



La Grada - Memoria
Bruno Delgado Herrero

1. Introducción

La Grada es un bar con una manera innovadora de vivir el deporte en directo. Este proyecto se basa en la creación de una API REST y una aplicación web que la utiliza, de forma que permite a los usuarios comprar entradas para asistir a estos eventos y a los administradores llevar la gestión de partidos, eventos, y usuarios.

2. Plan de empresa

PLAN DE EMPRESA BRUNO DELGADO FINAL

3. Descripción del sistema

1. Frontend (Angular)

- Tecnologías:

- Angular 18.2.0
- Angular Material 18.2.14
- TypeScript 5.5.2
- RxJS 7.8.0
- JWT para autenticación

- Estructura:
 - Arquitectura modular
 - Componentes reutilizables
 - Servicios para lógica de negocio
 - Sistema de rutas
 - Integración con Material Design

2. Backend (Spring Boot)

- Tecnologías:
 - Spring Boot 3.4.2
 - Java 17
 - Spring Security
 - Spring Data JPA
 - PostgreSQL
 - JWT para autenticación
 - Lombok
 - SpringDoc OpenAPI (documentación API)
 - ZXing (generación de códigos QR)
 - Jakarta Mail (envío de emails)

- Estructura:

```
src/main/java/com/triana/salesianos/dam/lagrada/  
├── config/      # Configuraciones de Spring  
├── controller/  # Controladores REST  
├── dto/         # Objetos de transferencia de datos  
├── model/       # Entidades JPA  
├── repo/        # Repositorios JPA  
├── security/    # Configuración de seguridad  
├── service/     # Lógica de negocio  
├── util/        # Utilidades  
├── validation/  # Validaciones personalizadas  
└── error/       # Manejo de errores
```

- Características:
 - API RESTful
 - Autenticación JWT
 - Validación de datos
 - Documentación OpenAPI
 - Manejo de archivos (uploads/)
 - Generación de códigos QR

- Envío de emails
- Base de datos PostgreSQL
- Docker support (Dockerfile y docker-compose.yml)

3. Infraestructura

- Base de Datos: PostgreSQL
- Contenedorización: Docker
- Documentación API: OpenAPI/Swagger
- Testing: JUnit, Mockito

4. Características del Sistema

1. Seguridad:

- Autenticación JWT
- Spring Security
- Validación de datos
- Manejo de errores personalizado

2. Funcionalidades:

- Gestión de usuarios
- Manejo de archivos
- Generación de códigos QR
- Sistema de emails
- API RESTful documentada

3. Desarrollo:

- Soporte para desarrollo local con H2
- Configuración Docker para producción
- Testing automatizado
- Documentación API integrada

4. Integración:

- Frontend y Backend completamente separados
- Comunicación vía API REST
- Autenticación JWT compartida
- Manejo de CORS configurado

Historias de Usuario

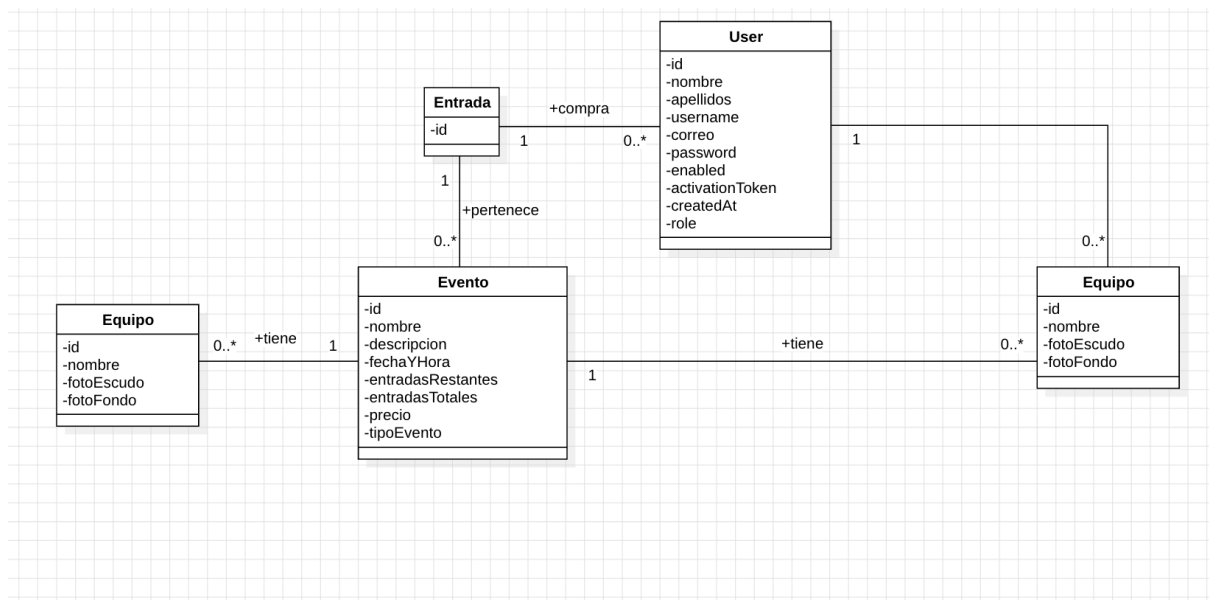
<https://github.com/users/brrrr1/projects/5/views/1>

4. Modelado y diseño

4.1 Prototipado en Figma de la aplicación

<https://www.figma.com/design/42QDQxnjjTn9qP54kUGjZA/Figma-Web-App?node-id=0-1&t=5orGPFkSTgjSKdth-1>

4.2 Diagrama de clases del modelo de dominio de la API.



5. Diseño

5.1 Diagrama de clases del diseño

