# Curso de Java Fullstack Completo (springboot, hibernate, JWT, API Rest)

<https://youtu.be/7vHzVN0EiQc>

<https://youtu.be/7vHzVN0EiQc?t=8354>

Sacar una cuenta en Google Cloud gratuita por un año.

# Instalar IDE

Moy usa IntelliJ Idea[[1]](#footnote-1), en la versión Community, que tiene algunas limitaciones. La versión paga es más completa.

Yo prefiero usar Visual Studio Code.

# Instalar Maven

Instalar Maven como aplicación independiente.

Descargar el instalador desde:

<https://maven.apache.org/download.cgi>

Descomprimir. Copiar todo el contenido del directorio en el destino. En mi caso, usé

C:\Program Files\Maven

No le pongo la versión, para no tener que corregir la variable de entorno cada vez que actualizo.

En las variables de entorno, agrego:

MAVEN\_HOME con el valor C:\Program Files\Maven

Edito la variable path y le agrego:

%MAVEN\_HOME%\bin

Observar el uso de la variable %MAVEN\_HOME%

Abrir una consola de Windows. Escribir:

mvn -v

# Instalar Java

Instala Java como aplicación independiente.

Descargar los instaladores del JDK desde las siguientes páginas:

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>

<https://www.oracle.com/ar/java/technologies/javase/javase8u211-later-archive-downloads.html>

Para los tutoriales podemos necesitar las versiones anteriores del JDK. Es conveniente tenerlas instaladas, aunque solo una de ellas será la que esté activa en un dado momento.

En las variables de entorno, configurar

JAVA\_HOME con el valor C:\Program Files\Java\jdk-11.0.13

en mi caso particular.

Editar la variable path para incluir:

%JAVA\_HOME%\bin

Notar el uso de los signos % en la variable. Esto es conveniente cuando queremos cambiar la versión del JDK activa.

# Configurar Java Runtime en VS Code

ctrl + shift + p

Preferences: Open Setings (JSON)

Se abre settings.json. La parte que nos interesa es:

  "java.configuration.runtimes": [

    {

      "name": "JavaSE-11",

      "path": "C:\\Program Files\\Java\\jdk-11.0.13",

      "default": true

    },

    {

      "name": "JavaSE-17",

      "path": "C:\\Program Files\\Java\\jdk-17.0.2",

      "default": false

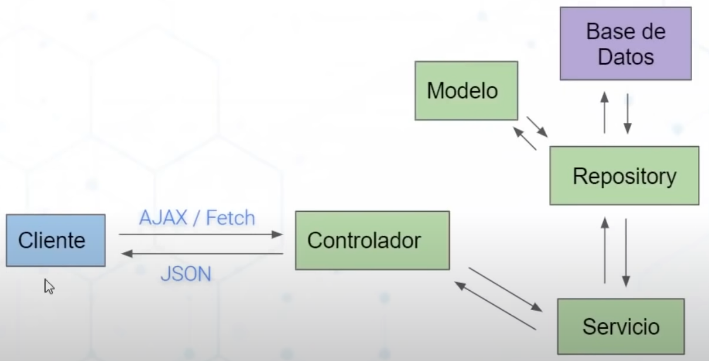
    }

  ],

Debe ser algo de este estilo. Hay que adaptarlo a la computadora de cada uno.

# Patrón de diseño MVC

# Arquitectura REST



REST

# WampServer

Moy usa Xampp. Yo prefiero WampServer.

En PhpMyAdmin crear la base de datos con el siguiente código:

# Inicializar el proyecto en Spring Boot

Ya sea con <https://start.spring.io/> o con la extensión de VS Code.

Maven

Java

2.6.4

tup

moy

War

11

## Dependencias

Spring Web

# Clases

Hay dos clases: MoyApplication y ServletInitializer. La primera es ya conocida. La segunda aparece porque elegimos packaging war. Se suele decir que este es el packging apropiado para una aplicación web. Pero realmente, también se puede desplegar una aplicación web en un jar.

# En pom.xml

Veremos que aparece la dependencia:

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-tomcat</artifactId>

            <scope>provided</scope>

        </dependency>

que no vemos cuando el packaging es jar. Esto debería aparecer automáticamente, pero si no lo hace, lo podemos agregar manualmente.

# Front end

Para el front end no vamos a usar un framework como Angular o React, aunque se podría. Vamos a usar el siguiente template:

<https://startbootstrap.com/theme/sb-admin-2>

El archivo ya está descargado en la misma carpeta en la que está este documento.

# Configurar acceso a la base de datos

<https://www.javadevjournal.com/spring-boot/spring-boot-mysql/>

En pom.xml agregar estas dos dependencias:

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

            <version>2.6.4</version>

        </dependency>

        <dependency>

            <groupId>mysql</groupId>

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

            <version>8.0.28</version>

            <scope>runtime</scope>

        </dependency>

En application.properties agregar lo siguiente:

spring.datasource.url=jdbc:mysql://localhost:3306/cursojava

spring.datasource.username=root

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

## Hibernate Properties

# The SQL dialect makes Hibernate generate better SQL for the chosen database

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8InnoDBDialect

# Spring Transaction Maven Dependency

<https://www.sourcecodeexamples.net/2019/12/spring-tx-maven-dependency.html>

Copy below Spring Transaction maven dependency and paste in your project pom.xml file:

<!-- https://mvnrepository.com/artifact/org.springframework/spring-tx -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>5.3.17</version>

</dependency>

<https://en.wikipedia.org/wiki/Jakarta_Transactions#@Transactional_annotation>

El artículo de la Wikipedia es lo único más o menos corto y correcto que encontré. Lo demás es o muy largo o conceptualmente deficiente.

# Agregar en pom-xml

        <dependency>

            <groupId>mysql</groupId>

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

            <version>8.0.28</version>

            <scope>runtime</scope>

        </dependency>

# Agregar en pom-xml

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

        </dependency>

1. <https://www.jetbrains.com/idea/download/#section=windows> [↑](#footnote-ref-1)