare e l’utilizzo di strumenti all’avanguardia per gli informatici.

Ho avuto modo di imparare un nuovo linguaggio (non studiato durante il corso della triennale), ovvero JavaFX e seguire in azienda un corso di formazione che mi ha aperto la strada nel mondo delle applicazioni web