Per prima cosa ho creato un nuovo progetto maven da linea di comando andando a specificare:

(dopo essermi spostato utilizzando il comando cd in C:\Users\soprasteria\Desktop\Repository\uiip37Repo)

mvnarchetype:generate -DgroupId=it.Carlo.Soprasteria.Curricula -DartifactId= Curricula.CarloDeGennaro -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

Pertanto ho specificato:

* come groupId: it.Carlo.Soprasteria.Curricula
* come ArtifactID: Curricula.CarloDeGennaro

Dopodichè dalla root del nostro progetto, ovvero da dove è presente il file pom.xml, ovvero dalla cartella Curricula.CarloDeGennaro eseguiamo il comando:

* mvn package

ed il comando:

* mvn eclipse:eclipse

Dopodichè vado ad importare il progetto dalla Spring tool Suite(STS)

(file-import-existing projects into workspace – browse e seleziono la cartella Curricula.CarloDeGennaro)

Modifico il file pom.xml aggiungendo le dipendenze per mysql, che permetteranno appunto la connessione al db, ovvero:

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.44</version>

</dependency>

Dopodiché implemento una classe ConnectionManager all’interno del package it.Carlo.soprasteria.Curricula che si occuperà della connessione con il db.

Nello stesso package metto anche la funzione main chiamata “App.java” in cui definirò l’acquisizione dei dati e la visualizzazione di questi tramite standard input e standard output.

A questo punto ho creato tre package, in particolare:

1. it.Carlo.soprasteria.Curricula.model

in cui si metteranno le classi che rappresentano le tabelle di un RDBMS;

la [classe](https://it.wikipedia.org/wiki/Classe_(informatica)) con i relativi [metodi](https://it.wikipedia.org/wiki/Metodo_(informatica)) ed attributi (POJO(Plain Old Java Object)) rappresenta un'[entità tabellare](https://it.wikipedia.org/wiki/Modello_relazionale) di un [RDBMS](https://it.wikipedia.org/wiki/RDBMS);

1. it.Carlo.soprasteria.Curricula.dao

in cui metterò le interfacce in cui saranno definite le funzioni che voglio siano implementate.

Il DataAccessObject (DAO) astrae l’implementazione per l’accesso ai dati sottostanti per consentire al Client un accesso trasparente alla sorgente dati. Il DAO implementa le operazioni CRUD (Create, Read, Update, Delete).

Definiamo l'interfaccia per accedere ai dati ad essa associati a livello di persistenza;

1. it.Carlo.soprasteria.Curricula.impl

Qui verrà creata una classe per ogni interfaccia definita nel package 2 che ne implementa i metodi.

A questo punto creo un nuovo branch da develop.

Quindi per prima cosa se sono in un branch diverso da develop mi sposto nel branch develop con il comando:

* + git checkout develop

creo un nuovo branch con il comando:

* + git checkout –b EsFinaleCurriculaCarlo

e aggiorno anche il server con il nuovo branch con il comando:

* + git push –u origin EsFinaleCurriculaCarlo

dopodichè vado ad aggiungere i miei dati dalla working directory alla staging area(index) con il comando:

* + git add .

dopodichè porto i file dalla staging area nell’HEAD con il comando:

* + git commit –m ‘Esercizio4\_curricula’;

vado a fare il push dei dati con il comando:

* + git push origin EsFinaleCurriculaCarlo

infine faccio la poll request