Trabajo de Fin de Grado

Álvaro López Marcos

Anexo III

Tabla de contenido

[Introducción 3](#_Toc112954261)

[2. Diseño de datos 4](#_Toc112954262)

[2.1. Diagrama de base de datos 4](#_Toc112954263)

[3. Diseño arquitectónico 5](#_Toc112954264)

[3.1. Angular 5](#_Toc112954265)

[3.2. Spring 6](#_Toc112954266)

[3.2.1. Services 6](#_Toc112954267)

[6](#_Toc112954268)

[3.1.2. Repositories 7](#_Toc112954269)

[7](#_Toc112954270)

[3.1.3. Entities 8](#_Toc112954271)

[8](#_Toc112954272)

[3.1.4. Controllers 9](#_Toc112954273)

[9](#_Toc112954274)

[3.1.5. Enums 10](#_Toc112954275)

[10](#_Toc112954276)

[3.1.6. Diagrama completo 11](#_Toc112954277)

[4. Diagrama de despliegue 12](#_Toc112954278)

# Introducción

# 2. Diseño de datos

Todos los datos manejados por la aplicación son guardados en el MySQL Server, exceptuando los archivos multimedia que son almacenados en el servidor que contiene la aplicación web, en la base de datos se almacenan los datos necesarios para acceder a dichos archivos.

## Imagen que contiene Escala de tiempo Descripción generada automáticamente2.1. Diagrama de base de datos

La base de datos relacional en la que se almacena la información de la aplicación es definida por el siguiente diagrama:

# 3. Diseño arquitectónico

El proyecto consta de dos aplicaciones que se comunican por medio de peticiones REST. Por un lado, tenemos la aplicación que implementa el FrontEnd, desarrollada en el framework Angular y por el otro lado una aplicación desarrollada en Spring Framkework, que contiene la lógica de nuestra aplicación, comunicada con el Servidor SQL que contiene la base de datos en la que se aloja toda la información de nuestra aplicación. Entre el frontend y el Backend tendremos un balanceador de carga.

## 3.1. Angular

Las aplicaciones Angular se dividen en componentes que se comunican entre sí y se crean unos a otros, en nuestro caso su distribución es la siguiente:

Diagrama

Descripción generada automáticamente

## 3.2. Spring

La aplicación se divide en los siguientes paquetes de clases:

### 3.2.1. Services

### Texto Descripción generada automáticamente

### 3.1.2. Repositories

### Texto, Carta Descripción generada automáticamente

### 3.1.3. Entities

### Imagen que contiene Diagrama Descripción generada automáticamente

### 3.1.4. Controllers

### Texto Descripción generada automáticamente

### 3.1.5. Enums

### Texto Descripción generada automáticamente

### 3.1.6. Diagrama completo

## 4. Diagrama de despliegue

Diagrama

Descripción generada automáticamente