

**Desarrollo de Soluciones Cloud**

**(NRC-1534) viernes 19:00 – 22:00**

**Proyecto Ez Learning**

**Profesor** : Jose Santos Ventura Arteaga

**Alumno** : Dominguez Oliva Edwin Donato

Jean Patrick Kerkhoff Urday

Christopher Andrew Roudebush

Cristian Velazquez Ramirez

**2020-I**

**Ez Learning**

Aplicación web de enseñanza virtual desarrollada usando Java 8, Spring Boot, MySql y Materialize.



Esta aplicación empezó como un proyecto académico en agosto del 2019, desarrollada para el curso de Desarrollo de Aplicaciones Empresariales II en Isil, Lima, Perú.

Es una plataforma de enseñanza virtual donde puedes explorar cursos, profesores y registrarte para tomar tantos cursos como desees.

**Casos de Uso**

* Explorar cursos
* Explorar profesores
* Registrarse en la plataforma
* Editar perfil
* Registrarse en cursos
* Acceder a cursos

**Tecnologías**

Utiliza Thymeleaf como el motor de plantillas para el frontend, el cual fue estilizado utilizando Materialize.

El backend está desarrollado con Java 8, utilizando Spring Boot con Spring MVC, Spring JPA y Spring Security como dependencias.

Utiliza 2 perfiles de aplicación, uno para desarrollo y otro para producción. El de desarrollo utiliza una base de datos en memoria H2, mientras que la de producción utiliza MySql. Ambas utilizan versionado de Sql utilizando Flyway.

La aplicación web se encuentra alojada en Heroku, mientras que la base de datos se encuentra alojada en una instancia de AWS RDS.

**Servidor de Aplicaciones**

* Apache Tomcat embed 9.0.24
* Instancia hobby dyno en Heroku
* 512 MB RAM
* 1 procesador

**Base de Datos**

* Instancia AWS RDS
* MySQL 5.7.22
* 1 vCPU
* 1GiB RAM

**Fases del Proyecto**

**Fase 1 (Completada):**

Migrar el proyecto Spring Boot a una instancia EC2 corriendo Java8.

Este proyecto se encuentra alojado en el repositorio <https://github.com/donnatto/ez-learning>

Para esta fase se creó una instancia de EC2 en la región de Virginia del Norte, con Ubuntu Server 18.04 LTS, de tipo t2.micro, 16 GB de SSD, políticas de seguridad HTTP, HTTPS y del puerto 8080 públicas para que las personas puedan acceder a la página sin problemas.

Una vez que la instancia ha sido creada, nos conectamos a esta mediante SSH y actualizamos el sistema usando los comando **sudo apt update** y **sudo apt upgrade**

Se instalo el jdk y jre de Java8 utilizando el comando **sudo apt install openjdk-8-jdk openjdk-8-jre**

Clonamos nuestro proyecto desde github con el comando **git clone https://github.com/donnatto/ez-learning.git**

Instalamos maven para empacar nuestro jar ejecutable con **sudo apt install maven**

Nos dirigimos a la carpeta ez-learning con **cd ez-learning** y ejecutamos **mvn package** para generar el jar.

Ahora nos vamos a la carpeta target que es donde se encuentra ubicado nuestro jar usando **cd target** y ejecutamos nuestro jar con el comando **nohup java -jar platform-0.0.1-SNAPSHOT.jar &**

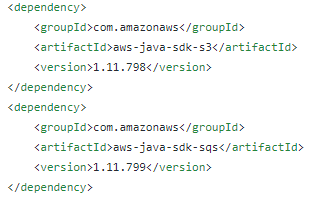
Lo que hará que nuestro jar corra en el background y no se detenga al cerrar la conexión desde nuestro terminal.

Listo. Ahora nuestra página está disponible entrando a <http://ec2-54-159-2-2.compute-1.amazonaws.com:8080/> y también http://54.153.66.176:8080/

**Fase 2:**

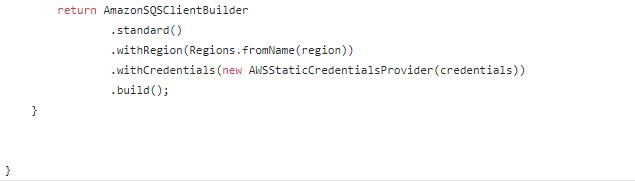
Implementación del bucket de S3 para almacenar todos los archivos estáticos del proyecto y enlazarlos a este.

* Dependencias AWS en el POM

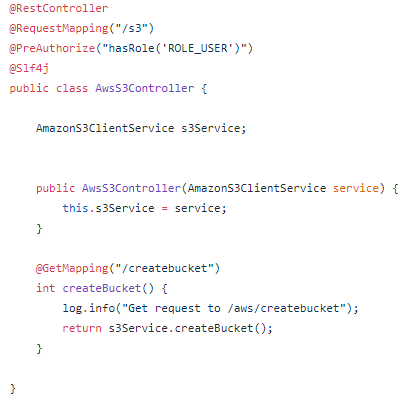


* **Aws Config**

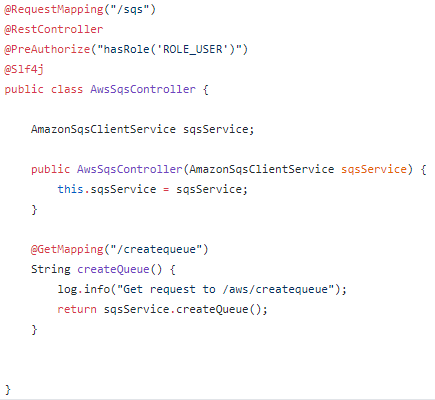




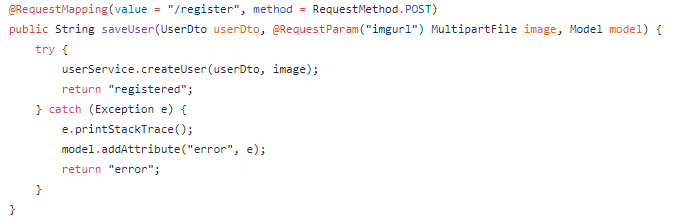
* **AwsS3Controller**



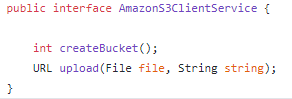
* **AwsSqsController**



* **UserController** (Método saveUser Captura la imagen del usuario)



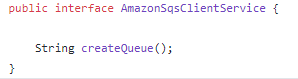
* **AmazonS3ClientService**



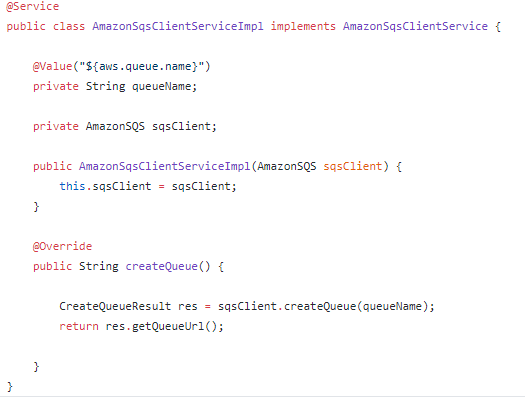
* **AmazonS3ClientServiceImpl**



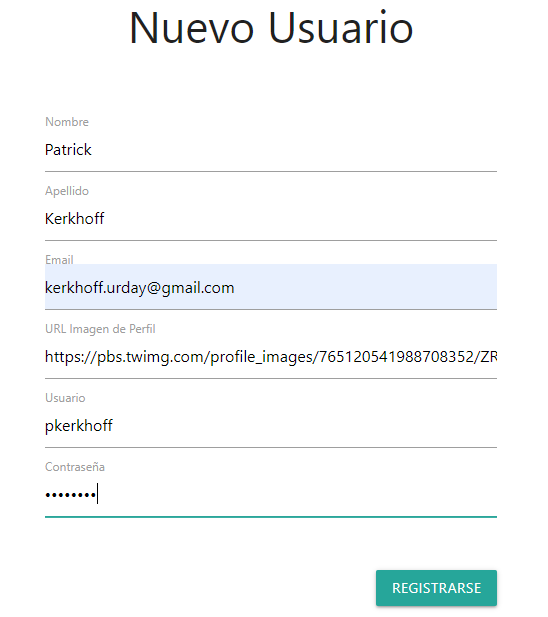
* **AmazonSqsClientService**



* **AmazonSqsClientServiceImpl**

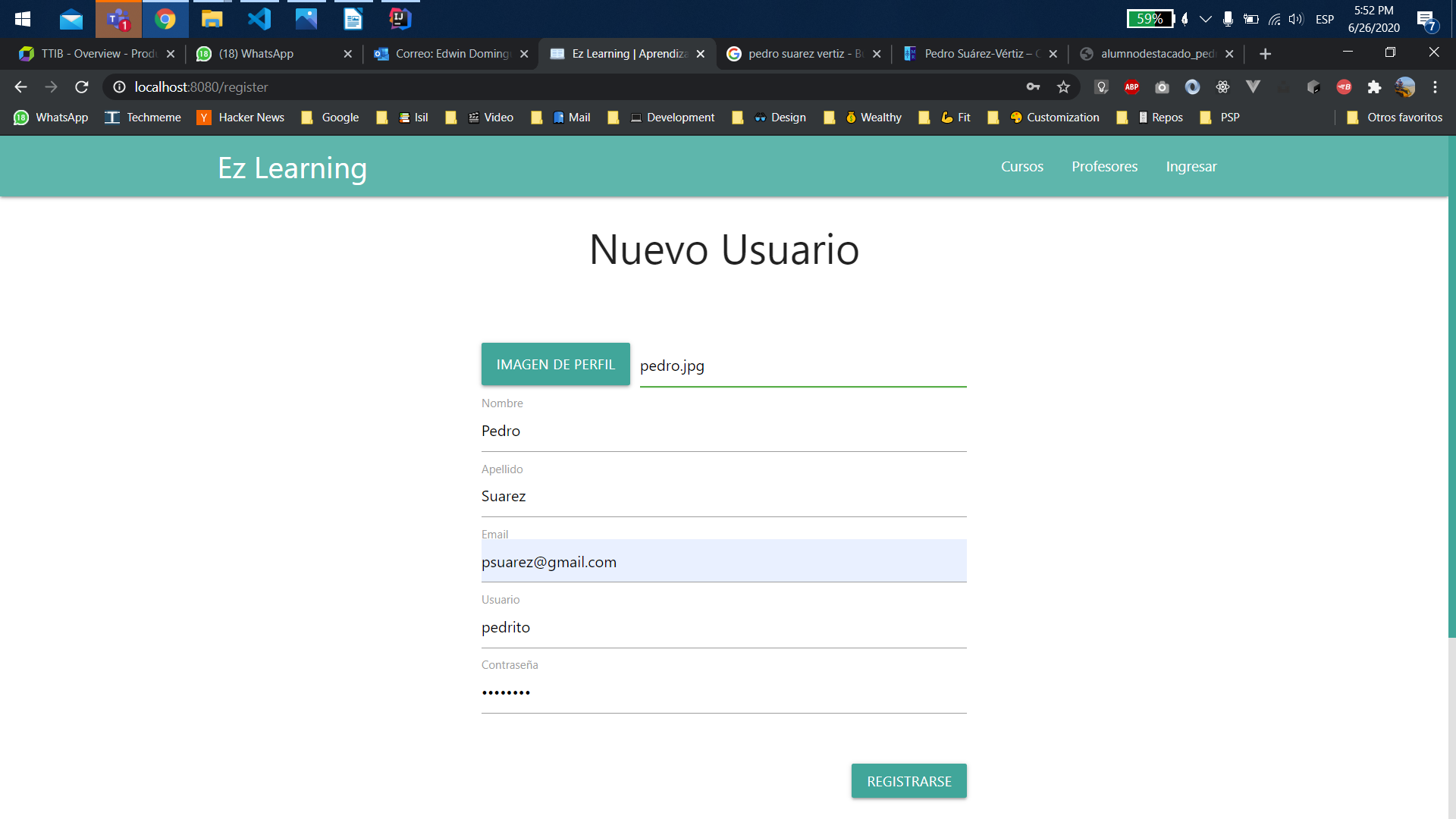
****

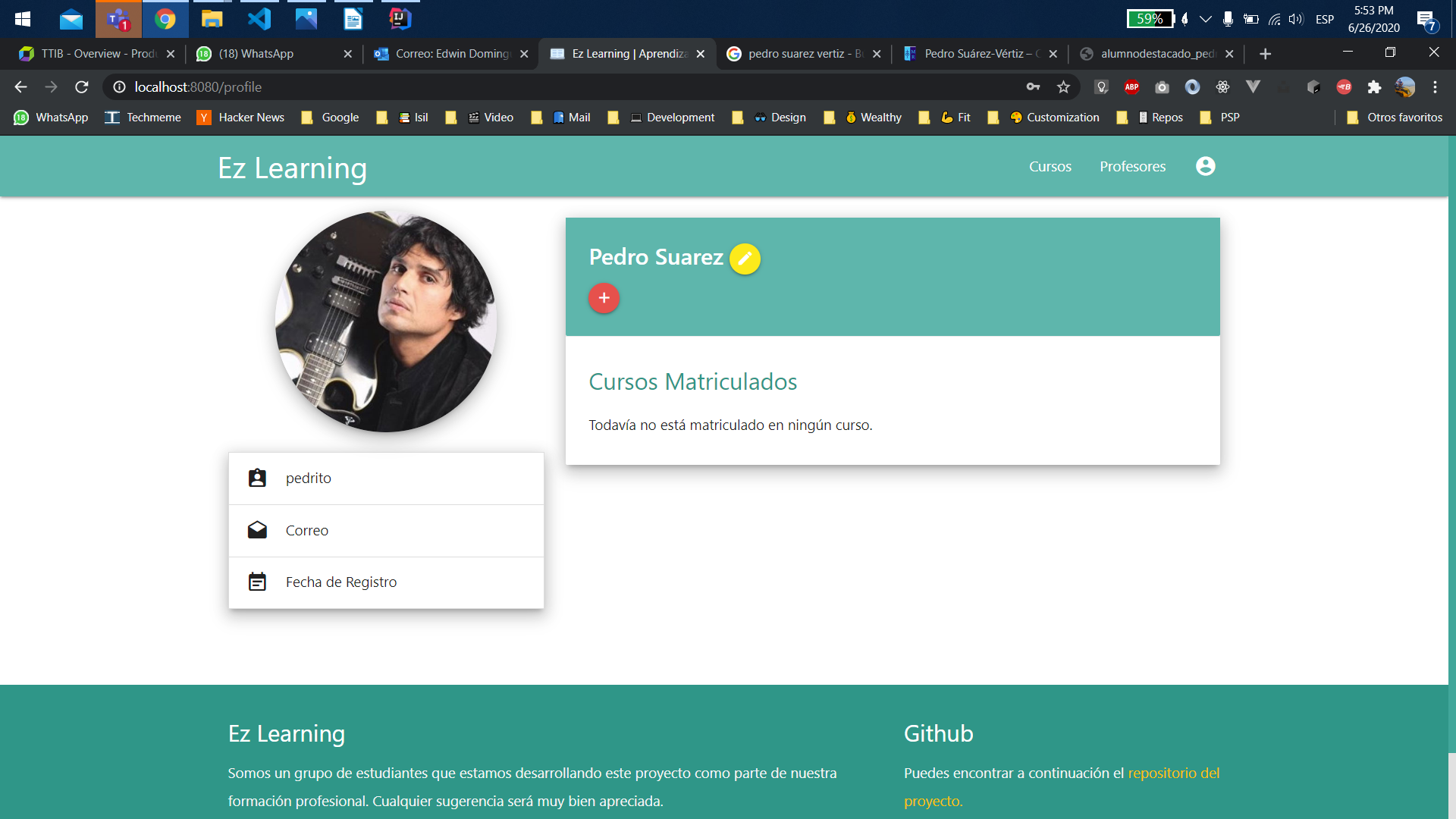
* **Visualización del proceso:**

**Antes. -** Registro de usuario con foto mediante URL, que luego al iniciar sesión y ver los detalles del perfil, la foto se puede visualizar.

**Después**

Registro de usuario con imagen de perfil que se sube al bucket de S3, devuelve la url de esta y se guarda en la BD. Cuando se accede al perfil de usuario se muestra dicha imagen desde el S3.





**Fase 3 (Por implementar):**

Dockerizar el proyecto.