

InfoGym

CFGS DAW

CURSO 2020/2021

NOMBRE/S DEL/DE LOS ALUMNO/S

COLEGIO CALASANZ SALAMANCA



1. Introducción	3
2. Descripción de la aplicación	4
3. Tecnologías escogidas y justificación	5
4. Diseño de la aplicación	6
4.1. Diagramas y definición de casos de uso	6
5. Arquitectura de la aplicación	
6. Manual de despliegue	8



Introducción

El proyecto de fin de grado superior lo vamos a realizar de forma conjunta los alumnos: Santiago Rodríguez Ortiz de Zárate y Rodrigo Pascual Montero. La temática del proyecto consiste en el desarrollo de una aplicación web de un gimnasio en la cual plasmaremos los conocimientos adquiridos en estos dos años de curso. El nombre de nuestra aplicación será "InfoGym", la cual servirá para la gestión de un gimnasio.

Nuestra aplicación es muy intuitiva y con ella se podrá gestionar el gimnasio, para ello se mostrara una tabla con los nombres de los clientes en la cual aparecerá el numero id, el nombre , la edad, la cuota que se le haya cobrado, ya que si ha cogido una oferta el precio podrá cambiar y por último si ha pagado o no.

Además estos datos se podrán modificar en un apartado siempre y en todo momento.

También se puede añadir un nuevo cliente en la pestaña añadir y saldrá un formulario para poner los datos de los clientes.

Por último hay una pestaña con la que podrás eliminar al cliente seleccionado.



2. Tecnologías escogidas y justificación

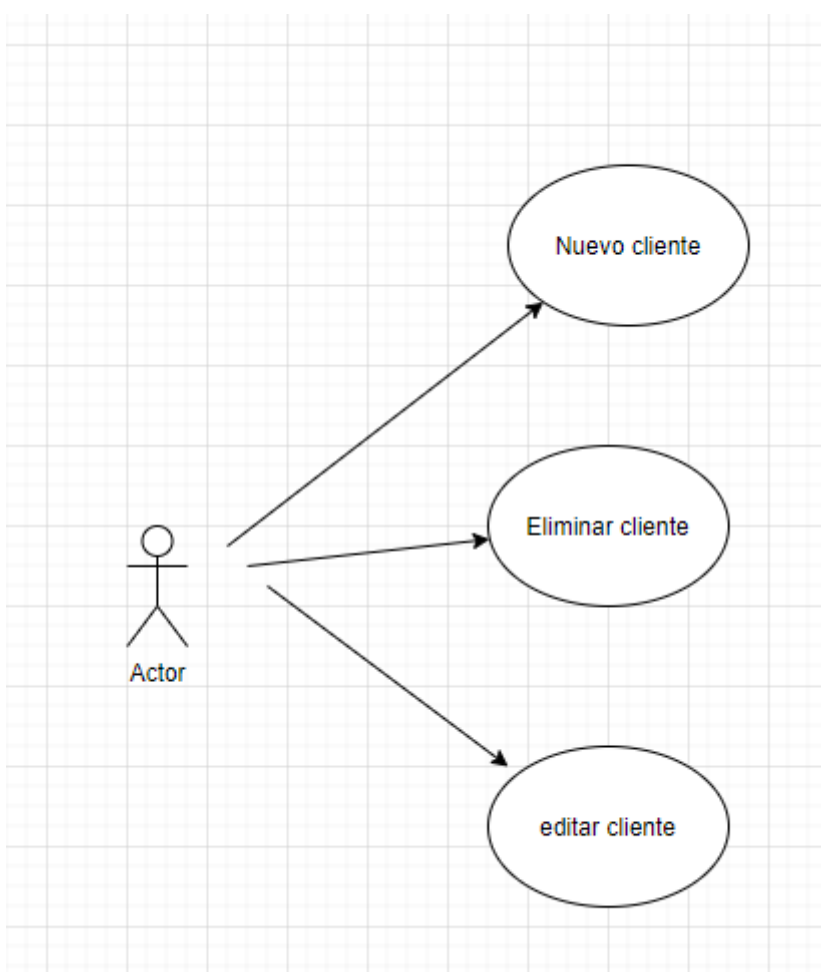
Las tecnologías a utilizar serán aquellas vistas en clase que más nos han gustado.

Para la parte back usaremos el framework Spring ya que nos será de mucha ayuda y es un framework que ya conocemos.

Para la parte visual se utilizará utilizarán jsp donde se utilizará html y css.

3. Diseño de la aplicación

3.1 Diagramas y definición de casos de uso





Definición de casos de uso:

Nombre: nuevo cliente

Precondición: La vista de inicio tiene que cargarse correctamente

Descripción: El usuario podrá crear un nuevo cliente.

Paso 1 : introducir el nombre de usuario.

Paso 2 : introducir edad

Paso 3 : cuota.

Paso 4 : introducir si ha pagado o no.

Postcondición: Se crea un nuevo usuario

Nombre: Registro

Precondición: La vista de registro tiene que cargarse correctamente

Descripción: El usuario podrá modificar un cliente

Paso 1 : Se mostrara el formulario y se podrá modificar cualquier aspecto

Postcondición: El usuario se modifica correctamente

Nombre: eliminar

Precondición: La vista de beneficios tiene que cargarse correctamente

Descripción: El usuario podrá eliminar un cliente de la base de datos

Paso 1 : pulsar el botón apuntarse

Paso 2 : eliminar cliente

Paso 3 : pulsar enviar.

Postcondición se elimina correctamente

Además nuestra aplicación estará formada por una base de datos con una sola tabla donde estarán reflejados los datos del cliente



clientes
id
nombre
edad
couta
cobrado

4.Arquitectura de la aplicación

Nuestro proyecto lo realizaremos con la arquitectura MVC (Modelo Vista Controlador).

Los paquetes controller tendremos los controladores, serían ClientesController y NoClientesController.



```
▼ gymnasio [boot]
  ▼ src/main/java
    ▼ com
      > App.java
    ▼ com.controladores
      > ClienteController.java
    ▼ com.entidades
      > Cliente.java
    ▼ com.repositorios
      > ClienteRepository.java
  ▼ src/main/resources
    application.properties
  > JRE System Library [JavaSE-1.8]
  > Maven Dependencies
  ▼ src
    ▼ main
      ▼ webapp
        ▼ views
          Clienteform.jsp
          Clientes.jsp
  > target
  pom.xml
```




Manual de despliegue

Para el despliegue hemos optado por la opción de utilizar Docker.

Lo que hay que hacer para desplegar nuestra app es lo siguiente:

Creamos una carpeta llamada scriptSQL dentro de la carpeta que tenemos la cual es gim-master y en ella meteremos nuestra base de datos.

A continuación, creamos un archivo dockerfile dentro de esa misma carpeta para la base de datos.

Dentro de nuestra carpeta llamada clientesgym, creamos otro archivo Dockerfile el cual es para el backend.

En esa misma carpeta hacemos un git bash y creamos una imagen con el nombre de rodrigo313/bbdd

(**docker build -t rodrigo313/bbdd .**)

para la imagen de la base de datos, a continuación, creamos una red llamada infogym

(**docker network create infogym**).

Después creamos un contenedor para la base de datos con el nombre de mi_mysql y la contraseña Rodrigo313

```
(docker run -d --network infogym --name mi_mysql -e
MYSQL_ROOT_PASSWORD=Rodrigo313 -e MYSQL_DATABASE=clientes -e
MYSQL_PASSWORD=Rodrigo313 -p 3307:3306 rodrigo313/bbdd).
```

Ya tendríamos nuestro contenedor y nuestra imagen para la base de datos, a continuación vamos a hacer lo propio con el backend. Primero para hacer el backend tendremos que cambiar en el application.properties en el spring.datasource.url la palabra localhost por la de nuestro contenedor el cual es mi_mysql. Una vez hecho eso, haremos un maven install del proyecto para que nos genere un archivo .war. Cogemos ese archivo .war el cual está en la carpeta target y la pegamos en la carpeta de nuestro proyecto, en este caso, una carpeta



llamada clientesgym, una vez hecho esto, hacemos un git bash y utilizamos el siguiente comando para crear una imagen llamada rodrigo313/backend

(**docker build -t rodrigo313/backend .**), a continuación, con el siguiente comando nuestra app ya estaría funcionando con docker

(**docker run --network infogym --name infogym_back -p 8080:8080 rodrigo313/backend**).

Una vez hecho esto ya lo tendríamos todo y solo faltaría poner en el navegador lo siguiente:

<http://localhost:8080/gimnasio/clientes/>