

Documentación Proyecto SoccerSocialHub @
SoccerSocialHub Abel García Sánchez



SoccerSocialHub

Fecha: 12-06-2024



ÍNDICE

1. Introducción	3
2. Identificación de las necesidades del proyecto	4
3. Breve análisis/comparativa con las alternativas del mercado	5
4. Justificación del proyecto	6
5. Uso de stack tecnológico. Justificación del mismo.	7
6. Esquema de la BD	9
7. Prototipo en Figma	12
8. Definición API REST publicación servicios	19
9. Manual de despliegue	20
10. Conclusiones y Postmortem	35



1. Introducción

La gestión efectiva de partidos de fútbol, tanto a nivel amateur como organizado, presenta desafíos significativos en cuanto a coordinación, comunicación y organización. En respuesta a estas necesidades, surge Soccer Social Hub, una plataforma diseñada para facilitar la creación, gestión y participación en eventos deportivos. Este proyecto se enfoca en ofrecer una solución integral que mejora la experiencia de los usuarios al proporcionar herramientas eficientes para la administración de partidos, comunicación efectiva entre jugadores y seguridad en la autenticación de usuarios. A través de una combinación cuidadosamente seleccionada de tecnologías y una interfaz intuitiva, Soccer Social Hub se posiciona como una opción innovadora en el mercado, dirigida tanto a organizadores de eventos como a entusiastas del fútbol.



2. Identificación de las necesidades del proyecto

La gestión de partidos de fútbol en grupos, tanto a nivel nacional como internacional, presenta varios desafíos significativos. Estos desafíos incluyen la coordinación de horarios, la inscripción y cancelación de jugadores, la aprobación de nuevos participantes y la organización general de los eventos. Además, la ausencia de una plataforma centralizada complica la comunicación y aumenta la probabilidad de errores y malentendidos. Soccer Social Hub se propone como una solución integral para:

1. Facilitar la creación y gestión de partidos.
2. Mejorar la comunicación entre los jugadores.
3. Proporcionar herramientas administrativas eficientes.
4. Garantizar la seguridad y autenticación de los usuarios.



3. Breve análisis/comparativa con las alternativas del mercado

En el mercado existen varias aplicaciones destinadas a la gestión de actividades deportivas, como *TeamSnap*, *Sportlyzer* y *Spond*. Sin embargo, estas plataformas presentan algunas limitaciones:

TeamSnap: Popular en equipos deportivos, pero con una interfaz que puede ser confusa para nuevos usuarios y funcionalidades premium de alto costo.

Sportlyzer: Ofrece una buena gestión de entrenamientos, pero su enfoque está más orientado hacia equipos organizados y menos hacia la gestión de partidos casuales.

Spond: Buena interfaz de usuario y funcionalidades sociales, pero carece de herramientas avanzadas de control administrativo.

Soccer Social Hub se diferencia al enfocarse específicamente en la gestión de partidos de fútbol, proporcionando una solución más ajustada y con un mejor equilibrio entre facilidad de uso y control administrativo.



4. Justificación del proyecto

El proyecto Soccer Social Hub es justificado por la necesidad de una plataforma específica para la gestión eficiente de partidos de fútbol. La existencia de varias soluciones genéricas en el mercado no aborda de manera óptima las necesidades únicas de este tipo de eventos deportivos. Soccer Social Hub pretende llenar este vacío al ofrecer una interfaz intuitiva, funcionalidad robusta y herramientas de control administrativo, todo centrado en mejorar la experiencia de los usuarios y administradores.



5. Uso de stack tecnológico. Justificación del mismo.

El stack tecnológico utilizado en Soccer Social Hub ha sido cuidadosamente seleccionado para garantizar un equilibrio óptimo entre rendimiento, facilidad de desarrollo y escalabilidad. A continuación se detallan las tecnologías empleadas y la justificación de su elección:

Back-end:

Java con Spring Boot: Utilizado para la lógica del backend debido a su robustez, escalabilidad y amplia adopción en el desarrollo de aplicaciones empresariales. Spring Boot simplifica la configuración y el desarrollo de aplicaciones basadas en Java, permitiendo una implementación rápida y eficiente de la lógica de negocio y operaciones de base de datos, además de exponer una API RESTful.

IntelliJ IDEA: IDE elegido para el desarrollo en Java por su potente conjunto de herramientas y su integración fluida con otros componentes del stack. IntelliJ facilita el desarrollo, depuración y mantenimiento del código, mejorando la productividad del equipo.

Front-end:

Angular 17: Seleccionado para el desarrollo de las vistas interactivas y la conexión con el backend. Angular proporciona una plataforma completa para construir aplicaciones web dinámicas y responsivas, permitiendo un desarrollo estructurado y mantenible.

Ionic: Empleado junto con Angular para desarrollar una interfaz de usuario dinámica y compatible con múltiples plataformas. Ionic facilita la creación de aplicaciones móviles y web que ofrecen una experiencia de usuario consistente y atractiva.

Bootstrap 5: Utilizado para los estilos de los elementos y el diseño del layout grid. Bootstrap asegura una presentación visual atractiva y responsiva, mejorando la usabilidad y la experiencia del usuario final.

Visual Studio Code: Utilizado para el desarrollo en Angular e Ionic debido a su flexibilidad y extensibilidad. Visual Studio Code proporciona un entorno de desarrollo ligero y eficiente, con numerosas extensiones que facilitan el trabajo con tecnologías modernas de frontend.

Base de Datos:



MySQL: Base de datos relacional elegida por su alto rendimiento, escalabilidad y soporte comunitario. MySQL maneja eficientemente el almacenamiento y la gestión de datos, proporcionando una base sólida para las operaciones de la aplicación.

Integración y Flujo de Trabajo:

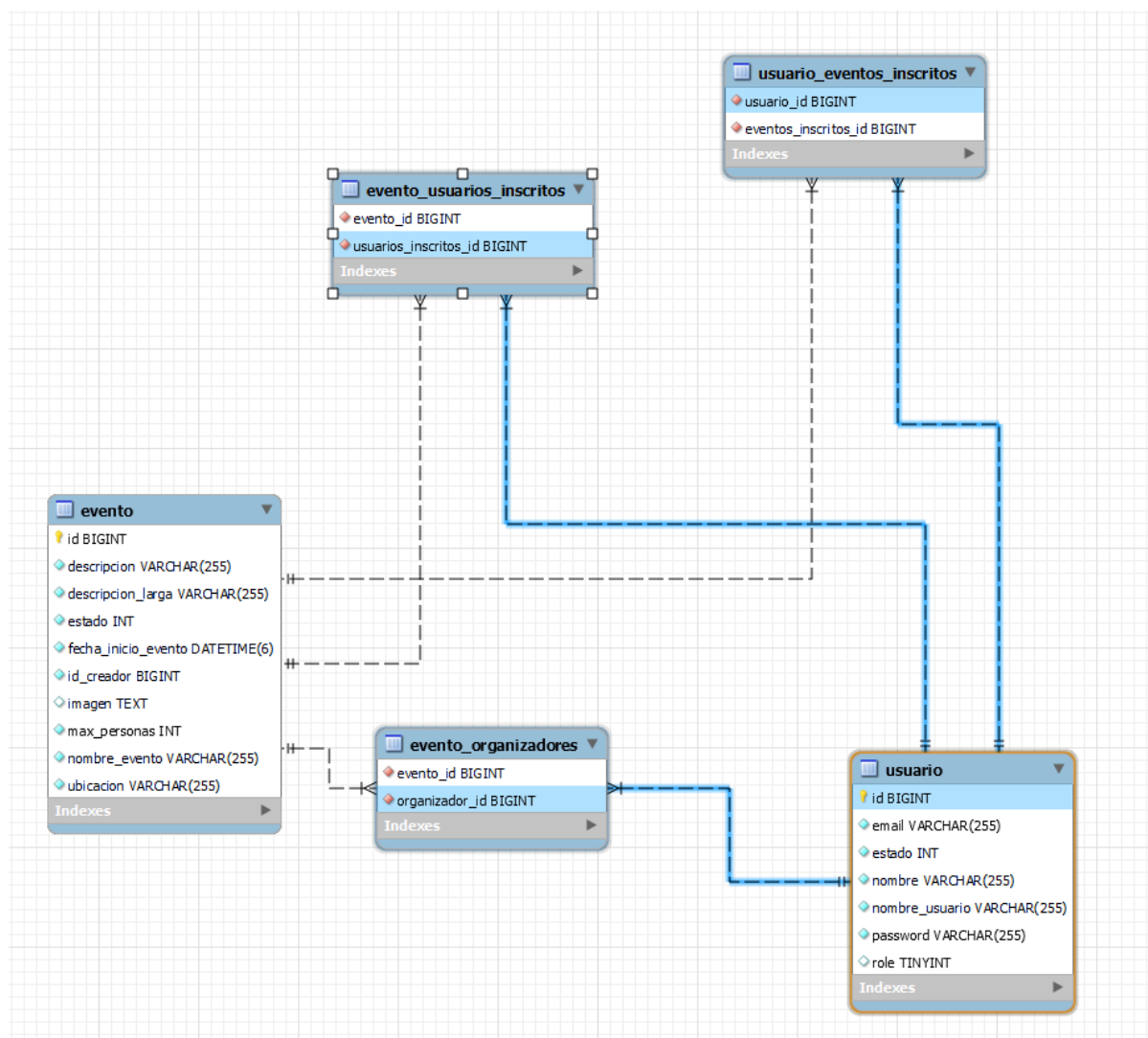
La integración de estas tecnologías ha proporcionado un flujo de trabajo eficiente para el desarrollo de Soccer Social Hub:

1. Spring Boot maneja la lógica del negocio y las operaciones de la base de datos, exponiendo una API RESTful.
2. Angular 17 consume esta API, presentando los datos a los usuarios a través de interfaces dinámicas y responsivas.
3. Ionic asegura que la interfaz de usuario sea compatible con múltiples plataformas, mejorando la accesibilidad.
4. Bootstrap 5 se encarga del diseño y la presentación, asegurando una experiencia de usuario cohesiva y atractiva.
5. MySQL proporciona una solución escalable y de alto rendimiento para la gestión de datos.

Este stack tecnológico ha permitido desarrollar una aplicación robusta, eficiente y fácil de mantener, asegurando que Soccer Social Hub cumpla con los requisitos y expectativas de sus usuarios finales.

6. Esquema de la BD

El esquema de la base de datos de Soccer Social Hub ha sido diseñado para gestionar eficientemente la información relacionada con los eventos (partidos), los usuarios y sus inscripciones. A continuación, se presenta una descripción detallada de cada tabla y sus relaciones, apoyada por el diagrama adjunto.



Descripción de las Tablas:

1. usuario

- id (BIGINT): Identificador único del usuario.
- email (VARCHAR(255)): Dirección de correo electrónico del usuario.



- estado (INT): Estado del usuario (activo, inactivo, etc.).
- nombre (VARCHAR(255)): Nombre del usuario.
- nombre_usuario (VARCHAR(255)): Nombre de usuario para login.
- password (VARCHAR(255)): Contraseña del usuario (almacenada segura).
- role (TINYINT): Rol del usuario (administrador, jugador, etc.).

2. evento

- id (BIGINT): Identificador único del evento.
- descripcion (VARCHAR(255)): Descripción corta del evento.
- descripcion_larga (VARCHAR(255)): Descripción detallada del evento.
- estado (INT): Estado del evento (aceptado, cancelado, por aceptar.).
- fecha_inicio_evento (DATETIME(6)): Fecha y hora de inicio del evento.
- id_creador (BIGINT): Identificador del usuario que creó el evento.
- imagen (TEXT): Imagen asociada al evento.
- max_personas (INT): Número máximo de personas que pueden inscribirse en el evento.
- nombre_evento (VARCHAR(255)): Nombre del evento.
- ubicacion (VARCHAR(255)): Ubicación del evento.

3. evento_organizadores

- evento_id (BIGINT): Identificador del evento.
- organizador_id (BIGINT): Identificador del usuario organizador del evento.

4. evento_usuarios_inscritos

- evento_id (BIGINT): Identificador del evento.
- usuarios_inscritos_id (BIGINT): Identificador del usuario inscrito en el evento.

5. usuario_eventos_inscritos

- usuario_id (BIGINT): Identificador del usuario.
- eventos_inscritos_id (BIGINT): Identificador del evento al que el usuario está inscrito.

Relaciones entre Tablas

La tabla evento se relaciona con la tabla usuario a través de la columna id_creador, indicando el usuario que creó el evento.



La tabla evento_organizadores actúa como una tabla de relación entre evento y usuario, permitiendo la asignación de uno o más organizadores a un evento.

La tabla evento_usuarios_inscritos y usuario_eventos_inscritos gestionan las inscripciones de usuarios a eventos, estableciendo relaciones many-to-many entre usuario y evento.

7. Prototipo en Figma

Vistas en prototipo Figma : Enlace

<https://www.figma.com/design/Fy42KEhfdxayKeTQAcuZRI/TFG-SoccerSocialClub?node-id=117-1748&t=FfX8wXFPcj5jKELH-1>

1. Vista Login :

La vista de inicio de sesión permite a los usuarios autenticarse en la aplicación. Incluye campos para el correo electrónico y la contraseña, así como un botón para iniciar sesión.



2. Vista Registro:

La vista de registro permite a los nuevos usuarios crear una cuenta. Los campos incluyen correo electrónico, contraseña, nombre de usuario. Un botón de registro completa el proceso.



3. Vista Página Principal User:

La vista principal del usuario muestra los partidos disponibles y a los que el usuario está inscrito. Cada partido se presenta con detalles como el nombre, la descripción, la fecha y las opciones para apuntarse o desapuntarse, también un botón de ver más.



4. Vista Página Principal Admin:

La vista principal del administrador proporciona dos botones para acceder al panel de control con acceso a la gestión de usuarios y eventos. Los administradores pueden aprobar o rechazar nuevos partidos y usuarios.



5. Vista crear partido:

Esta vista permite a los organizadores crear nuevos partidos. Incluye campos para el nombre del partido, descripción, fecha de inicio, ubicación y la capacidad de agregar organizadores.

**Nombre del Partido****Descripción Breve**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt

Organizadores

Abel García

Descripción del partido

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Fecha Inicio

15/05/2024

Crear Partido



6. Vista administrador control usuarios:

La vista de control de usuarios permite a los administradores gestionar los usuarios registrados. Incluye opciones para aprobar y rechazar usuarios con su respectivo motivo.



7. Vista administrador control eventos deportivos:

La vista de control de eventos deportivos proporciona a los administradores herramientas para gestionar los eventos creados. Incluye opciones para aprobar y rechazar eventos con su respectivo motivo.





8. Definición API REST publicación servicios

Enlace documentación API REST:

<https://documenter.getpostman.com/view/32189101/2sA3XPC2vw>



9. Manual de despliegue

Primeros pasos antes de iniciar el laboratorio de AWS.

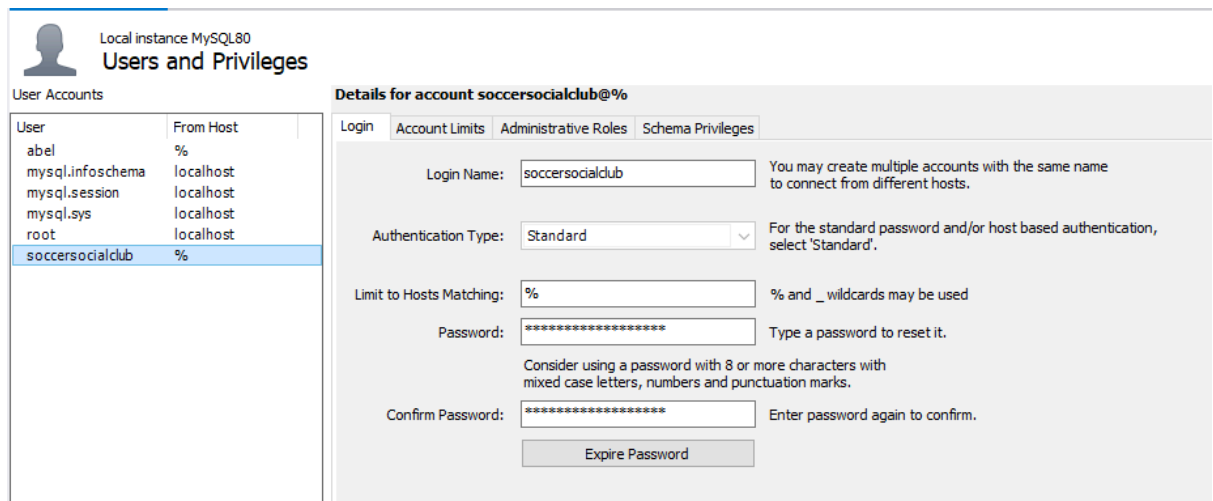
1. Configuración de CORS para dar acceso tanto al puerto 8100 en local como a nuestra ruta de elastic beanstalk.

```
1 package com.project.SoccerSocialClub.SoccerSocialClub.Config;
2
3 import ...
4
5
6
7 @Configuration
8 public class WebConfig implements WebMvcConfigurer {
9
10     @Override
11     public void addCorsMappings(CorsRegistry registry) {
12         registry.addMapping(pathPattern: "/api/**")
13             .allowedOrigins("http://localhost:8100", "http://elasticbeanstalk-us-east-1-300878465235.s3-website-us-east-1.amazonaws.com")
14             .allowedMethods("GET", "POST", "PUT", "DELETE", "OPTIONS")
15             .allowedHeaders("*")
16             .allowCredentials(true);
17     }
18 }
19 }
```

2. Variables de entorno, creamos a la altura de la raíz del proyecto el fichero .env e añadimos las siguientes variables.

```
src > .env
You, hace 3 días | 1 autor (You)
1 MYSQL_ROOT_PASSWORD=soccersocialclub12345
2 MYSQL_DATABASE=soccersocial
3 MYSQL_USER=soccersocialclub
4 MYSQL_PASSWORD=soccersocialclub12345
5 MYSQL_HOST=localhost
6 MYSQL_PORT=3306
7 CORS_ALLOWED_ORIGIN=http://localhost:8100
8
9
```

3. Creamos un usuario en MySQL Workbench al que asignamos permisos de administrador, más tarde lo usaremos para la conexión con RDS (contraseña mínimo 8 caracteres).



4. Configuración Spring Boot y empaquetado de la aplicación en archivo .jar utilizando el comando 'mvn clean install -DskipTest'

```
Abel@ABEL MINGW64 ~/Desktop/TFG/src/SoccerSocialClub (master)
$ mvn clean install -DskipTests
```

Aquí podemos ver como se creo en la carpeta target correctamente :

```
Abel@ABEL MINGW64 ~/Desktop/TFG/src/SoccerSocialClub/target (master)
$ ls
SoccerSocialClub-0.0.1-SNAPSHOT.jar SoccerSocialClub-0.0.1-SNAPSHOT.jar.original
```

5. Configuración Angular, creación de carpeta environments donde dentro se crearán dos archivos, el primero de ellos environments.ts que será el archivo configurado para la ejecución en local y en segundo lugar el environments.prod.ts que será para cuando la aplicación se despliegue a internet en este caso con AWS.

Archivo environments.prod.ts



```
src > SoccerSocialClubFront > src > environments > TS environment.prod.ts > ...
You, hace 5 días | 1 author (You)
1 export const environment = {
2   production: true,
3   apiURL: 'http://soccersocialclub.us-east-1.elasticbeanstalk.com/api'
4 };
5
```

Archivo environments.ts

```
src > SoccerSocialClubFront > src > environments > TS environment.ts > ...
...
1
2
3 export const environment = {
4   production: false,
5   apiURL: 'http://localhost:8081/api'
6 };
7
```

A continuación, usamos la variable apiURL en todas las llamadas a los endpoints del BackEnd en los archivos necesarios, importamos el environment damos uso a apiURL como aquí podemos comprobar.

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { environment } from 'src/environments/environment';
import { RegisterInterface } from '../interfaces/register-interface';

You, hace 6 días | 1 author (You)
@Injectable({
  providedIn: 'root'
})
export class RegisterService {

  constructor(private http : HttpClient) { }

  public register(usu : RegisterInterface) : Observable<any> {
    return this.http.post(`${environment.apiURL}/auth/register`, usu);
  }
}
```

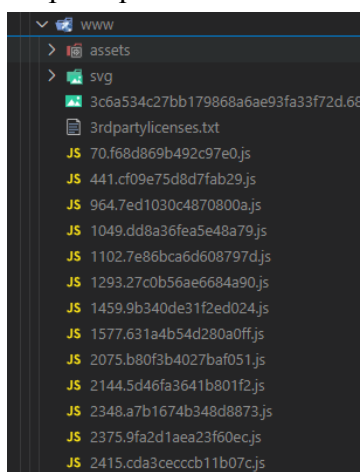
Por último vamos al archivo angular.json añadimos lo siguiente. Una vez añadido empaquetamos con 'ng build --configuration production' y se creará la carpeta 'www' que más tarde usaremos para desplegar el FrontEnd.

```
"configurations": {
  "production": {
    "budgets": [
      {
        "type": "initial",
        "maximumWarning": "2mb",
        "maximumError": "5mb"
      },
      {
        "type": "anyComponentStyle",
        "maximumWarning": "2kb",
        "maximumError": "4kb"
      }
    ],
    "fileReplacements": [
      {
        "replace": "src/environments/environment.ts",
        "with": "src/environments/environment.prod.ts"
      }
    ]
  }
}
```

Comando para compilar la parte del FrontEnd.

```
PS C:\Users\Abel\Desktop\TFG\src\SoccerSocialClubFront> ng build --configuration production
```

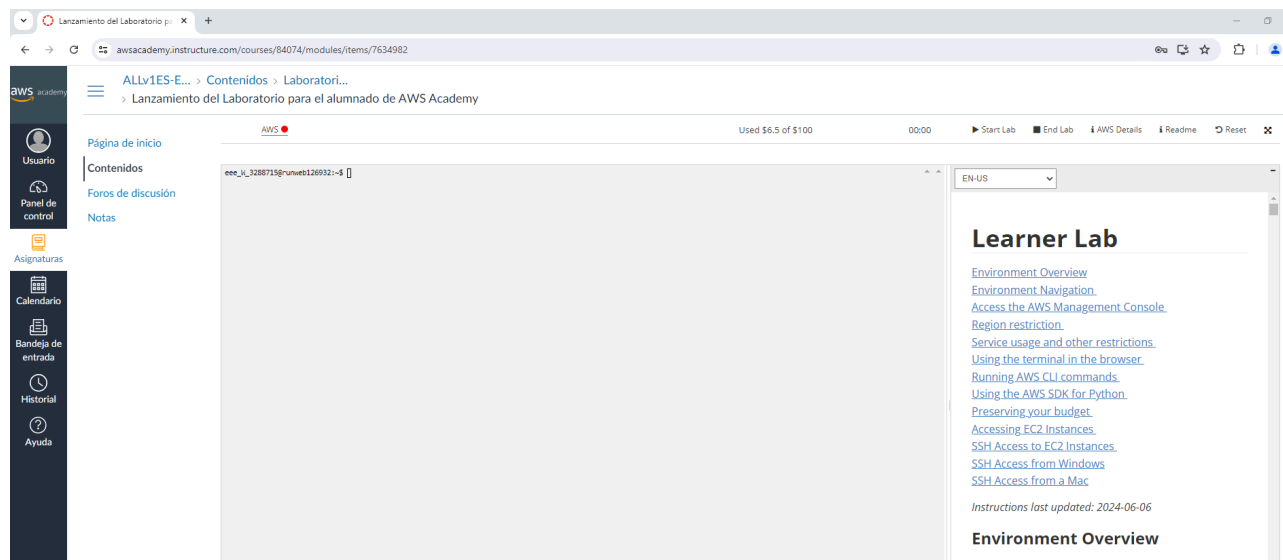
Carpeta que se creara.





Despliegue en AWS.

Accedemos a AWS Academy e iniciamos sesión con nuestra cuenta y buscamos la opción de ‘AWS Academy Learner Lab’ accedemos a módulos e iniciamos el laboratorio.



Accedemos a Elastic Beanstalk en AWS y creamos un nuevo entorno. Entraremos en la parte de la configuración del entorno. Le añadimos un nombre y un dominio.



Paso 1
Configuración del entorno

Paso 2
Configuración del acceso al servicio

Paso 3 - opcional
Configuración de redes, bases de datos y etiquetas

Paso 4 - opcional
Configuración del escalado y del tráfico de instancias

Paso 5 - opcional
Configuración de actualizaciones, monitoreo y registros

Paso 6
Revisión

Configuración del entorno Información

Nivel de entorno Información
Amazon Elastic Beanstalk tiene dos tipos de niveles de entorno para admitir diferentes tipos de aplicaciones web.

☒ **Entorno de servidor web**
Ejecute un sitio web, una aplicación web o una API web que atienda solicitudes HTTP. [Más información](#)

☐ **Entorno de trabajo**
Ejecute una aplicación de proceso de trabajo que procese cargas de trabajo de ejecución prolongada bajo demanda o realice tareas de forma programada. [Más información](#)

Información de la aplicación Información

Nombre de aplicación

La longitud máxima es de 100 caracteres.

► **Etiquetas de aplicación (opcional)**

Información del entorno Información
Elija el nombre, el subdominio y la descripción del entorno. No se pueden cambiar más adelante.

Nombre del entorno

Debe tener entre 4 y 40 caracteres. El nombre solo puede contener letras, números y guiones. No puede comenzar ni terminar por un guion. Este nombre debe ser único dentro de una región de su cuenta.

Dominio
 .us-east-1.elasticbeanstalk.com

Verificar disponibilidad

En el apartado más abajo seleccionamos cargar código y la plataforma Java con la versión que tengamos, en mi caso la 21 por último cargamos el .jar generado en apartados anteriormente.

Plataforma

Ramificación de la plataforma

Versión de la plataforma

Código de aplicación Información

☐ Aplicación de ejemplo

☒ **Versión existente**
Versiones de la aplicación que ha cargado.

☒ **Cargar el código**
Cargue un paquete de código fuente desde su equipo o copie uno desde Amazon S3.

Etiqueta de versión
Nombre único para esta versión del código de la aplicación.

⚠ La etiqueta de la versión de la aplicación es obligatoria

Origen del código fuente. Tamaño máximo de 500 MB

☒ **Archivo local**

Cargar aplicación

El archivo debe tener un tamaño máximo de archivo inferior a 500 MB

☐ URL pública de S3

Abrir

Este equipo > Escritorio > TFG > src > SoccerSocialClub > target

Organizar	Nueva carpeta	Nombre	Fecha de modificación	Tipo	Tamaño
		clases	14/06/2024 14:09	Carpeta de archivos	
		generated-sources	14/06/2024 14:09	Carpeta de archivos	
		generated-test-sources	14/06/2024 14:09	Carpeta de archivos	
		maven-archiver	14/06/2024 14:09	Carpeta de archivos	
		maven-status	14/06/2024 14:09	Carpeta de archivos	
		test-classes	14/06/2024 14:09	Carpeta de archivos	
		SoccerSocialClub-0.0.1-SNAPSHOT.jar	14/06/2024 14:09	Executable Jar File	60.901 KB
		SoccerSocialClub-0.0.1-SNAPSHOT.jar.orig...	14/06/2024 14:09	Archivo ORIGINAL	57 KB

Nombre: SoccerSocialClub-0.0.1-SNAPSHOT.jar

Todos los archivos (*.*)

Abrir **Cancelar**

Pasamos al paso dos Configuración del acceso al servicio, aquí añadimos el rol de LabRole, la clave vockey y el perfil LabInstanceProfile.



Paso 1
[Configuración del entorno](#)

Paso 2
Configuración del acceso al servicio

Paso 3 - opcional
[Configuración de redes, bases de datos y etiquetas](#)

Paso 4 - opcional
[Configuración del escalado y del tráfico de instancias](#)

Paso 5 - opcional
[Configuración de actualizaciones, monitoreo y registros](#)

Paso 6
[Revisión](#)

Configuración del acceso al servicio Información

Acceso al servicio

Los roles de IAM, asumidos por Elastic Beanstalk como rol de servicio, y los perfiles de instancia de EC2 permiten a Elastic Beanstalk crear y administrar su entorno. Tanto el rol de IAM como el perfil de instancia deben estar asociados a políticas administradas de IAM que contengan los permisos necesarios. [Más información](#)

Rol de servicio

☐ Crear y utilizar un nuevo rol de servicio

☒ Usar un rol de servicio existente

Roles de servicio existentes

Elija un rol de IAM existente para que Elastic Beanstalk asuma como rol de servicio. El rol de IAM existente debe tener las políticas administradas de IAM necesarias.

LabRole

Par de claves de EC2

Seleccione un par de claves de EC2 para iniciar sesión de forma segura en sus instancias de EC2. [Más información](#)

vockey

Perfil de instancia de EC2

Elija un perfil de instancia de IAM con políticas administradas que permitan a las instancias de EC2 realizar las operaciones necesarias.

LabInstanceProfile

Ver los detalles de los permisos

Cancelar

Ir a revisión

Anterior

Siguiente

Al darle a siguiente pasaremos al paso 3 en el que VPC lo dejamos por defecto en la parte inferior configuramos la base de datos usando los datos de sesión de mysql y del archivo .env antes configurado.



Configuración de base de datos

Elija un motor y un tipo de instancia para la base de datos del entorno.

Motor

mysql

Versión del motor

8.0.35

Clase de instancia

db.t3.small

Almacenamiento

Elija un número comprendido entre 5 GB y 1024 GB.

5

GB

Nombre de usuario

soccersocialclub

Contraseña

.....

Disponibilidad

Baja (una AZ)

Política de eliminación de bases de datos

Esta política se aplica al desacoplar una base de datos o al terminar el entorno asociado a ella.

☐ Crear instantánea

Elastic Beanstalk guarda una instantánea de la base de datos y, a continuación, la elimina. Puede restaurar una base de datos a partir de una instantánea al agregar una base de datos a un entorno de Elastic Beanstalk o al crear una base de datos independiente. Es posible que se apliquen cargos por almacenar instantáneas de base de datos.

☐ Retener

La base de datos desacoplada seguirá estando disponible y operativa fuera de Elastic Beanstalk.

☒ Eliminar

Elastic Beanstalk termina la base de datos. La base de datos dejará de estar disponible.

El paso 4 lo dejamos todo por defecto y pasamos al 5 en el que tendremos que crear las siguientes propiedades de entorno que aún no son definitivas.



Propiedades del entorno

Las siguientes propiedades se transfieren en la aplicación como propiedades de entorno. [Más información](#)

Nombre	Valor	
GRADLE_HOME	/usr/local/gradle	Eliminar
M2	/usr/local/apache-maven/bin	Eliminar
M2_HOME	/usr/local/apache-maven	Eliminar
CORS_ALLOWED_ORIGIN	http://localhost:8100	Eliminar
MYSQL_DATABASE	soccersocial	Eliminar
MYSQL_HOST	127.0.0.1	Eliminar
MYSQL_PASSWORD	soccersocialclub12345	Eliminar
MYSQL_PORT	3306	Eliminar
MYSQL_USER	soccersocialclub	Eliminar
SERVER_PORT	5000	Eliminar
Agregar propiedad de entorno		

Cancelar

Anterior

Siguiente

Entramos en RDS y vemos cómo se creó una instancia clicamos en ella y accedemos a editar las reglas de entrada donde añadiremos las siguientes reglas.



EC2 > Grupos de seguridad > sg-0a2a0e39c98360083 - rds-awseb-e-czunnpagpz-stack-awsebrdsdbsecuritygroup-1f5g1scag3one-8tbf > Editar reglas de entrada

Editar reglas de entrada Información

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

Reglas de entrada Información

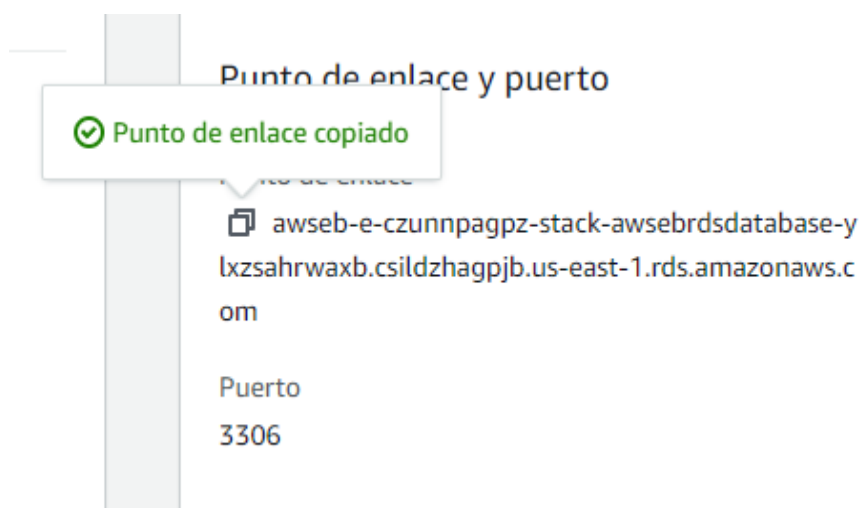
ID de la regla del grupo de seguridad	Tipo <small>Información</small>	Protocolo <small>Información</small>	Intervalo de puertos <small>Información</small>	Origen <small>Información</small>	Descripción: opcional <small>Información</small>	
sg-0413f965abaf99758	SSH	TCP	22	Personalizada	Q 0.0.0.0/0 X	Eliminar
sg-01e7516eb2035867d	MySQL/Aurora	TCP	3306	Personalizada	Q 188.26.220.57/32 X	Eliminar
sg-02969fc71d7d00bae	Todos los TCP	TCP	0 - 65535	Personalizada	Q 0.0.0.0/0 X	Eliminar

[Agregar regla](#)

⚠ Las reglas con el origen 0.0.0.0/0 o ::/0 permiten que todas las direcciones IP tengan acceso a la instancia. Le recomendamos que configure las reglas del grupo de seguridad para permitir el acceso únicamente desde direcciones IP conocidas.

[Cancelar](#) [Previsualizar los cambios](#) [Guardar reglas](#)

Pasamos a la conexión con la base de datos Mysql para ello copiamos el punto de enlace de nuestro RDS.



Accedemos a Mysql Workbench y le damos a una nueva conexión donde pegamos el punto de enlace de nuestro RDS, rellenamos los campos de contraseña y usuario estos tienen que coincidir con los anteriores usados para que tengamos éxito al establecer la conexión.



Manage Server Connections

MySQL Connections

- Local instance MySQL80
- AWS-RDS

Connection Name: AWS-RDS

Connection Remote Management System Profile

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: awseb-e-czunpagpz-stack- Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: soccersocialclub Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

New Delete Duplicate Move Up Move Down Test Connection Close

Antes de crear nuestra base de datos tendremos que ir de nuevo a propiedades de entorno y cambiar el MYSQL_HOST por el punto de enlace de RDS.



Propiedades del entorno

Las siguientes propiedades se transfieren en la aplicación como propiedades de entorno. [Más información](#)

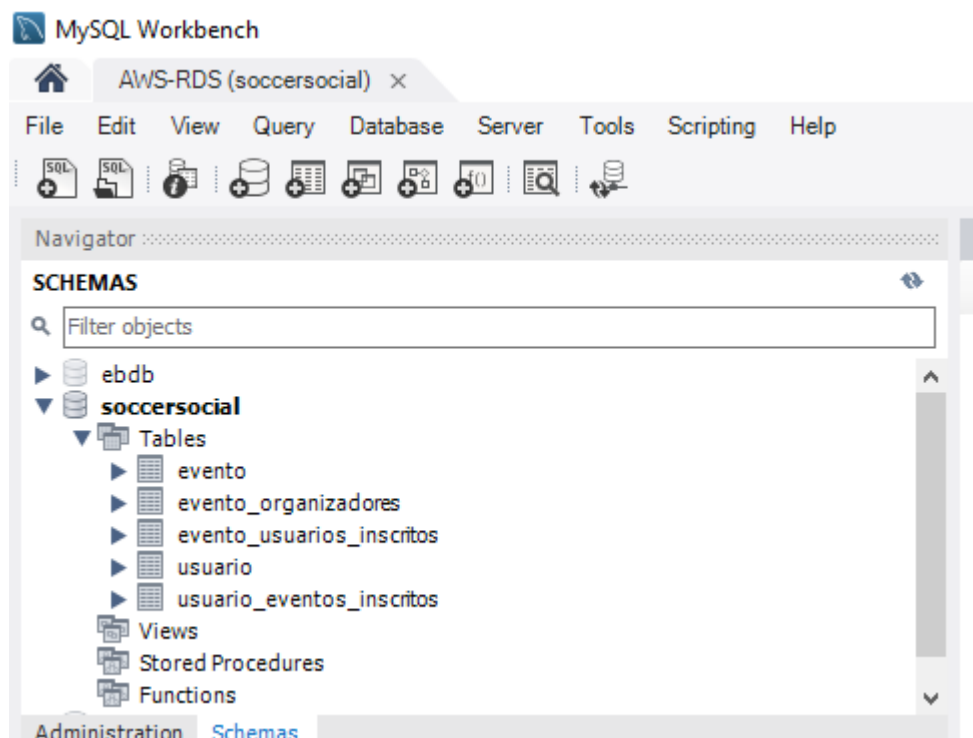
Nombre	Valor	
AWS_EXECUTION_ENV	true	Eliminar
CORS_ALLOWED_ORIGIN	http://elasticbeanstalk-us-east-1-30087	Eliminar
GRADLE_HOME	/usr/local/gradle	Eliminar
M2	/usr/local/apache-maven/bin	Eliminar
M2_HOME	/usr/local/apache-maven	Eliminar
MYSQL_DATABASE	soccersocial	Eliminar
MYSQL_HOST	awseb-e-czunnpagpz-stack-awsebrdsdal	Eliminar
MYSQL_PASSWORD	soccersocialclub12345	Eliminar
MYSQL_PORT	3306	Eliminar
MYSQL_USER	soccersocialclub	Eliminar
SERVER_PORT	5000	Eliminar
Agregar propiedad de entorno		

Cancelar

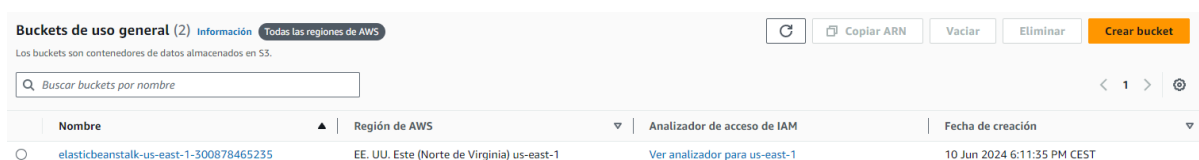
Continuar

Aplicar

Ahora ya al establecer la conexión y crear nuestra base de datos al recargar se crearán las tablas solas ya que nos pillara el .jar antes cargado del BackEnd.



Despliegue parte FrontEnd accedemos a S3 y entramos al bucket creado por Elastic Beanstalk.



En las propiedades editamos a alojamiento de sitios web estáticos, habilitamos esta opción con la siguiente configuración.



[Amazon S3](#) > [Buckets](#) > [elasticbeanstalk-us-east-1-300878465235](#) > Editar alojamiento de sitios web estáticos

Editar alojamiento de sitios web estáticos [Información](#)

Alojamiento de sitios web estáticos

Utilice este bucket para alojar un sitio web o redirigir solicitudes. [Más información](#)

Alojamiento de sitios web estáticos

- ☐ Desactivar
☒ Habilitar

Tipo de alojamiento

- ☒ Alojamiento de un sitio web estático
Utilice el punto de enlace del bucket como dirección web. [Más información](#)
- ☐ Redirigir las solicitudes de un objeto
Redirija las solicitudes a otro bucket o dominio. [Más información](#)

Para que sus clientes puedan obtener acceso al contenido en el punto de enlace del sitio web, debe hacer que todo el contenido sea legible públicamente. Para ello, puede editar la configuración Bloquear acceso público de S3 del bucket. Para obtener más información, consulte [Utilizar Bloquear acceso público de Amazon S3](#)

Documento de índice

Especifique la página predeterminada o de inicio del sitio web.

index.html

Documento de error - *opcional*

Esto se devuelve cuando se produce un error.

index.html

Reglas de redireccionamiento: *opcionales*

Accedemos a la pestaña de Permisos y añadimos la siguiente política de bucket.

Política de bucket

La política del bucket, escrita en JSON, proporciona acceso a los objetos almacenados en el bucket. Las políticas de bucket no se aplican a los objetos que pertenecen a otras cuentas. [Más información](#)

Editar

Eliminar

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "eb-af163bf3-d27b-4712-b795-d1e33e331ca4",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::300878465235:role/LabRole"
      },
      "Action": [
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::elasticbeanstalk-us-east-1-300878465235",
        "arn:aws:s3:::elasticbeanstalk-us-east-1-300878465235/resources/environments/"
      ]
    },
    {
      "Sid": "eb-58950a8c-feb6-11e2-89e0-0800277d041b",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:DeleteBucket",
      "Resource": "arn:aws:s3:::elasticbeanstalk-us-east-1-300878465235"
    }
  ]
}
```



Por último, pasamos al apartado de objetos donde subiremos el contenido de la carpeta ‘www’ y una vez subido marcaremos todos los archivos, en el botón de acciones seleccionaremos la opción de hacer público mediante ACL.

elasticbeanstalk-us-east-1-300878465235 [Información](#)

[Objetos](#) [Propiedades](#) [Permisos](#) [Métricas](#) [Administración](#) [Puntos de acceso](#)

Objetos (88) [Información](#) [Copiar URI de S3](#) [Copiar URL](#) [Descargar](#) [Abrir](#) [Eliminar](#) [Acciones](#) [Crear carpeta](#) [Cargar](#)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas puedan acceder a los objetos, puede configurar los permisos de forma explícita. [Más información](#)

🔍 Buscar objetos por prefijo

<input checked="" type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño
<input checked="" type="checkbox"/>	elasticbeanstalk	elasticbeanstalk	10 Jun 2024 7:28:35 PM CEST	
<input checked="" type="checkbox"/>	1049.dd8a36fea5e48a79.js	js	14 Jun 2024 2:32:04 PM CEST	
<input checked="" type="checkbox"/>	1102.7e86bca6d608797d.js	js	14 Jun 2024 2:32:04 PM CEST	
<input checked="" type="checkbox"/>	1293.27c0b56ae6684a90.js	js	14 Jun 2024 2:32:05 PM CEST	
<input checked="" type="checkbox"/>	1459.9b340de31f2ed024.js	js	14 Jun 2024 2:32:06 PM CEST	
<input checked="" type="checkbox"/>	1577.631a4b54d280a0ff.js	js	14 Jun 2024 2:32:06 PM CEST	
<input checked="" type="checkbox"/>	1718367194247-SoccerSocialClub-0.0.1-SNAPSHOT.jar	jar	14 Jun 2024 2:13:15 PM CEST	
<input checked="" type="checkbox"/>	2075.b80f3b4027baf051.js	js	14 Jun 2024 2:32:07 PM CEST	
<input checked="" type="checkbox"/>	2144.5d46fa5641b801f2.js	js	14 Jun 2024 2:32:07 PM CEST	
<input checked="" type="checkbox"/>	2348.a7b1674b548d8873.js	js	14 Jun 2024 2:32:08 PM CEST	
<input checked="" type="checkbox"/>	2375.9fa2d1aea23f60ec.js	js	14 Jun 2024 2:32:08 PM CEST	
<input checked="" type="checkbox"/>	2415.cda3cecccb11b07c.js	js	14 Jun 2024 2:32:09 PM CEST	
<input checked="" type="checkbox"/>	2560.3395abbf4641a48f.js	js	14 Jun 2024 2:32:09 PM CEST	
<input checked="" type="checkbox"/>	2885.81cea01698112be9.js	js	14 Jun 2024 2:32:10 PM CEST	
<input checked="" type="checkbox"/>	3162.900d552025a90c33.js	js	14 Jun 2024 2:32:10 PM CEST	
<input checked="" type="checkbox"/>	3506.6d48a737ab8c946772.ic	ic	14 Jun 2024 2:32:11 PM CEST	

Acciones

- Descargar como
- Compartir con una URL prefirmada
- Calcular el tamaño total
- Copiar
- Trasladar
- Iniciar restauración
- Consultar con S3 Select
- Editar acciones
 - Cambiar el nombre del objeto
 - Editar clase de almacenamiento
 - Editar cifrado del lado del servidor
 - Editar metadatos
 - Editar etiquetas
 - Hacer público mediante ACL

Tamaño	Acción
46.0 KB	Estándar
11.5 KB	Estándar
19.7 KB	Estándar
27.2 KB	Estándar
72.0 KB	Estándar
12.0 KB	Estándar
5.4 KB	Estándar

Una vez terminados estos pasos ya tendremos terminado el despliegue correctamente y podremos acceder mediante el enlace de elasticbeanstalk a nuestro proyecto.

Enlace al proyecto : [SoccerSocialHub](#)



10. Conclusiones y Postmortem

El desarrollo del proyecto Soccer Social Hub ha sido un proceso muy bueno que ha permitido abordar diversas necesidades en la gestión de partidos de fútbol. A continuación, se presentan las conclusiones y reflexiones obtenidas durante el desarrollo del proyecto, así como un análisis postmortem de los aspectos destacados:

Conclusiones:

Satisfacción de Necesidades: Soccer Social Hub ha logrado satisfacer las necesidades identificadas en la gestión de partidos de fútbol, proporcionando una plataforma integral para la creación, organización y comunicación de eventos deportivos.

Diferenciación en el Mercado: La comparativa con alternativas existentes resalta la capacidad de Soccer Social Hub para ofrecer una solución más ajustada y equilibrada entre facilidad de uso y control administrativo, lo que le otorga una ventaja competitiva en el mercado.

Tecnología Apropiaada: La elección del stack tecnológico ha sido acertada, permitiendo el desarrollo de una aplicación robusta, eficiente y fácil de mantener. La integración de tecnologías como Java, Angular, Ionic y MySQL ha proporcionado un flujo de trabajo eficiente y una experiencia de usuario consistente.

Prototipo Funcional: El prototipo desarrollado en Figma ha servido como una herramienta invaluable para visualizar y validar el diseño de la interfaz de usuario, permitiendo iteraciones rápidas y mejoras basadas en el feedback recibido.

API REST Documentada: La documentación clara y detallada de la API REST facilita la integración de Soccer Social Hub con otras aplicaciones y servicios, permitiendo una mayor interoperabilidad y expansión futura.

Postmortem:



Lecciones Aprendidas: Durante el desarrollo del proyecto, se han identificado y aprendido diversas lecciones relacionadas con la gestión de proyectos, la elección de tecnologías y la comunicación efectiva entre el equipo de desarrollo.

Feedback de Usuarios: Se ha recopilado feedback de usuarios durante el proceso de desarrollo, lo que ha permitido realizar ajustes y mejoras en la plataforma para satisfacer mejor sus necesidades y expectativas.

Desafíos Superados: Se han enfrentado diversos desafíos técnicos y de diseño a lo largo del desarrollo del proyecto, como la integración de tecnologías frontend y backend, la gestión de la base de datos y la creación de una interfaz de usuario intuitiva.

Éxito en la Entrega: El proyecto ha sido entregado dentro del plazo establecido y ha cumplido con los objetivos definidos inicialmente, lo que demuestra el trabajo efectivo.

En resumen, Soccer Social Hub ha sido un proyecto exitoso que ha logrado satisfacer las necesidades de los usuarios y destacarse en el mercado de la gestión de partidos de fútbol. El análisis postmortem realizado permitirá aprender de esta experiencia y aplicar las lecciones aprendidas en futuros proyectos.