

CICLO FORMATIVO DE GRADO SUPERIOR**DESARROLLO DE APLICACIONES MULTIPLATAFORMA****ACCESO A DATOS - 2º CURSO**

AppEventos

Curso: 2021/22

AUTOR: Mario Parrilla Maroto

Introducción

Motivación y objetivos

He decidido realizar una aplicación de eventos para dar solución a las personas que quieren quedar o pedir cita para un evento o situación, como por ejemplo realizar una reunión, quedar alguien para contratarle...

Descripción de la aplicación

Esta aplicacion, te permitirá controlar los usuarios de la aplicacion movil y poder ver los eventos y sus detalles

Especificación de requisitos

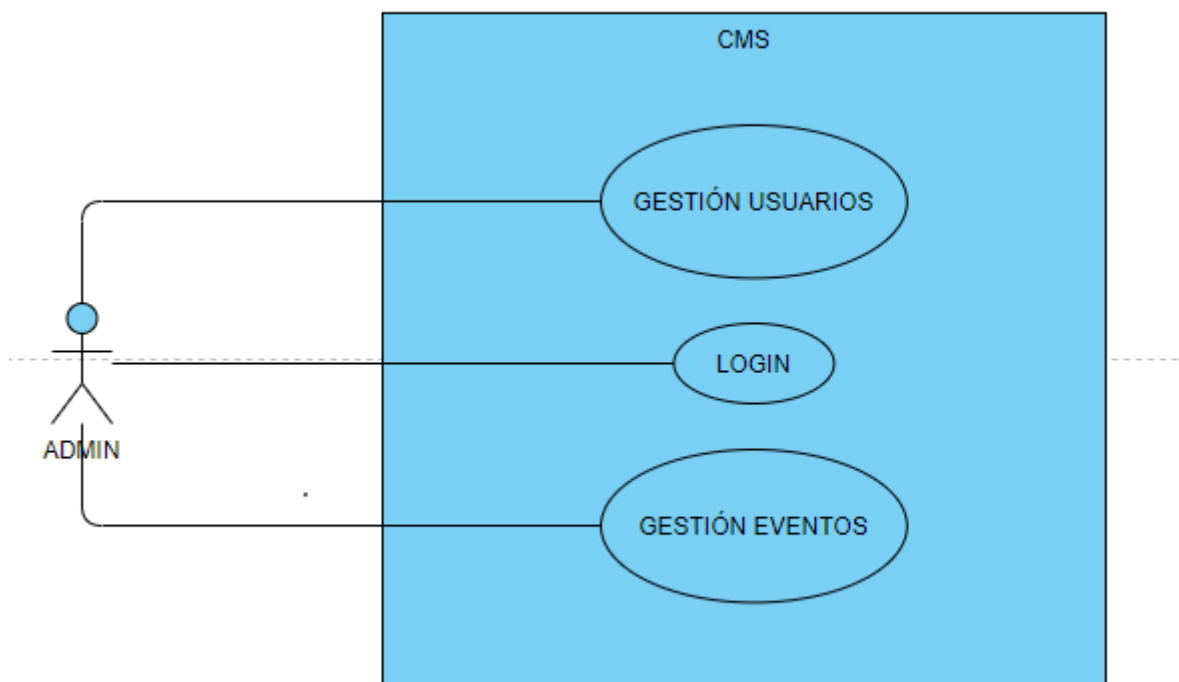
Requisitos funcionales CMS

RQ1 - Login: El administrador se deberá logear antes de poder entrar a las funcionalidades. Según quiera, el administrador podrá administrar usuarios o eventos

RQ2 - Gestión Usuarios: Se podrá administrar usuarios, donde podras ver su información, editar su información, añadir usuarios y eliminar usuarios

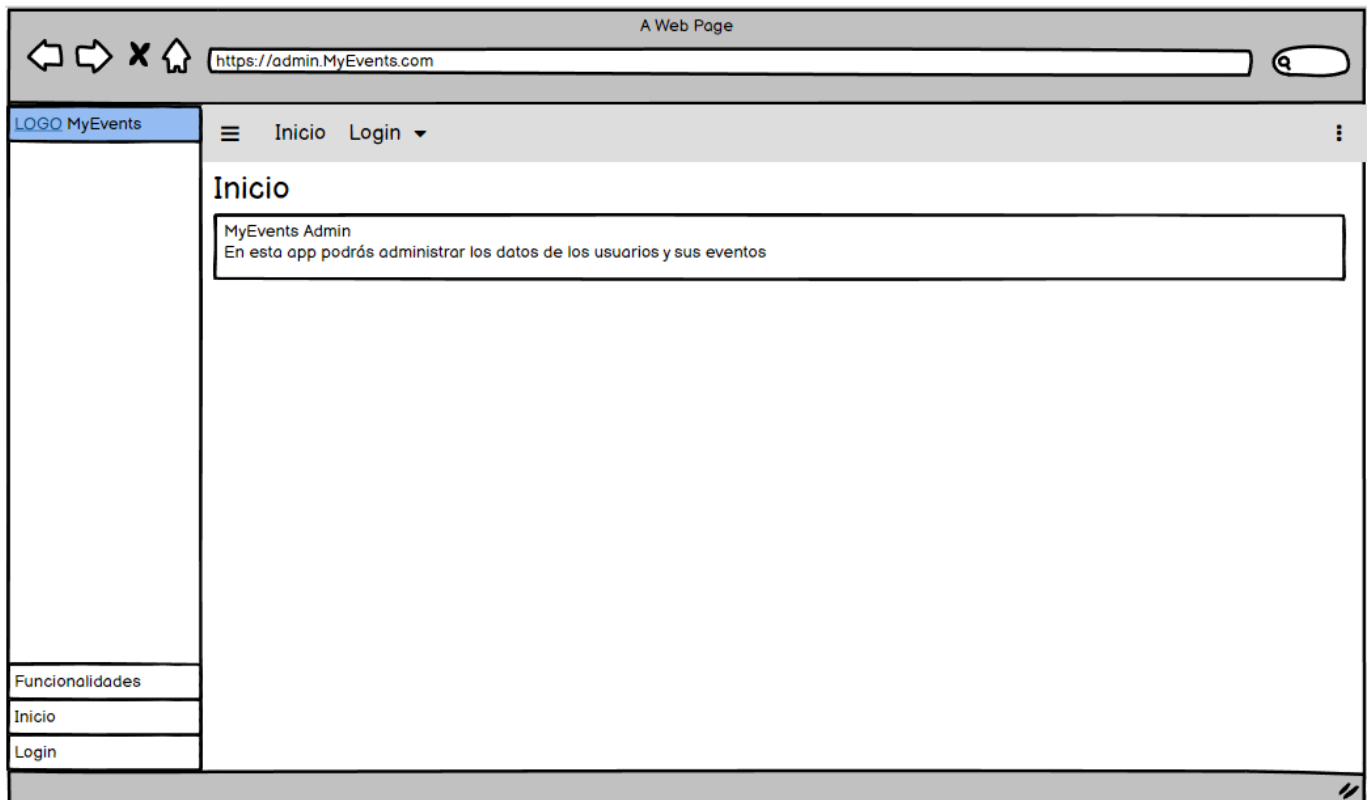
RQ3 - Gestión Eventos: Se podrá administrar citas o eventos, podras ver su información, editar su información, añadir usuarios y eliminar eventos

Diagrama Casos de Uso:

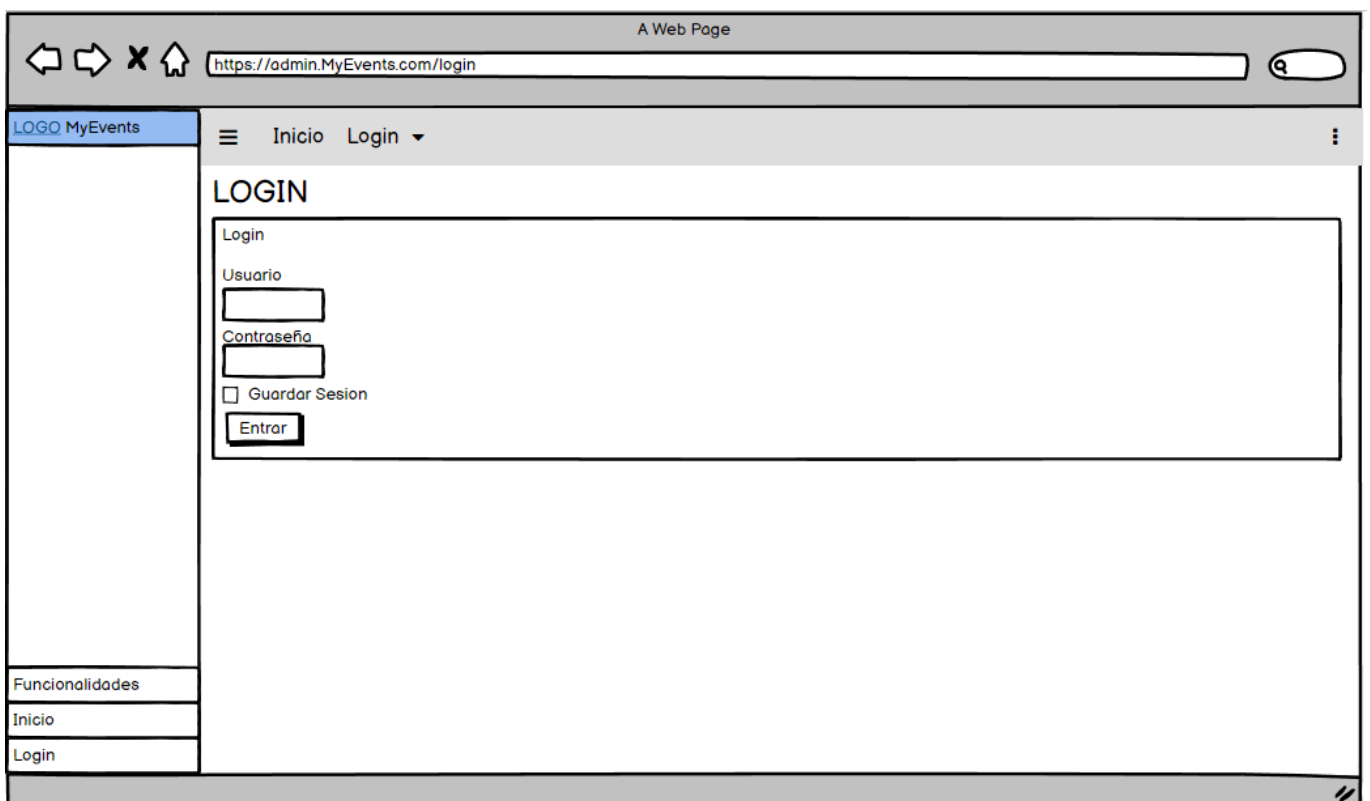


Wireframe

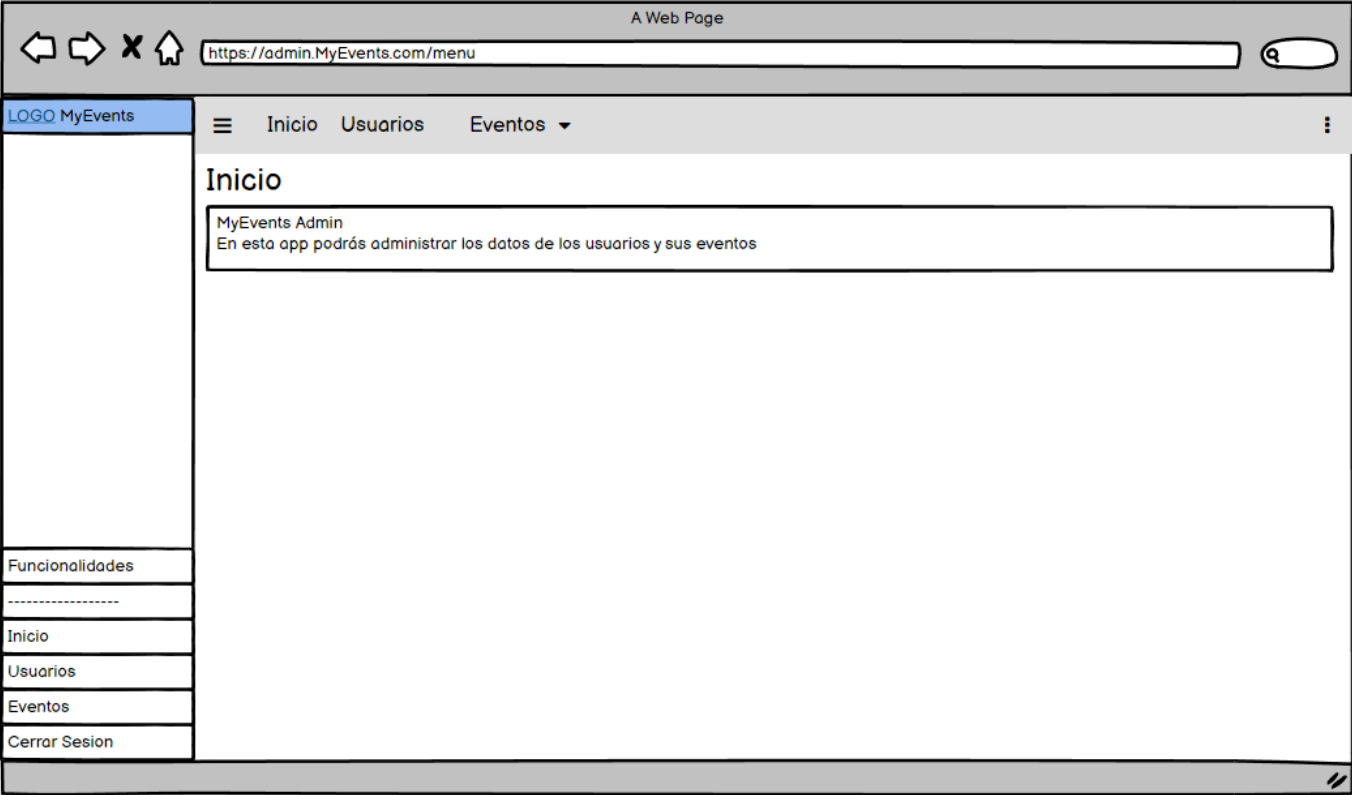
Página Inicio Sin Logearse: En esta pagina se podrá ver el inicio y ir a la pagina de login.



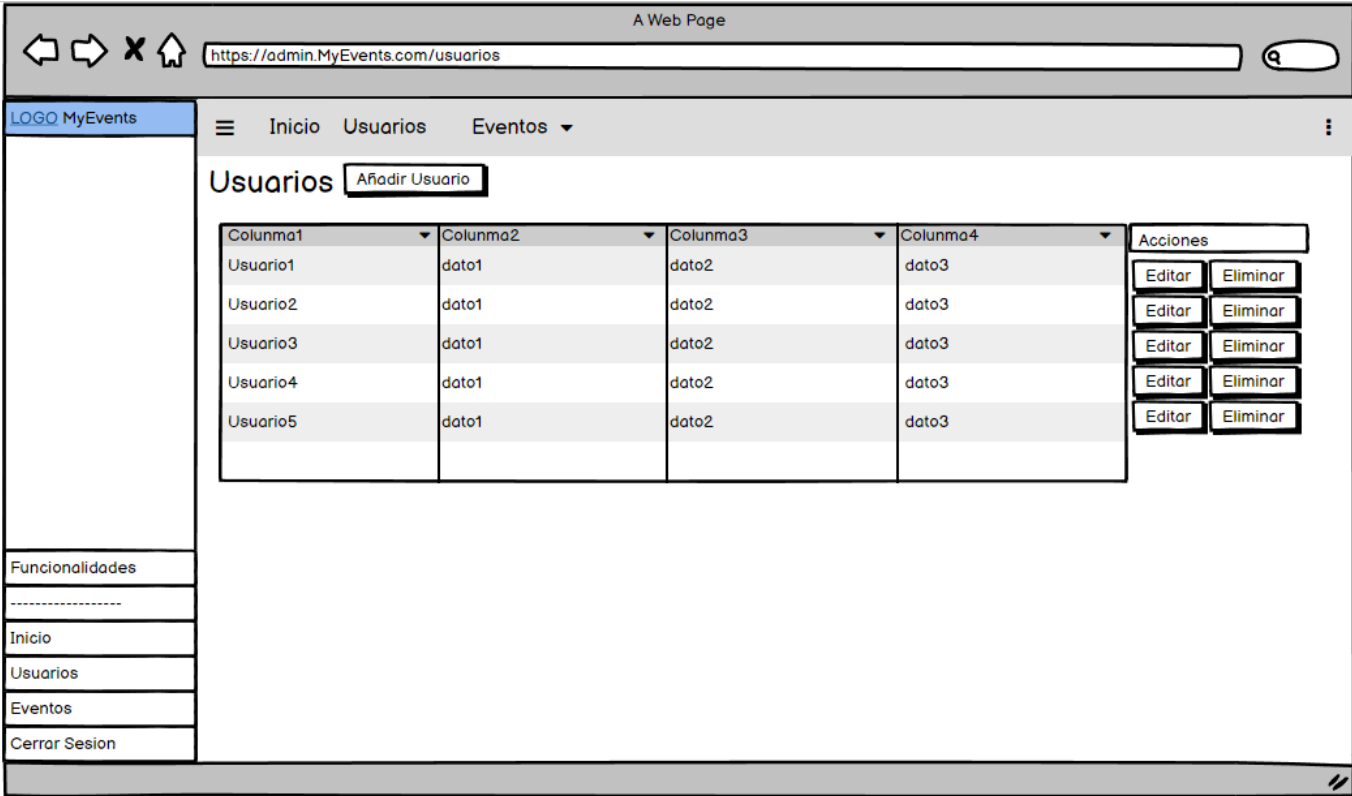
Página Login: En esta pagina podremos logearnos con nuestro usuario admin y guardar sesion.



Página Inicio Con Login: Una vez logeados, podremos ver más opciones como usuarios, eventos y cerrar sesion.



Página Usuarios: No mostrará una tabla donde podremos ver la información de los usuarios y las acciones como añadir, editar y eliminar usuarios.



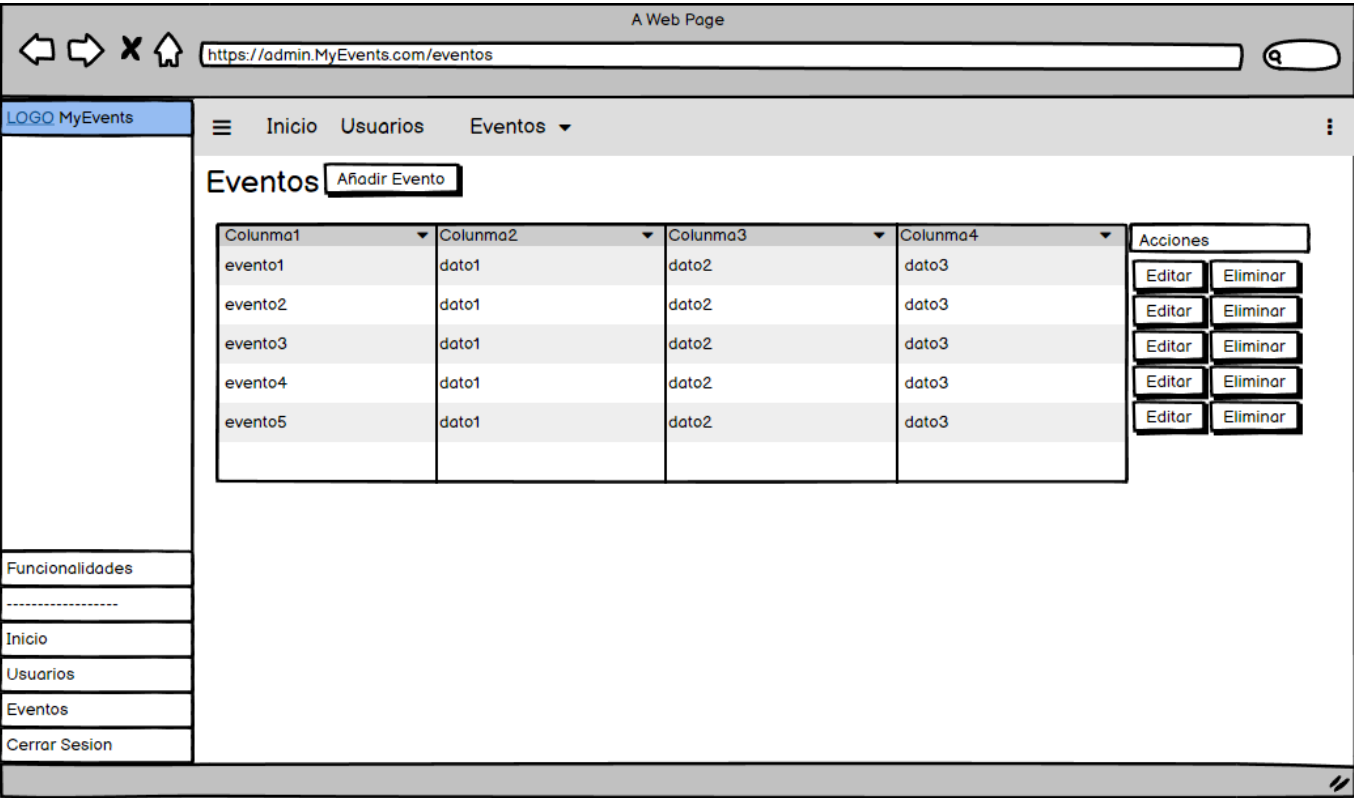
Página Agregar Usuario: En esta página podremos agregar un nuevo usuario incluyendo su información.

The screenshot shows a web browser window titled "A Web Page" with the URL `https://admin.MyEvents.com/agregarUsuario`. The page has a header with a logo "LOGO MyEvents" and navigation links: "Inicio", "Usuarios", and "Eventos". A sidebar on the left lists "Funcionalidades" and a menu with "Inicio", "Usuarios", "Eventos", and "Cerrar Sesión". The main content area is titled "Usuarios" and contains a form with four input fields labeled "dato1", "dato2", "dato3", and "dato4", followed by an "Añadir Usuario" button.

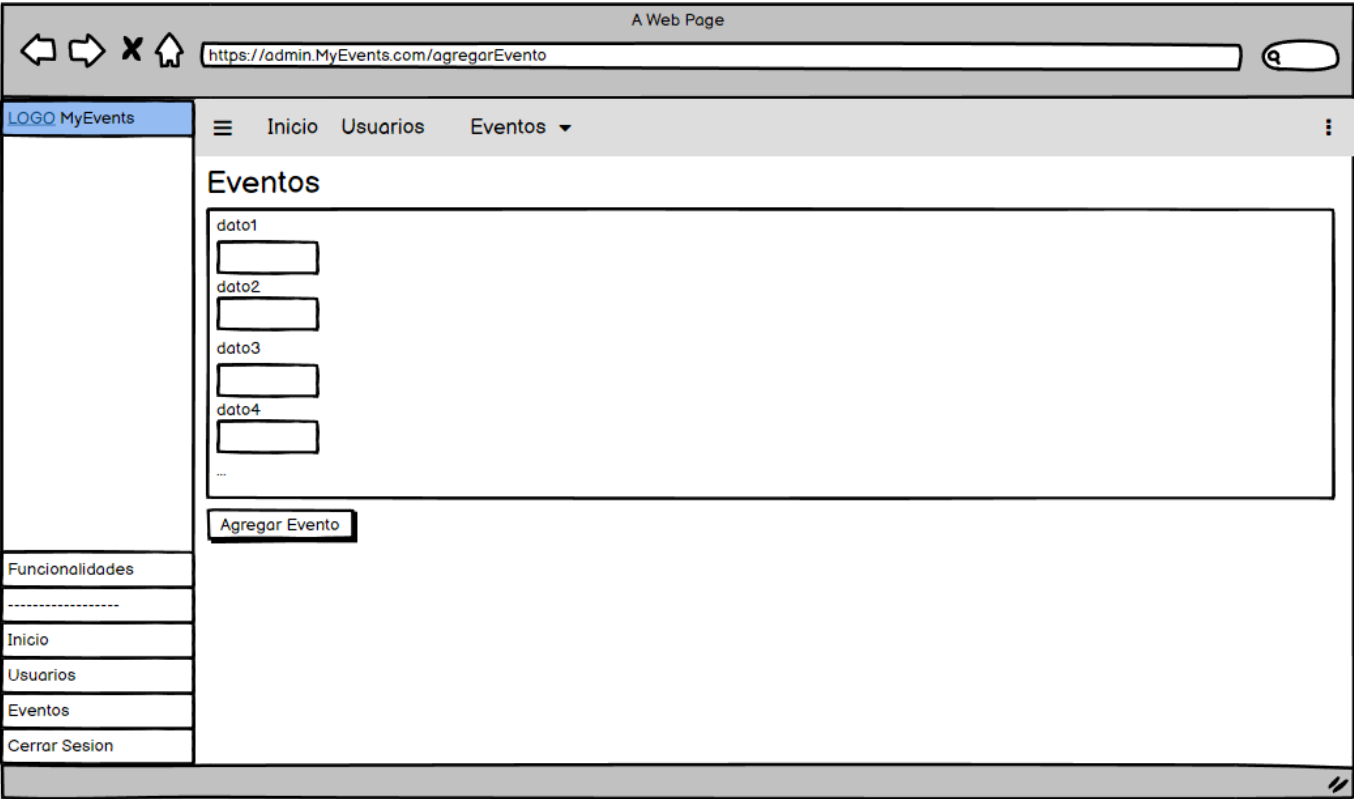
Página Modificar Usuario: En esta página, podremos editar la información del usuario.

The screenshot shows a web browser window titled "A Web Page" with the URL `https://admin.MyEvents.com/editarUsuario`. The page layout is identical to the previous one, but the main content area is titled "Usuarios" and contains a form with four input fields labeled "dato1", "dato2", "dato3", and "dato4", followed by an "Editar Usuario" button.

Página Eventos: No mostrará una tabla donde podremos ver la información de los eventos y las acciones como añadir, editar y eliminar eventos.



Página Agregar Evento: En esta página podremos agregar un nuevo evento incluyendo su información.



Página Modificar Evento: En esta página, podremos editar la información del evento.

The screenshot shows a web browser window titled "A Web Page" with the URL <https://admin.MyEvents.com/editarEvento>. The page has a header with a logo "LOGO MyEvents" and navigation links: "Inicio", "Usuarios", and "Eventos". The main content area is titled "Eventos" and contains a form with four input fields labeled "dato1", "dato2", "dato3", and "dato4", followed by an "Editor Evento" button. A sidebar on the left lists "Funcionalidades" and a menu with "Inicio", "Usuarios", "Eventos", and "Cerrar Sesion".

Página Errores: Esta página se mostrará cuando ocurra cualquier error con la página.

The screenshot shows a web browser window titled "A Web Page" with the URL <https://admin.MyEvents.com/error>. The page displays a "LOGO" and an error message: "ERROR: RECURSO NO ENCONTRADO".

API REST

API REST -- USUARIOS

Obtener todos los usuarios: Usando el metodo GET, podremos obtener todos los datos de los usuarios desde la ruta `/api/usuario`

MyEvents API Usuarios / Get Todos Usuarios MyEvents API

GET localhost:8080/api/usuario Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 142 ms 403 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "userID": 1,
4     "username": "admin2",
5     "email": "admin2@admin.com",
6     "password": "admin",
7     "phoneNumber": "0000000000",
8     "enabled": true
9   },
10  {
11    "userID": 2,
12    "username": "admin",
13    "email": "admin@admin.com",
14    "password": "admin",
15    "phoneNumber": "0000000000",
```

Obtener un usuario: Usando el metodo GET, podremos obtener todos los datos de un usuario desde la ruta /api/usuario/{userID}

MyEvents API Usuarios / Get Usuario ID MyEvents API

GET localhost:8080/api/usuario/2 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

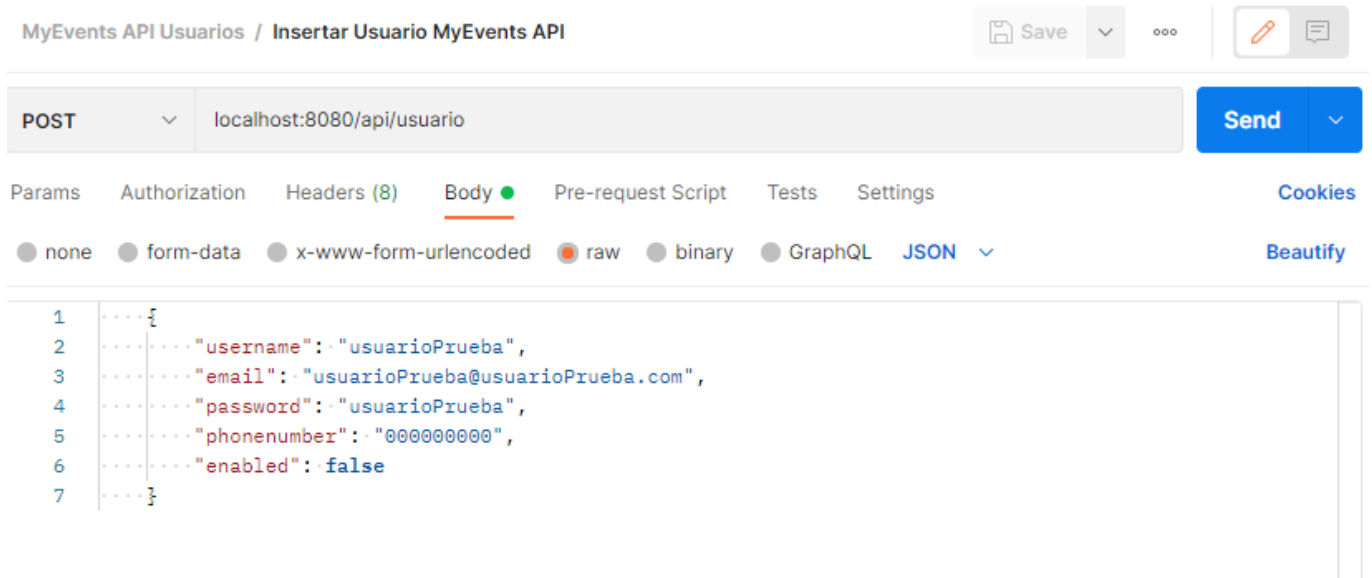
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 18 ms 281 B Save Response

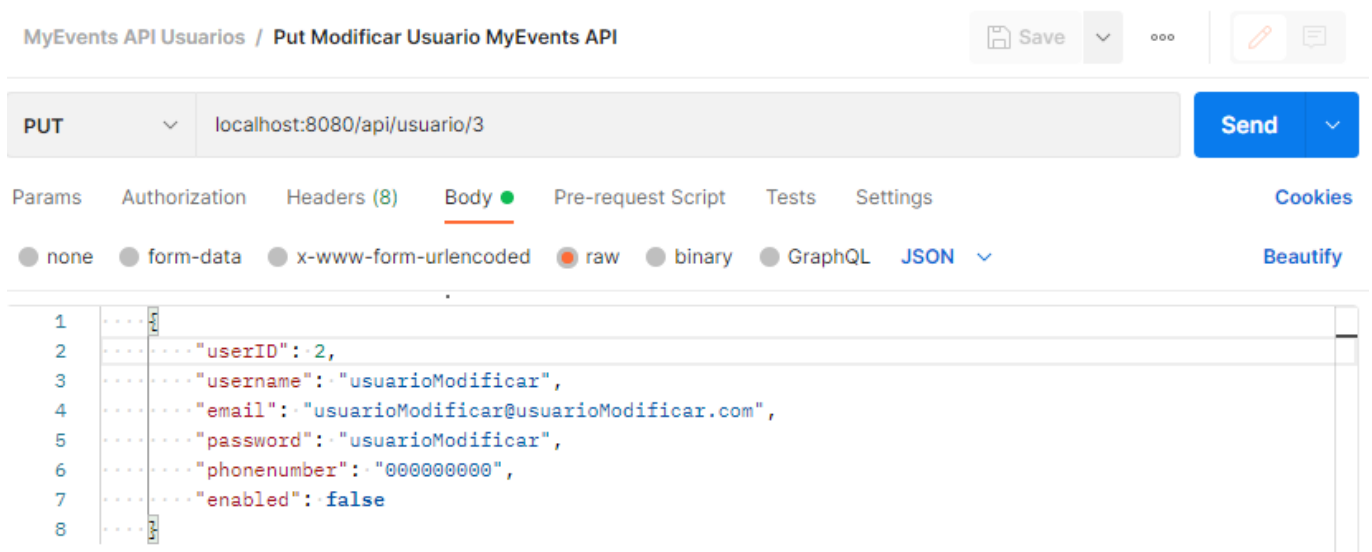
Pretty Raw Preview Visualize JSON

```
1 {
2   "userID": 2,
3   "username": "admin",
4   "email": "admin@admin.com",
5   "password": "admin",
6   "phoneNumber": "0000000000",
7   "enabled": true
8 }
```


Crear un usuario: Usando el metodo POST, podremos crear un usuario desde la ruta /api/usuario con el body que se ve en la imagen. (Todos los valores del body son obligatorios NotNull)



Modificar un usuario: Usando el metodo PUT, podremos modificar a un usuario desde la ruta /api/usuario/{userID} con el body indicado en la imagen. (Todos los valores del body son obligatorios NotNull)



Eliminar un usuario: Usando el metodo DELETE, podremos eliminar a un usuario desde la ruta /api/usuario/{userID}



API REST -- EVENTOS

Obtener todos los eventos: Usando el metodo GET, podremos obtener todos datos de los eventos desde la ruta /api/evento

MyEvents API Eventos / Get Todos Eventos MyEvents API

Save

GET localhost:8080/api/evento Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK 16 ms 1.17 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "eventID": 1,
3   "eventName": "Prueba",
4   "tema": "Prueba",
5   "startTime": "2021-10-24T12:48:50",
6   "endTime": "2021-10-24T12:48:51",
7   "available": true,
8   "eventPreference": false,
9   "coordinates": "31511451gh",
10  "videomeeting": null,
11  "userOwner": {
12    "userID": 1,
13    "username": "admin",
14    "email": "admin@myeventsapp.com",
15    "password": "admin",
16    "phoneNumber": "0000000000",
17    "cmsAdmin": true.
18  }
```

Obtener un evento: Usando el metodo GET, podremos obtener todos los datos de un evento desde la ruta /api/evento/{eventID}

MyEvents API Eventos / Get Evento ID MyEvents API

GET localhost:8080/api/evento/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results 200 OK 20 ms 677 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "eventID": 1,
3    "eventName": "Prueba",
4    "tema": "Prueba",
5    "startTime": "2021-10-24T12:48:50",
6    "endTime": "2021-10-24T12:48:51",
7    "available": true,
8    "eventPreference": false,
9    "coordinates": "31511451gh",
10   "videomeeting": null,
11   "userOwner": {
12     "userID": 1,
13     "username": "admin",
14     "email": "admin@myeventsapp.com",
15     "password": "admin",
16     "phoneNumber": "0000000000",
17     "cmsAdmin": true,
18     "enabled": true
19   },
20   "userSummoner": {
21     "userID": 2,
22     "username": "admin"
23   }
24 }

```

Crear un evento: Usando el metodo POST, podremos crear un evento desde la ruta /api/evento con el body que se ve en la imagen. (Todos los valores del body son obligatorios NotNull)

MyEvents API Eventos / Insertar Evento MyEvents API

POST localhost:8080/api/evento Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1  {
2    "eventName": "Prueba2",
3    "tema": "Prueba2",
4    "startTime": "2021-10-24T12:48:50",
5    "endTime": "2021-10-24T12:48:51",
6    "available": true,
7    "eventPreference": false,
8    "coordinates": "31511451gh",
9    "videomeeting": null,
10   "userOwner": {"userID": 1},
11   "userSummoner": {"userID": 2}
12 }

```

Modificar un evento: Usando el metodo PUT, podremos modificar un evento desde la ruta /api/evento/{userID} con el body indicado en la imagen. (Todos los valores del body son obligatorios NotNull)

MyEvents API Eventos / Put Modificar Evento MyEvents API

Save ...

PUT localhost:8080/api/evento/3 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   ... "eventID": 2,
3   ... "eventName": "Prueba2",
4   ... "tema": "Prueba2",
5   ... "startTime": "2021-10-24T12:48:50",
6   ... "endTime": "2021-10-24T12:48:51",
7   ... "available": true,
8   ... "eventPreference": false,
9   ... "coordinates": "31511451gh",
10  ... "videomeeting": null,
11  ... "userOwner": {"userID": 2},
12  ... "userSummoner": {"userID": 1}
13 }
```

Eliminar un evento: Usando el metodo DELETE, podremos eliminar un evento desde la ruta /api/evento/{eventID}

MyEvents API Eventos / Delete Evento MyEvents API

Save ...

DELETE localhost:8080/api/evento/2 Send

Requisitos no funcionales

Seguridad: se necesitará un token para acceder a la api

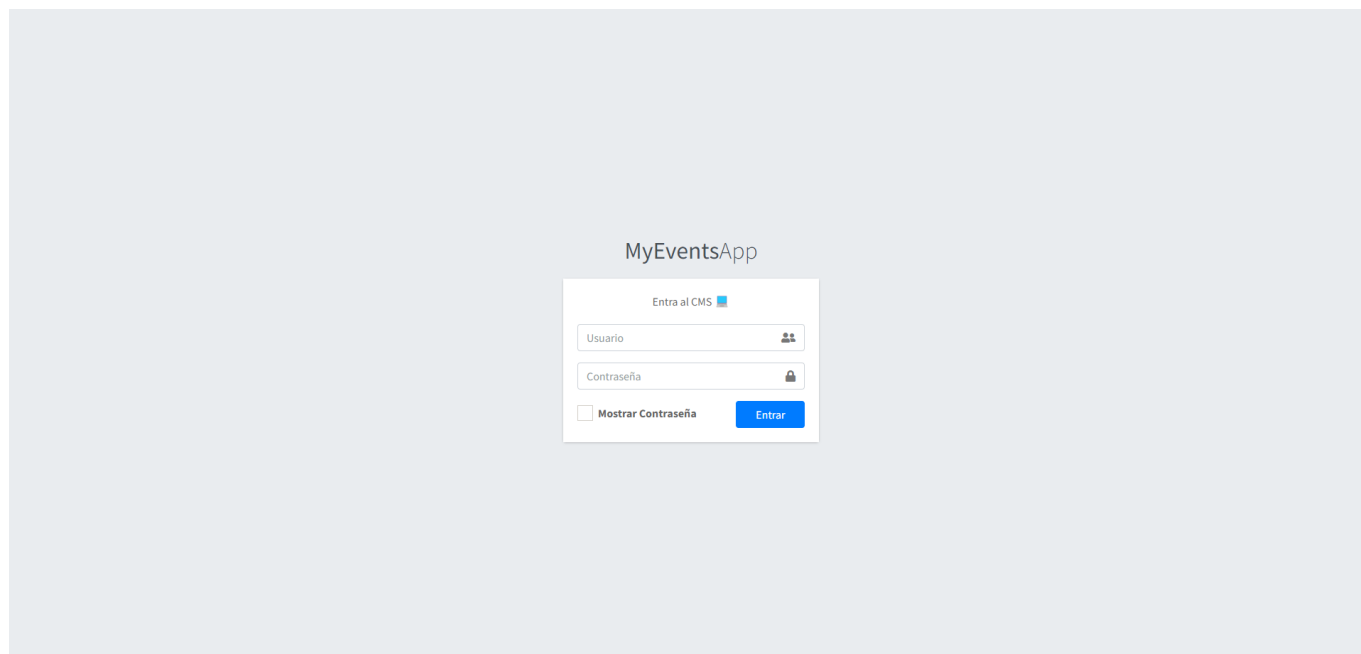
Conectividad: se necesitará conexión wifi para poder funcionar, con http

Dispositivos: se podrá usar desde cualquier dispositivo con un navegador web

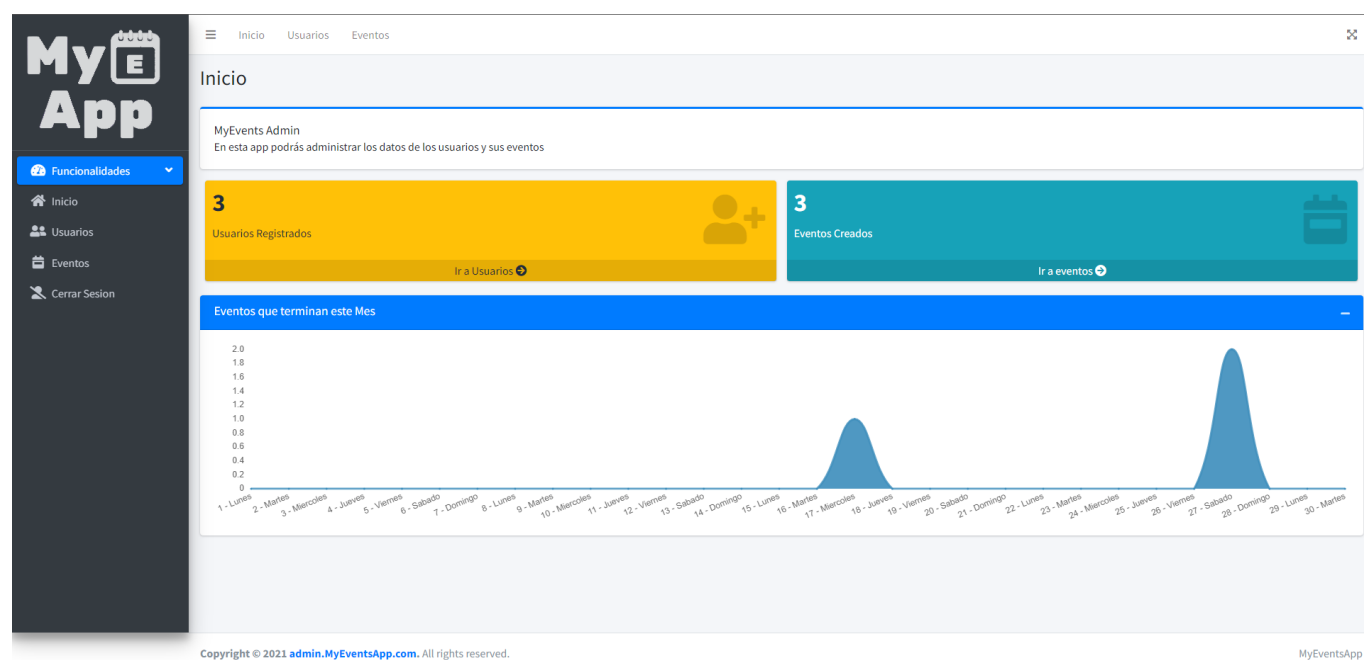
Análisis Funcional

Interfaz gráfico

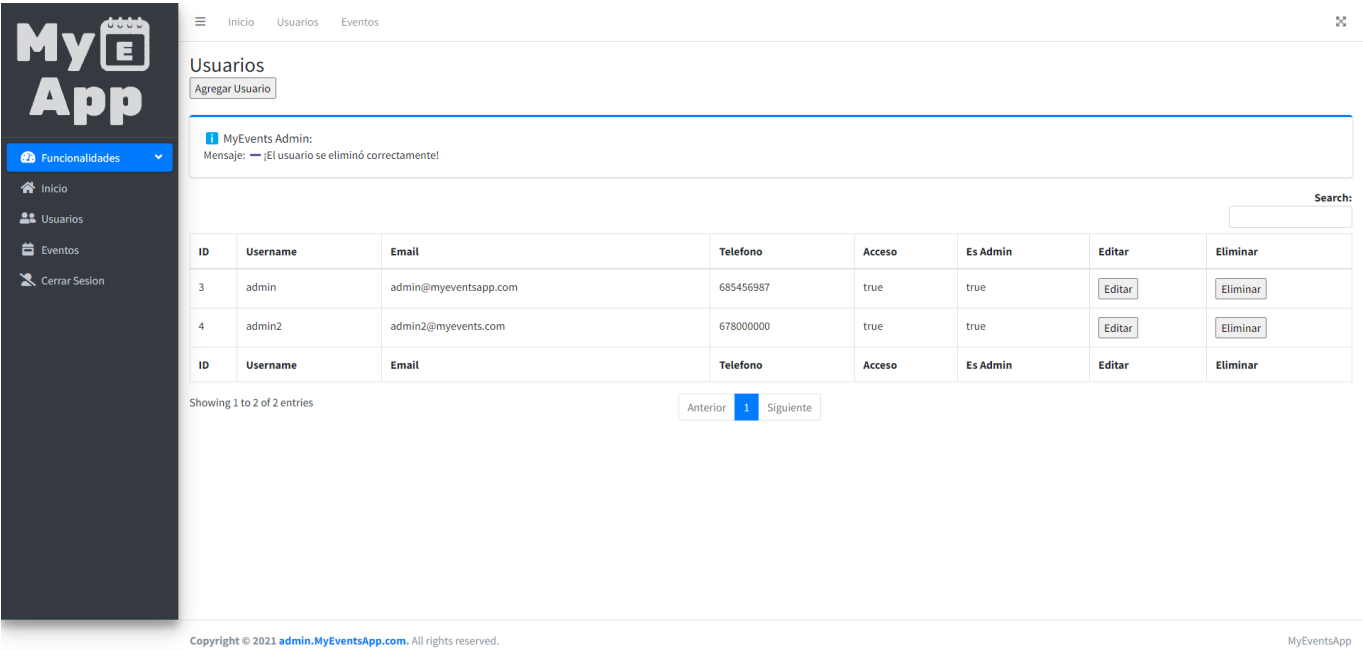
Página Login: En esta pagina podremos logearnos con nuestro usuario admin y guardar sesion.



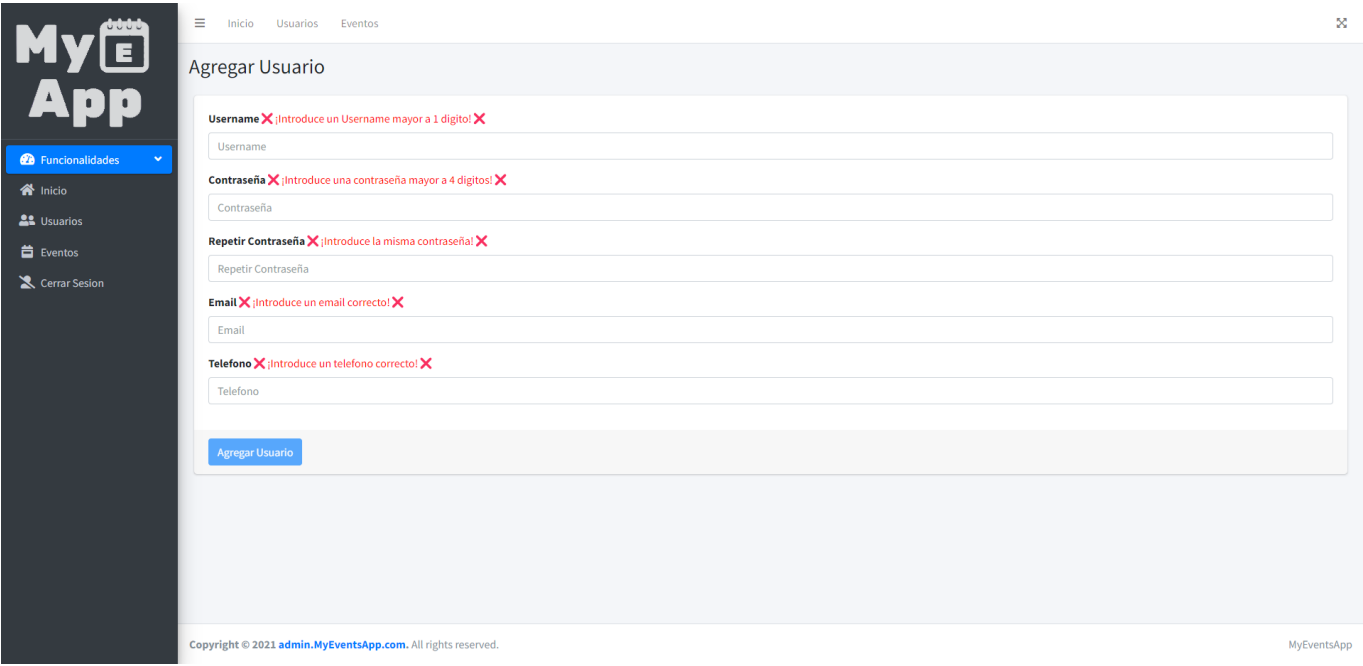
Página Inicio Con Login: Una vez logeados, podremos ver más opciones como usuarios, eventos y cerrar sesion.



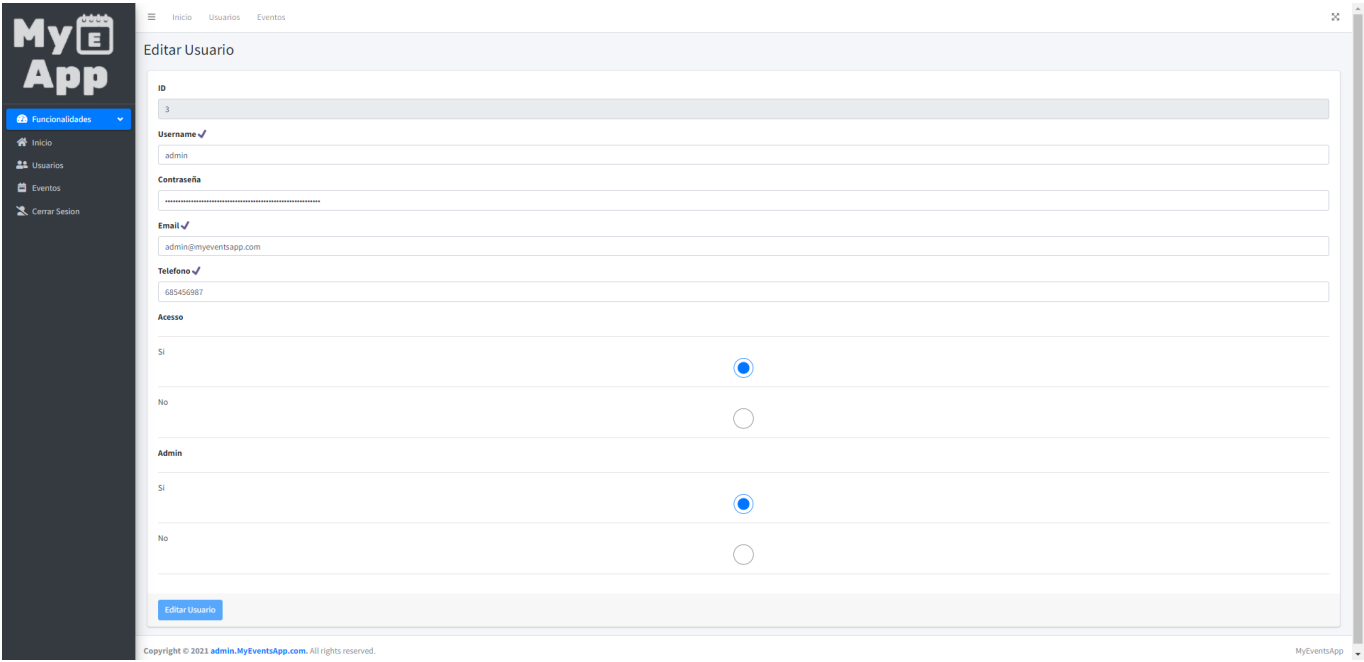
Página Usuarios: No mostrará una tabla donde podremos ver la información de los usuarios y las acciones como añadir, editar y eliminar usuarios.



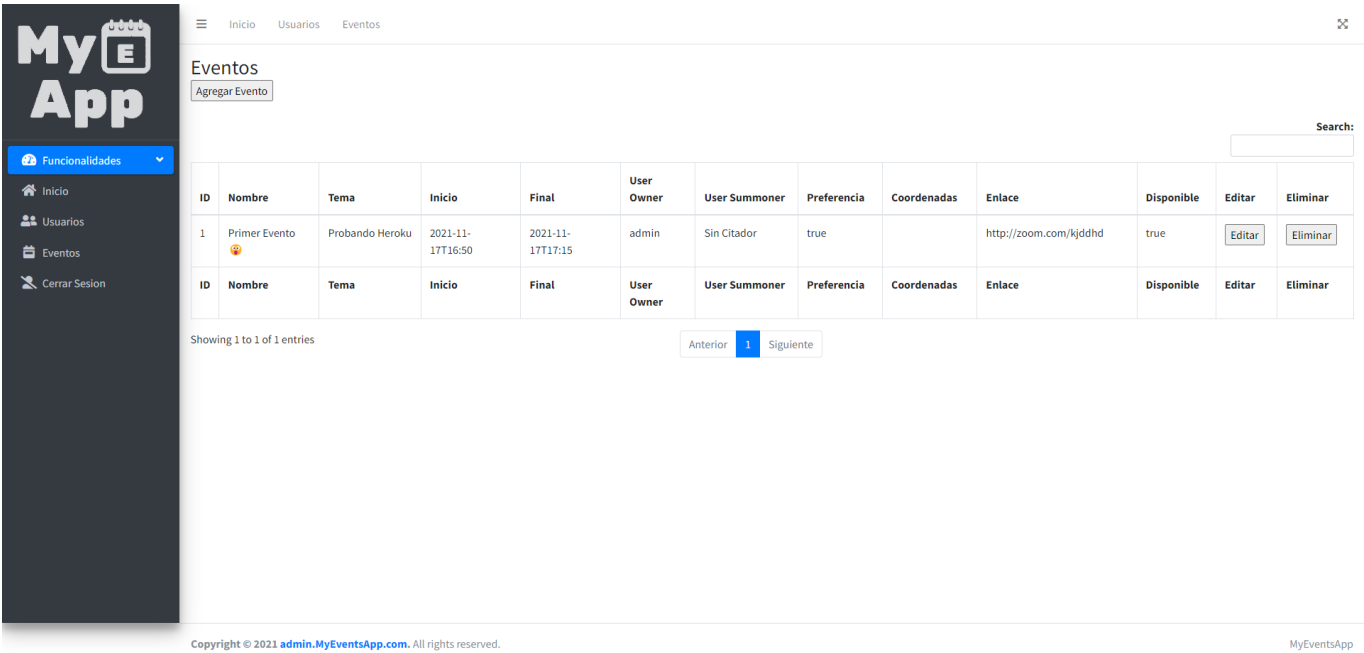
Página Agregar Usuario: En esta página podremos agregar un nuevo usuario incluyendo su información.



Página Modificar Usuario: En esta página, podremos editar la información del usuario.



Página Eventos: No mostrará una tabla donde podremos ver la información de los eventos y las acciones como añadir, editar y eliminar eventos.



Página Agregar Evento: En esta página podremos agregar un nuevo evento incluyendo su información.

MyEApp

Funcionalidades

Inicio

Usuarios

Eventos

Cerrar Sesión

Inicio

Usuarios

Eventos

Agregar Evento

ID Usuario Dueño **X** introduce un ID **X**

ID Usuario Dueño

ID Usuario Citador **X** introduce un ID **X**

ID Usuario Citador

Nombre Evento **X** introduce un nombre de evento mayor a 4 digitos **X**

Nombre Evento

Tema **X** introduce un nombre de evento mayor a 4 digitos **X**

Tema

Inicio **X** Elige una fecha de inicio **X**

mm/dd/yyyy --:-- --

Fin **X** Elige una hora de fin que sea mayor a la hora de inicio **X**

mm/dd/yyyy --:-- --

Coordenadas **X** introduce unas coordenadas mayor a 9 digitos **X**

Coordenadas

Preferencia del Evento

Presencial

☒

Videometing

☐

Habilitado

Si

☒

No

☐

Agregar Evento

Copyright © 2021 admin.MyEventsApp.com. All rights reserved.

MyEventsApp

Página Modificar Evento: En esta página, podremos editar la información del evento.

MyEApp

Funcionalidades

Inicio

Usuarios

Eventos

Cerrar Sesión

Inicio

Usuarios

Eventos

Editar Evento

ID Evento

1

ID Usuario Dueño ✓

1

ID Usuario Citador ✓

1

Nombre Evento ✓

Primer Evento

Tema ✓

Problemas de inicio

Inicio **X** Elige una fecha de inicio que no sea una fecha pasada o que no sea a esta hora **X**

11/17/2021 04:30 PM

Fin ✓

11/17/2021 05:30 PM

Enlace Videometing ✓

http://www.com/iglad

Preferencia del Evento

Presencial

☐

Videometing

☒

Habilitado

Si

☒

No

☐

Editar Evento

Copyright © 2021 myeventapp marlo, heribertoapp.com. All rights reserved.

MyEventsApp

Página Errores: Esta página se mostrará cuando ocurra cualquier error con la página.

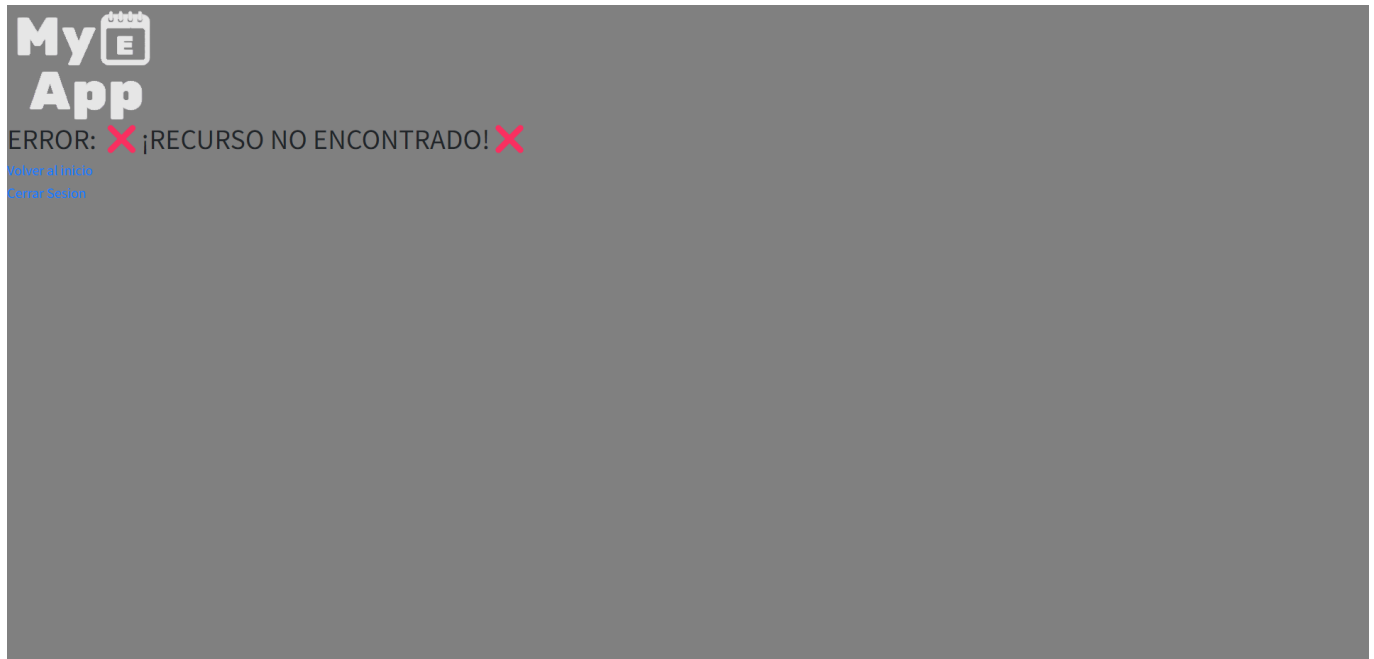
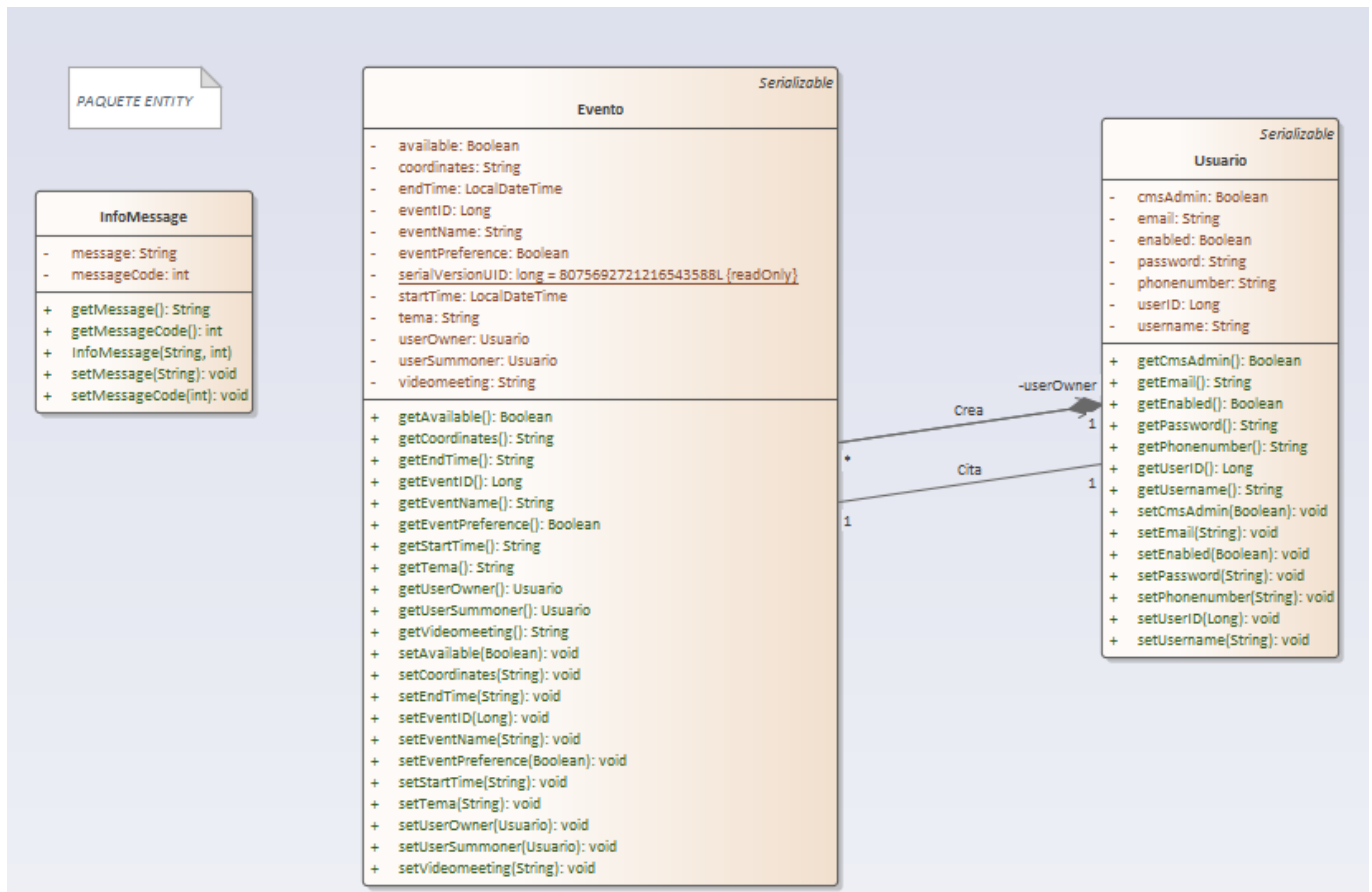


Diagrama de clases



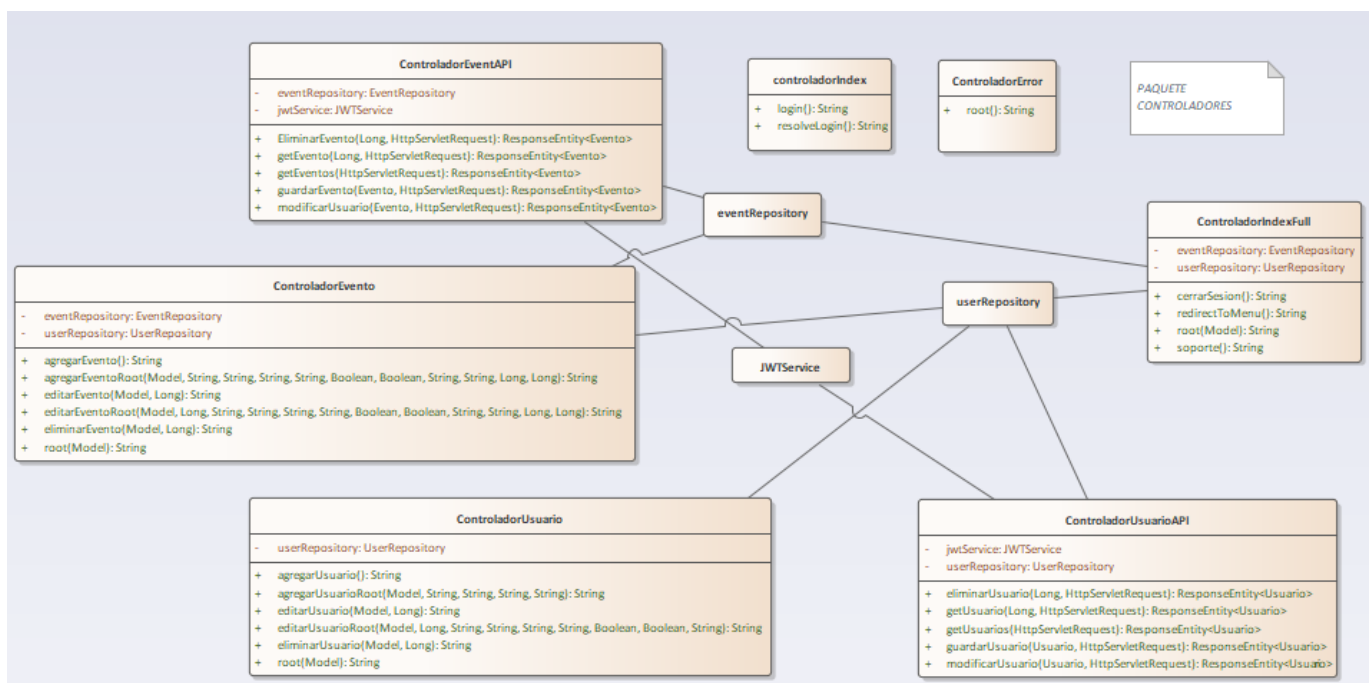
Clase `ServerEventosApplication`: Esta clase contiene el main del spring boot.



Clase InfoMensaje: Esta clase se utiliza para guardar información de mensajes que se quieran mostrar en el CMS como comunicar algún error.

Clase Evento: Esta clase almacena la información de cada evento que haya en nuestro CMS.

Clase Usuario: Esta clase almacena la información de cada usuario del CMS.



Clase ControladorIndex: Esta clase es el controlador principal, el cual responde a las llamadas a /login.

Clase ControladorError: Esta clase, responde a las llamadas a /error para mostrar paginas de errores.

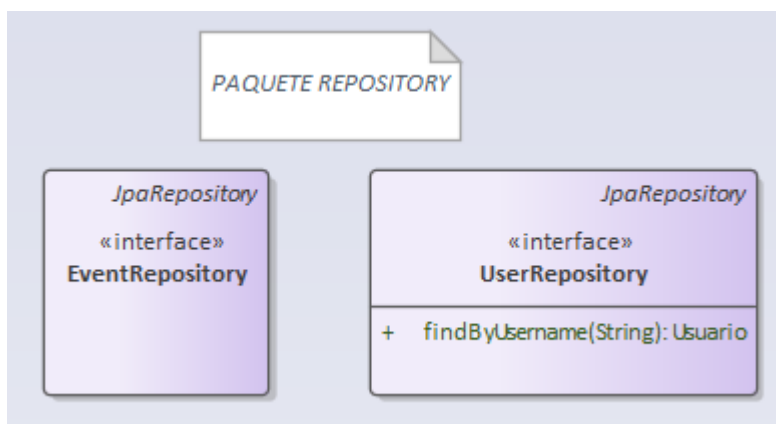
Clase ControladorIndexFull: Esta clase es el controlador donde se maneja las llamadas al menu principal, a cerrar sesion y a soporte.

Clase ControladorUsuario: Esta clase es el controlador donde realizaremos el CRUD para los usuarios desde el CMS.

Clase ControladorUsuarioAPI: Esta clase es el controlador donde la aplicación movil realizará las llamadas para realizar el CRUD de los usuarios.

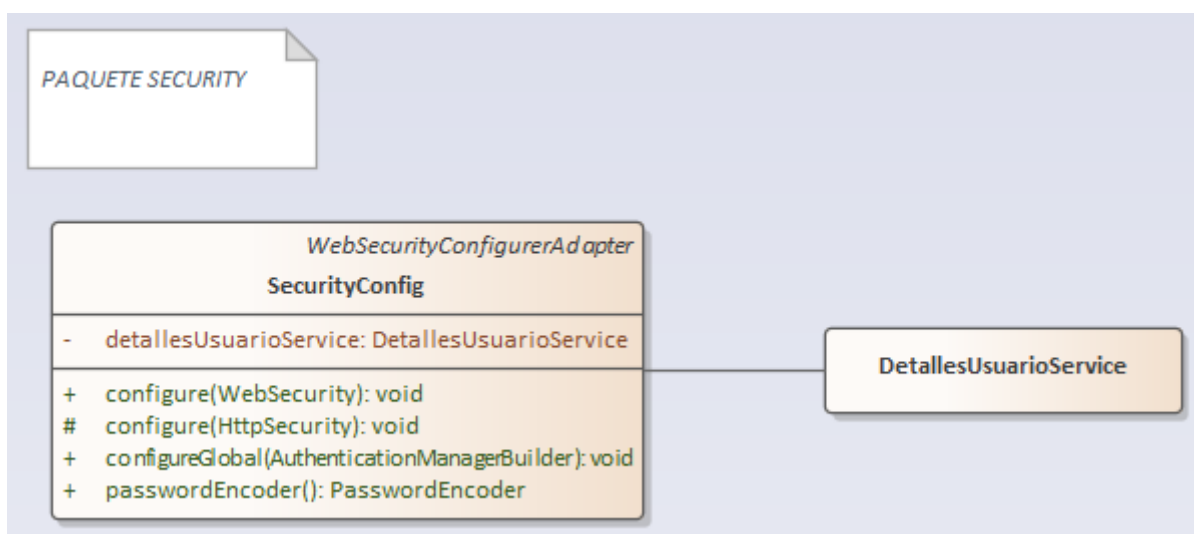
Clase ControladorEventos: Esta clase es el controlador donde realizaremos el CRUD para los eventos desde el CMS.

Clase ControladorControladorEventosAPI: Esta clase es el controlador donde la aplicación movil realizará las llamadas para realizar el CRUD de los eventos.

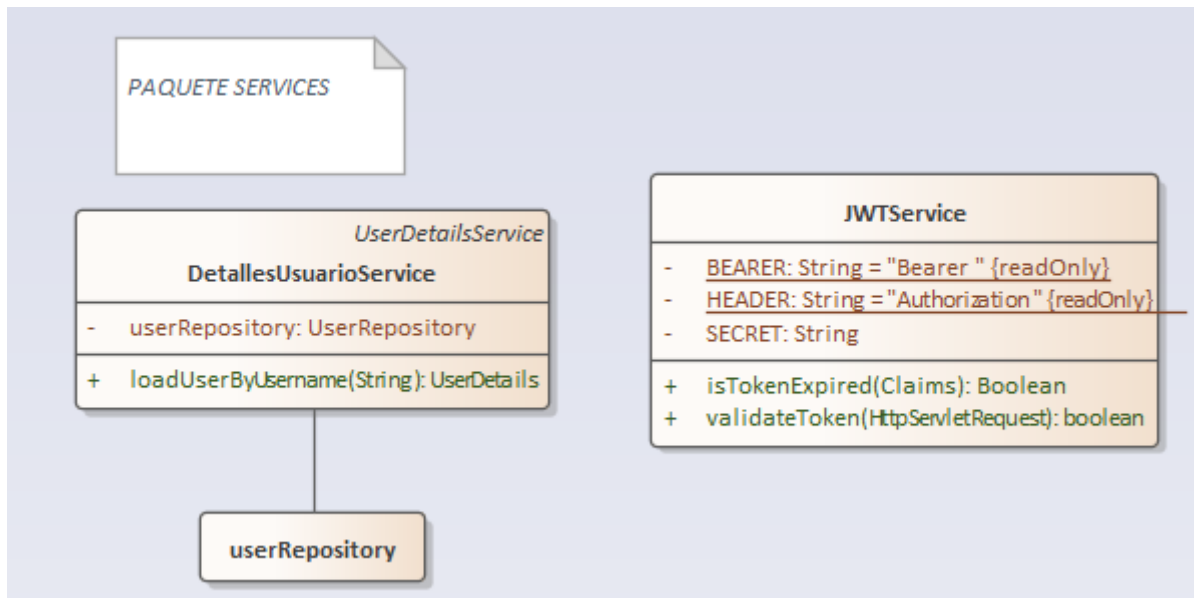


Clase EventRepository: Esta clase extiende de JpaRepository la cual tiene funciones para realizar las queries para los eventos.

Clase UserRepository: Esta clase extiende de JpaRepository la cual tiene funciones para realizar las queries para los usuarios.



Clase SecurityConfig: Esta clase es la que se encarga de manejar el acceso a los recursos de nuestro CMS y de encriptar las contraseñas de los usuarios.



DetallesUsuariosService: Esta clase se encarga de ver el rol de los usuarios para permitir accesos o no en el CMS.

JWTService: Esta clase es la que se encarga de verificar los tokens que limitan el acceso a nuestras API's del CMS.

Diagrama E/R

TABLA -- USUARIO

Todos los datos de esta tabla son Not Null, no pueden ser nulos

userID: Es la clave principal de la tabla, que será el identificador del usuario, que es de tipo bigint.

username: Es el nombre de usuario, que es de tipo varchar.

email: Es el correo electronico del usuario, que es de tipo varchar.

password: Es la contraseña del usuario, que es de tipo varchar.

phonenummer: Es el numero de telefono del usuario, que es de tipo varchar.

enabled: Nos servirá para saber si el usuario puede utilizarse o no, que es de tipo bit que en verdad es un boolean.

TABLA -- EVENTO

eventID: Es la clave principal de la tabla, que será el identificador del evento, que es de tipo bigint. NotNull.

eventname: Es el nombre de evento, que es de tipo varchar. NotNull.

eventname: Es el tema del evento, que es de tipo varchar. NotNull.

start_time: Es la fecha y hora del inicio del evento, que es de tipo Datetime. NotNull.

end_time: Es la fecha y hora de final del evento, que es de tipo Datetime. NotNull.

event_preference: Con este dato, según su valor, si es 0 será un evento presencial y si no, será un evento meeting y con esto trabajaremos con diferentes datos según este valor, es de tipo bit, que en realidad es un boolean. NotNull.

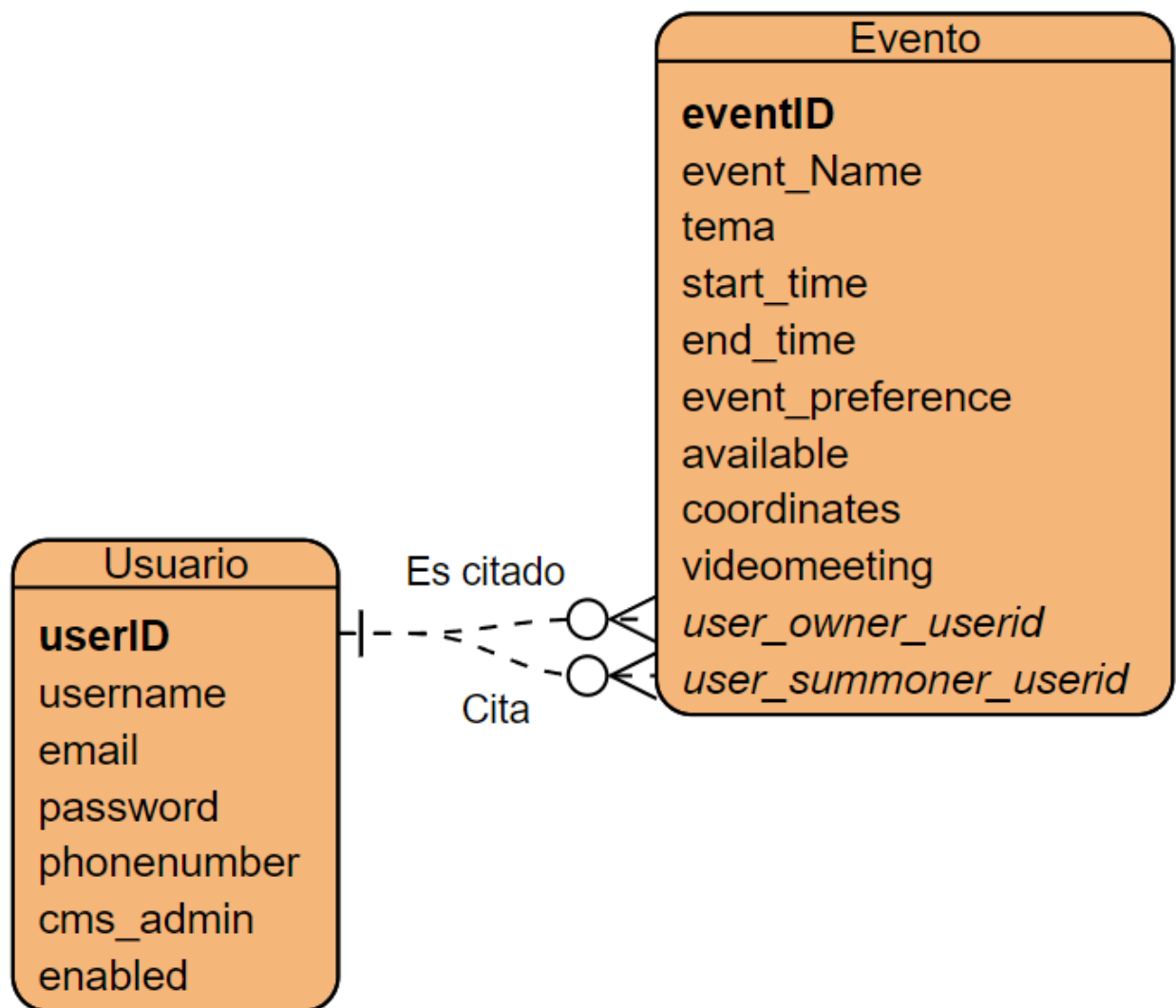
coordinates: Son las coordenadas de la localización del lugar de quedada del evento, es de tipo varchar.

videomeeting: Es el enlace de la videoconferencia del evento, es de tipo varchar.

available: Con este dato sabremos si el evento está activado o no según su valor (0 = desahabilitado / 1 = habilitado), es de tipo bit, pero en realidad es de tipo boolean. NotNull.

user_owner_id_user: es la id del usuario que ha creado el evento, es de tipo bigint. NotNull.

user_summoner_id_user: es la id del usuario que ha citado al creador el evento, es de tipo bigint.



Plan de pruebas

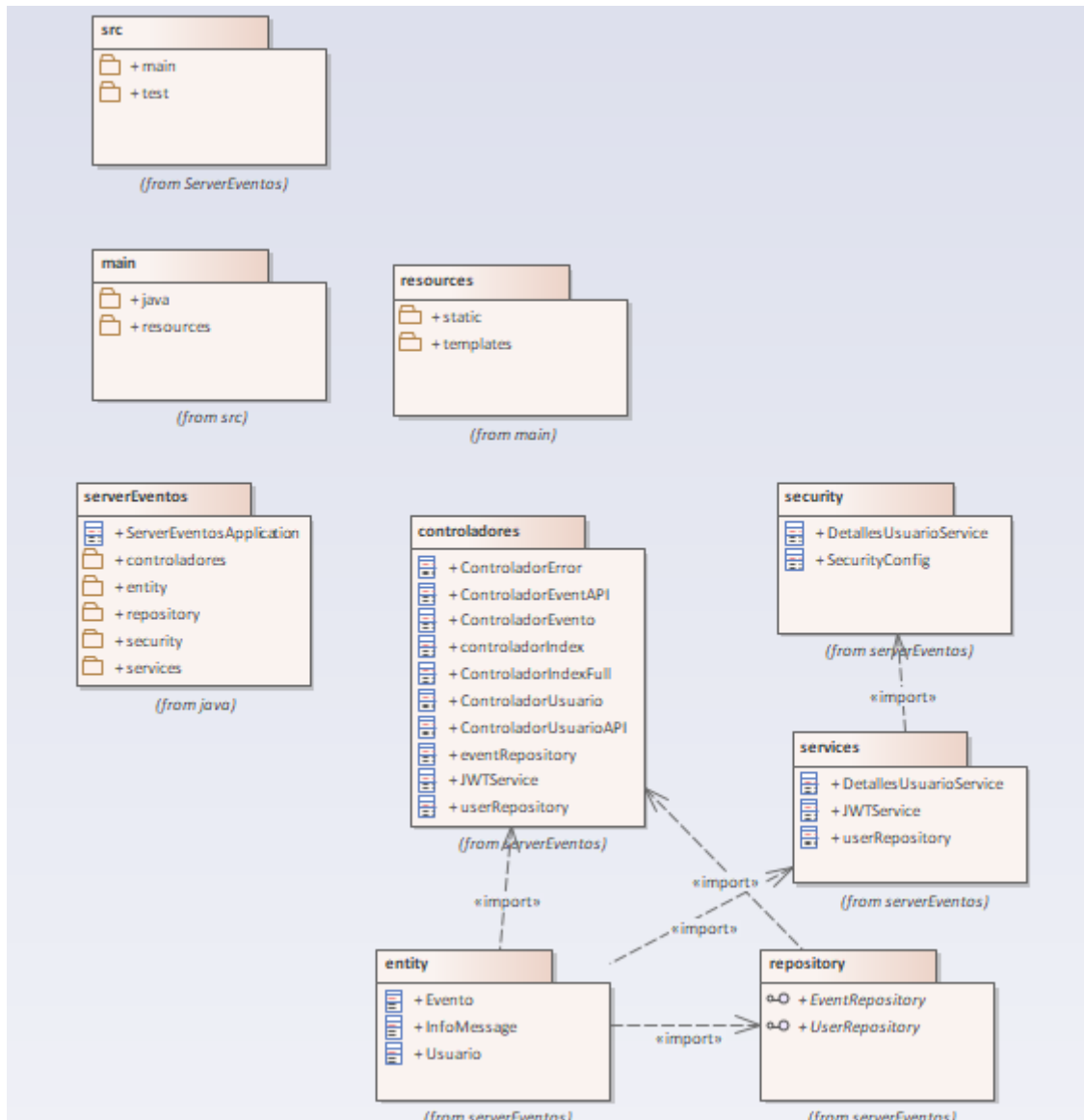
Prueba 1: Comprobar Existencia del un Usuario en el login: Se pasaran como parametros un usuario y una contraseña y si el usuario existe, nos devolverá un true, si no, nos devolverá false.

Prueba 2: Comprobar usuario agregado: Se pasará un objeto usuario y si el usuario se agrega correctamente, nos devolverá un true, si no, nos devolverá un false.

PLAN DE PRUEBAS EN DESARROLLO

Diseño Técnico

Diagrama de paquetes y de componentes



Paquete Controladores: En este paquete se almacenan las clases que se encargan del redireccionar a las vistas segun se solicite.

Paquete Entity: En este paquete se almacenan las clases de los objetos que almacenamos en la bd y otros objetos.

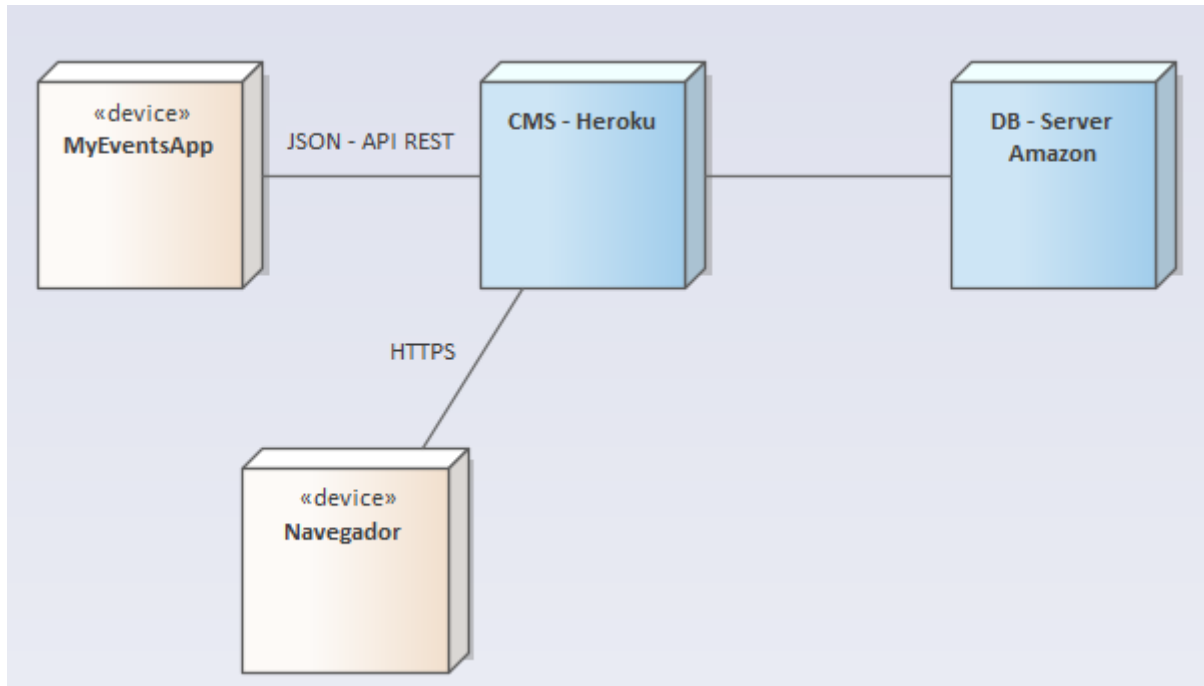
Paquete Repository: En este paquete se almacenan las clases que se encargan de realizar las queries.

Paquete services: En este paquete se almacenan las clases donde almacenamos funcionalidades complejas como la logica de JWT.

Paquete security: En este paquete se almacenan las clases se encargan de mantener el CMS seguro.

Paquete resources: En este paquete se almacena los recursos como las vistas html y archivos estaticos como css, js o imagenes.

Arquitectura del sistema



CMS: El CMS esta desplegado en la plataforma de Heroku. Este CMS esta desarrollado con Spring boot.

MyEventsApp: Esta es la aplicación android desarrollada con java.

Navegador: Se trata de un cliente web como chrome, firefox...

Base de datos: La base de datos esta hosteada en amazon ya que se utiliza un plugin en heroku que te la hostea ahí.

Entorno de desarrollo, librerías y servicios

Spring-Boot: Este es el framework que he utilizado para desarrollar el cms, el cual permite crear facilmente un servidor web.

Thymeleaf: Es el motor de plantillas java que he utilizado para renderizar mi html y que sea dinamico con los datos del cms.

JWT: Esta es la tecnologia que he utilizado para realizar un api rest seguro y con uso limitado a mi aplicación android. Se trata de un web token encriptado.

Base de datos: Se utiliza una base de datos relacional, MySQL.

Hibernate: Spring-boot, permite la generación de base de datos relacionales con JPA, a traves de esta tecnologia.

Spring-Boot Security: Es parte de spring-boot, y nos permite limitar el uso de los recursos de nuestro CMS según como lo configuremos entre otras muchas cosas.

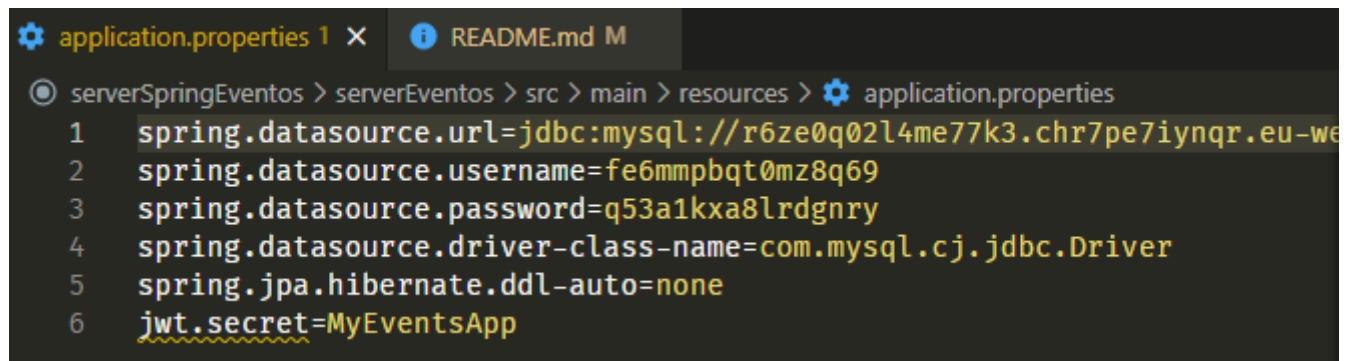
BCrypt: Esta es la tecnología que utiliza Spring boot security para poder encriptar las contraseñas de los usuarios para hacer la plataforma más segura.

Instrucciones para la compilación, ejecución y despliegue de la aplicación

Para poder acceder al cms debes usar el usuario rogelio (en minusculas) con la contraseña admin

Se deberá tener java 1.8 en el dispositivo además de conexión a internet, además de maven y spring boot.

Puedes cambiar la dirección de la base de datos en:



```
serverSpringEventos > serverEventos > src > main > resources > application.properties
1  spring.datasource.url=jdbc:mysql://r6ze0q02l4me77k3.chr7pe7iynqr.eu-we
2  spring.datasource.username=fe6mmpbqt0mz8q69
3  spring.datasource.password=q53a1kxa8lrdgnry
4  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5  spring.jpa.hibernate.ddl-auto=none
6  jwt.secret=MyEventsApp
```

Informe de pruebas

[Video](#)

Conclusiones

Conocimientos adquiridos

Reflexiona sobre el trabajo realizado durante el desarrollo de la aplicación web y sobre los conocimientos adquiridos, problemas encontrados, etc.

Gracias a este trabajo he aprendido a como utilizar el framework spring boot y crear con este un CMS con sus respectivos controladores, su respectiva seguridad, además se aprende a crear una API y como funciona. Además, he aprendido a utilizar los tokens web como JWT. También, he aprendido en que consiste un ORM como hibernate y como gracias a este podemos crear base de datos relacionales con instrucciones JPA. En el desarrollo del CMS no me he encontrado ningún error que me haya complicado el desarrollo.

En conclusión, gracias a esta práctica, he aprendido muchos conceptos a la hora de crear un CMS y sus funciones y a empezar a entender el desarrollo con el framework spring boot.

Mejoras futuras

- Mejorar visualmente en algunos aspectos.
- Añadir más gráficos.