**Trabajo Fin de Bootcamp**

**DIEGOPEDIA**

**Características del proyecto**

**Autor**: Diego Ayuso Del Campo

**DNI**: 05721486-Y

**Curso**: Bootcamp FullStack Developer

**Centro de formación**: The Bridge

**Empresa**: Accenture

**Fecha de comienzo de desarrollo**: 30/03/2023

**Fecha de finalización de desarrollo**: 23/04/2023

**Índice**

**Páginas 3-4**

**Páginas 3-5**

**Páginas 3-5**

**0. Definición del proyecto\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**0. Introducción\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**0.1. Requisitos mínimos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**0. Introducción**

Se confeccionará una aplicación que permita visualizar información a cerca de algunos científicos y de algunas de sus contribuciones. La aplicación será visible y accesible por el usuario mediante un front que se realizará con html, css y javascript mediante bootstrap.

El contenido de este documento es de carácter explicativo y recoge el procedimiento de desarrollo seguido. En primer lugar, se comentarán los requisitos mínimos suplidos.

En segundo lugar se presentará la información para mantener el proyecto, datos de acceso, configuraciones y ciertas directrices que se han seguido como premisas para estructurar y definir cada proyecto.

En tercer lugar, se comentará brevemente la estructura de cada subproyecto de trabajo que compone el proyecto en su conjunto con algunas especificaciones a tener en cuenta según el tipo de cliente.

En cuarto lugar, se ampliarán los recursos (base de datos usuarios, usuarios eliminados

**0.1. Requisitos mínimos**

Los requisitos que se suplirán en primera instancia vienen enumerados a continuación en orden creciente de carga de trabajo y catalogados según el nombre de la carpeta de trabajo en los entornos de desarrollo:

-ApiRestDiegoPedia: este proyecto será de tipo maven spring boot. El front consumirá esta api. Se devolverá una ResponseEntity para el resultado con la información solicitada en el request o con el error pertinente. El front mostrará esa response según el diseño establecido.

-SpringMVCDiegoPedia: este proyecto será de tipo maven spring boot. Mediante este proyecto se realizará el mantenimiento de la base de datos. Para ello se usará el recurso thymeleaf.

\*Dentro del paquete principal, existirán tres paquetes: \*.Controllers (con los controladores @Controller Cientifico y Contribucion), \*.Models (con las entidades de Hibernate Cientifico y Contribucion) y \*.Repositories (con los repositorios).

\*En los templates, se han añadido: carpeta fragmentos con footer.html, header.html, headerCientifico.html, headerContribucion.html; científicos.html; contribuciones.html; diegopedia.html, la vista de inicio; eliminarCientifico.html; eliminarContribucion.html; guardarCientifico.html; guardarContribucion.html.

\*En la carpeta static, se han añadido dos ficheros de recursos javascript: recursosBtnClear.js y recursosGuardarCientifico.js.

\*En el fichero application.properties se han definido los pares propiedad-valor como se muestra a continuación.

spring.jpa.hibernate.dcl-auto=none

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://mysql-dominioproyectosbbdd.alwaysdata.net/dominioproyectosbbdd\_mibasedatos

spring.datasource.username=305548\_diegoacce

spring.datasource.password=305548\_usuarioAdminBBDD

server.port=9097

spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

spring.thymeleaf.enabled=true

spring.thymeleaf.encoding=UTF-8

spring.datasource.hikari.data-source-properties.useUnicode=true

spring.datasource.hikari.data-source-properties.characterEncoding=UTF-8

spring.jpa.properties.hibernate.connection.characterEncoding=UTF-8

spring.jpa.properties.hibernate.connection.CharSet=UTF-8

spring.jpa.properties.hibernate.connection.useUnicode=true

default-character-set = utf8mb4

Ciertos comentarios adicionales a tener en cuenta son los siguientes:

+La propiedad spring.jpa.hibernate.naming.physical-strategy se ha definido al valor que indica que el nombre de los campos sigue la directriz CamelCase en lugar de \_Case. Este valor es, según la documentación consultada: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl.

spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

\*En el fichero pom, se han definido las dependencias relativas a spring-core, spring-boot-starter-thymeleaf, spring-boot-starter-web, spring-boot-starter-data-jpa, spring-boot-starter-test, spring-boot-starter, spring-data-commons, spring-data-jpa, junit y mysql-connector-java:

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.0.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.diego</groupId>

<artifactId>SpringMVCDiegoPedia</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SpringMVCDiegoPedia</name>

<description>Proyecto crud con thymeleaf</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>javax.persistence</groupId>

<artifactId>javax.persistence-api</artifactId>

<version>2.2</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>org.springframework.data</groupId>

<artifactId>spring-data-commons</artifactId>

<version>3.0.3</version>

</dependency>

<dependency>

<groupId>org.springframework.data</groupId>

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.32</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

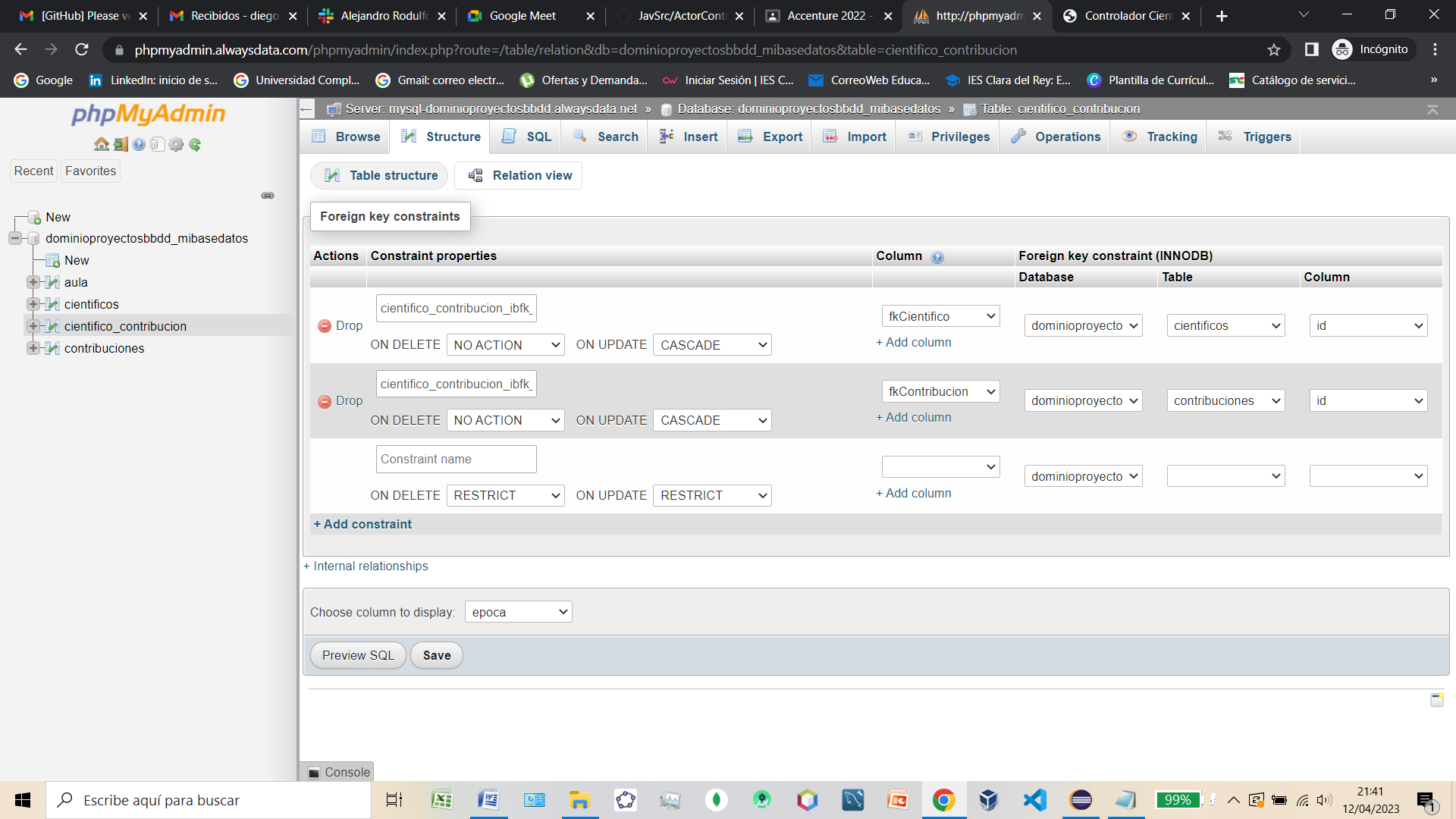
</build>

</project>

-Base de datos: la base de datos estará almacenada en un servidor de phpmyadmin. Inicialmente, tendrá 2 tablas: científicos y contribuciones.

\*La tabla científicos tendrá los campos: id (primary Auto increment), nombre (varchar(50), apellidos (varchar(50), nacionalidad (varchar(50), fechaNacimiento (varchar(10), fechaDefuncion (admite null) (varchar(10), enlaceMasInformacion (url de wikipedia) (varchar(300), enlaceFoto (url del jpg de wikipedia) (varchar(300).

\*La tabla contribuciones tendrá los campos: id (primary Auto increment), nombre (varchar(100)), campo (varchar(60)), rama (varchar(60)), enlaceMasInformacion (varchar(300)).

\*La tabla de la relación cientifico\_contribucion: id (primary Auto increment), fkCientifico, fkContribucion, epoca (year), aportacion (varchar(60)). Las claves foráneas tendrán restricciones de tipo no action y cascade para on delete y on update respectivamente (**imagen [1]**).

**Imagen [1].** Restricciones de claves foráneas.