



I.E.S. Juan Bosco – Alcázar de San Juan

DOCUMENTO PROYECTO FINAL

VotiumAPP

Daniel Porras Morales

**Ciclo Formativo de Grado Superior
Desarrollo de Aplicaciones Multiplataforma
Curso 2024/2025
Periodo Ordinario: Abril - Junio**

FICHA DEL PROYECTO	
Nombre del Alumno	
Daniel Porras Morales	
Título del Proyecto	
Votium App	
Fechas TUTORÍAS	Entregas / Revisiones
7 abril	Entrega Ficha del Proyecto Apartado 1
25 abril	Apartados 1 y 2 Completos
5 mayo	Apartados 1, 2, 3 y 4
19 mayo	Apartados 1, 2, 3, 4, 5 y 6
2 junio	Todos los Apartados
6 junio ENTREGA FINAL	Documentación Completa y Proyecto Completo

PLATAFORMAS / LENGUAJES / MOTOR B.D.

Java con Spring Boot, Oracle Database.

BREVE DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en el desarrollo de una aplicación que permite la creación de eventos personalizados, dentro de los cuales los usuarios pueden realizar predicciones o "apuestas amistosas" sobre lo que ocurrirá durante el evento. Está pensada para ser utilizada entre grupos de amigos de forma privada y sin implicaciones económicas, fomentando la participación y el entretenimiento en eventos como la Velada del Año de Ibai, competiciones deportivas, galas de premios, concursos musicales o cualquier otro tipo de acontecimiento compartido.

DECLARACIÓN DE ORIGINALIDAD

Declaro:

Que he realizado este trabajo de forma autónoma, sin copiar ni usar contenidos de otras fuentes (impresas o digitales) sin citarlas correctamente, tanto en el texto como en la bibliografía.

Comprendo:

Que el plagio —copiar ideas, textos o datos sin citar su origen y presentarlos como propios— es un delito contra la propiedad intelectual y será calificado automáticamente con Suspense, además de posibles sanciones académicas.

Acepto:

Que este proyecto puede utilizarse con fines educativos, citando al autor original (licencia CC BY), como material para los ciclos de Informática del IES Juan Bosco.

Fecha

Firma

16 de Abril de 2025



I. Introducción.....	2
a. Presentación del Proyecto.....	2
b. Objetivos del Proyecto.....	2
c. Justificación del Proyecto.....	2
II. Análisis de Requerimientos.....	2
a. Identificación de Necesidades y Requerimientos.....	2
b. Identificación del Público objetivo.....	2
c. Estudio de Mercado y Competencia.....	2
III. Diseño y Planificación.....	2
a. Definición de la Arquitectura del Proyecto.....	2
b. Diseño de la Interfaz de Usuario.....	2
c. Planificación de las Tareas y los Recursos necesarios.....	2
IV. Implementación y Pruebas.....	2
a. Desarrollo de las Funcionalidades del Proyecto.....	2
b. Pruebas Unitarias y de Integración.....	2
c. Corrección de Errores y Optimización del Rendimiento.....	2
V. Documentación.....	2
a. Documentación Técnica.....	2
b. Documentación de Usuario Final.....	2
c. Manual de Instalación y Configuración.....	2
VI. Mantenimiento y Evolución.....	3
a. Identificación de Posibles Mejoras y Evolución del Proyecto.....	3
b. Actualizaciones y Mejoras Futuras.....	3
VII. Conclusiones.....	3
a. Evaluación del Proyecto.....	3
b. Cumplimiento de Objetivos y Requisitos.....	3
c. Lecciones aprendidas y Recomendaciones para Futuros Proyectos.....	3
VIII. Bibliografía/Webgrafía y/o Referencias.....	3
a. Fuentes utilizadas en el Proyecto.....	3

I. Introducción

a. Presentación del Proyecto

Este proyecto consiste en una API que gestiona eventos (ej: eventos deportivos, elecciones, entregas de premios...) con distintas opciones para que los usuarios voten por el resultado que creen que será el ganador.

b. Objetivos del Proyecto

- Crear una API funcional que gestione Usuarios (Admins y usuarios consumidores), Eventos con diferentes Mercados y guardando las opciones seleccionadas por los Usuarios.
- Implementar operaciones CRUD completas sobre las entidades principales.
- Asegurar una interfaz intuitiva que facilite la navegación y el uso por parte de los usuarios, además de una interfaz atractiva que consiga que los usuarios se sientan cómodos.
- Cumplir con los estándares de usabilidad, mantenibilidad y seguridad.
- Incorporar un sistema de autenticación y control de acceso con JWT, para que ciertas acciones que necesiten autenticación de usuario.
- Muestra de datos mediante gráficos y otros componentes visuales.

c. Justificación del Proyecto

Este proyecto nace de mi necesidad, junto con la de mi grupo de amigos, de realizar “apuestas” dentro de eventos deportivos de forma organizada, divertida y sin implicar dinero real. La idea surgió concretamente durante la **Velada del Año de Ibai**, un evento de boxeo amateur, y que en nuestra experiencia despertó el deseo de predecir ganadores, resultados y momentos clave como si de una quiniela entre amigos se tratase.

A pesar de que existen plataformas de apuestas oficiales, ninguna se ajusta a un entorno privado, seguro y personalizado para grupos cerrados. Además, muchas de estas opciones implican dinero, lo cual no era el propósito del grupo. Por ello, surge esta aplicación: como una herramienta para **crear eventos personalizados**, permitir que los usuarios participen con predicciones o “apuestas internas” y llevar un control de manera automatizada.

La aplicación inicialmente surge para cubrir esta necesidad concreta, pero posteriormente se me ocurrió escalarla para otros tipos de eventos o competiciones entre amigos.

II. Análisis de Requerimientos

a. Identificación de Necesidades y Requerimientos

- Registro de usuarios con distintos roles (admin, usuario).
- CRUD de Eventos, Mercados....
- Gestión de las Opciones Seleccionadas.
- Visualización de estadísticas del Evento por Mercados, estadísticas de acierto por Usuario.
- Validación de datos y control de errores.
- Persistencia de datos en una base de datos relacional (Oracle).

b. Identificación del Público objetivo

El público objetivo de esta aplicación son principalmente **jóvenes y adultos** que disfrutan de compartir eventos deportivos, culturales o de entretenimiento con sus grupos de amigos, y que desean hacer predicciones o “apuestas amistosas” dentro de estos contextos. En especial, está orientada a **grupos cerrados** que buscan una experiencia más privada, sin la necesidad de plataformas de apuestas con dinero real y teniendo que tener la mayoría de edad.

También puede ser útil para **organizadores de eventos locales**, como torneos deportivos en pequeños municipios. En todos estos casos, la aplicación actúa como un canal de participación interactiva, fomentando la competitividad sana, el seguimiento del evento y la implicación de los asistentes.

c. Estudio de Mercado y Competencia

Actualmente, el mercado de las plataformas de apuestas y predicciones deportivas está dominado por grandes empresas que operan con dinero real, como **Bet365, Codere o Sportium**, cuyo acceso está regulado por leyes de juego y requiere validación de identidad, depósitos económicos, además de un ambiente que no está enfocado para un entorno familiar y/o amigable.

Existen también algunas aplicaciones enfocadas al entretenimiento, como **Superbru**, **ESPN Fantasy**, que permiten realizar predicciones deportivas, pero suelen estar ligadas a eventos concretos (como ligas de fútbol o torneos oficiales) y no ofrecen la posibilidad de **crear eventos personalizados ni controlar grupos cerrados de usuarios**.

Por otro lado, algunas personas recurren a métodos improvisados como hojas de cálculo compartidas, grupos de WhatsApp y/o grupos de Discord para registrar sus apuestas internas, lo cual **carece de automatización, control y escalabilidad**.

Frente a este panorama, esta aplicación se posiciona como una **alternativa ligera, privada y personalizable**. Permite crear eventos propios, añadir subeventos o "apuestas internas", llevar un seguimiento automático de aciertos.

En definitiva, no existe en el mercado actual una solución que combine la **simplicidad, privacidad y personalización** que ofrece este proyecto, lo que representa una clara **oportunidad de cubrir un nicho específico** dentro del ocio digital entre amigos.

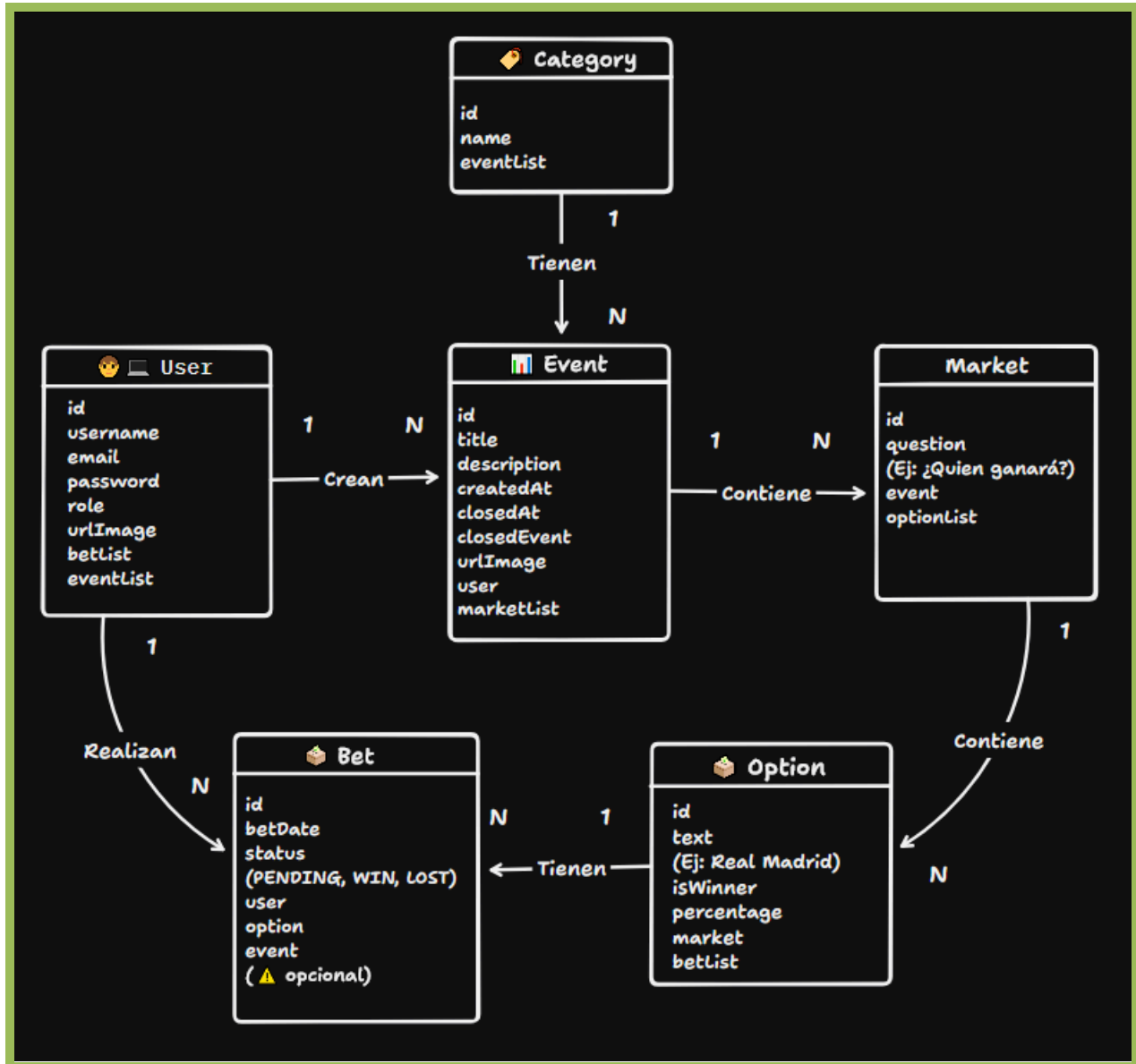
III. Diseño y Planificación

a. Definición de la Arquitectura del Proyecto

- Creación de API REST en Spring Boot, uso de Lombok y otras herramientas.
- Uso de JWT para creación de Tokens.
- Creación de Cliente Web en Spring Boot con Thymeleaf y Bootstrap.
- Base de datos en Oracle.

b. Diseño de Datos: Modelo Entidad-Relación y Esquema Relacional

En primer lugar podemos observar, el Modelo Entidad-Relación (MER), donde se representan gráficamente las entidades, atributos, relaciones y cardinalidades. A partir de este MER, se llevó a cabo el diseño de las entidades y su posterior creación a través de Jakarta y Hibernate, especificando las claves primarias, claves foráneas...



c. Diseño de la Interfaz de Usuario

La interfaz de usuario de Votium App está prevista bajo un enfoque profesional, limpio y funcional, pensado para facilitar la participación en votaciones y la gestión de eventos de forma intuitiva. A nivel visual, predominan los colores verde y dorado, que proyectan seriedad, confianza, fortuna y un toque distintivo.

🏠 Página de Inicio: impacto visual y propósito claro

Al acceder por primera vez a la aplicación, el usuario se encuentra con una landing page centrada en el mensaje del proyecto: qué es Votium y por qué usarlo. El diseño es

amplio, con mucho espacio en blanco y bloques bien definidos. En la parte superior se muestra el logo de Votium.

Justo debajo, una sección explica el funcionamiento básico de la aplicación: crear eventos, votar, ver resultados. Se acompaña con ilustraciones o mockups de ejemplo y botones de llamada a la acción como “Explorar eventos” o “Crear tu primer evento”, en dorado brillante para captar la atención.

Menú de Navegación: simple y accesible

El sistema de navegación está siempre visible en la parte superior (modo escritorio) o como un botón desplegable (modo móvil). Los enlaces están organizados horizontalmente con una tipografía clara y moderna, y el color dorado resalta la opción activa.

Este menú guía al usuario por toda la app, permitiendo acceder fácilmente a los diferentes apartados de la App.

Pantallas de Registro e Inicio de Sesión

La experiencia de autenticación se mantiene mínima y eficiente: fondo blanco con un cuadro central que contiene el formulario. Los campos están bien espaciados, con etiquetas flotantes y un botón dorado con efecto hover. Si hay errores de validación, se muestran mensajes claros en rojo bajo cada campo.

Creación de eventos: estructurada y asistida

Cuando un usuario crea un evento, entra en una vista tipo formulario dividido en secciones, guiándote paso a paso. Cada sección tiene un pequeño título y un ícono explicativo: por ejemplo, un icono de calendario para seleccionar la fecha de cierre o entradas dinámicas para las diferentes respuestas...

Pantalla de votación: enfocada, limpia y dinámica

Cuando un usuario accede a un evento abierto, se le presenta una tarjeta con el título, la descripción y las opciones de respuesta. Estas opciones se muestran como botones grandes o tarjetas clicables, alineadas en columna o grid, dependiendo del número.

El diseño evita cualquier tipo de distracción.

d. Planificación de las Tareas y los Recursos necesarios

- Semana 1: Elección de idea y desarrollo de las posibles funcionalidades

- Semana 2: Diseño de las entidades , así como sus atributos
- Semana 3: Creación del proyecto en Spring Boot y estructuración de paquetes
- Semana 4: Realización de la API (detalles por pulir)
- Semana 5-6-7: Realización de la Web (solución de problemas)
- Semana 8: Corrección de errores y demás aspectos por refinar
- Semana 9: Revisión final, preparación de documentos, archivos y demás material a entregar

IV. Implementación y Pruebas

a. Desarrollo de las Funcionalidades del Proyecto

Se han desarrollado las siguiente funcionalidades para el funcionamiento de **Votium**:

- Registro y autenticación de usuarios.
- Gestión de eventos, mercados y opciones.
- Registro de votos de los usuarios
- Categorizado de eventos.
- Área “Sobre mí” con CV y url a LinkedIn.

b. Pruebas Unitarias y de Integración

Durante el desarrollo de **Votium App**, se han realizado pruebas para garantizar el correcto funcionamiento de la API REST. Para ello, se ha utilizado Postman, donde he creado una colección personalizada que incluye diferentes categorías organizadas por entidad y/o controlador, desde la cuál mandamos peticiones y atacamos la API.

Cada subcategoría está orientada a probar de forma individual todos los endpoints de los distintos controladores del sistema, incluyendo:

- Controlador de Usuario con Autenticación JWT
- Controlador de Evento
- Controlador de Mercado
- Controlador de Opción de Votación

- Controlador de Voto
- Controlador de Categoría

Además, si el tiempo lo permite, me gustaría aprender e integrar pruebas unitarias usando JUnit con Spring Boot, con el objetivo de automatizar la validación del comportamiento interno de los servicios, repositorios y lógica de negocio. Esto permitiría llevar a cabo una verificación más exhaustiva del sistema, detectando errores antes del despliegue y asegurando una mayor calidad del código.

En resumen, el sistema ha sido probado manualmente mediante Postman en esta primera versión, y está planteada la evolución hacia un enfoque más automatizado mediante pruebas unitarias e incluso pruebas de integración completas.

c. Corrección de Errores y Optimización del Rendimiento

Durante el desarrollo de la API de **Votium App**, surgieron ciertos inconvenientes relacionados con la creación automática de las tablas en la base de datos mediante Spring JPA (Hibernate). En concreto, algunos nombres de entidades entraban en conflicto con palabras reservadas por el propio motor de JPA o la base de datos subyacente, lo que impedía la correcta generación del esquema.

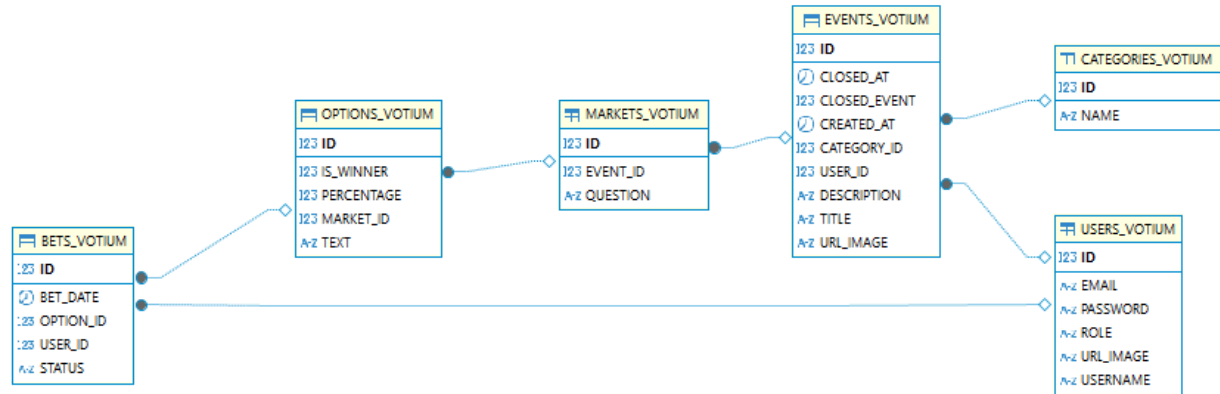
Este problema fue detectado rápidamente en la fase de arranque del proyecto, al revisar los mensajes del log. La solución consistió en renombrar ciertas tablas de forma explícita usando anotaciones como `@Table(name = "...")`.

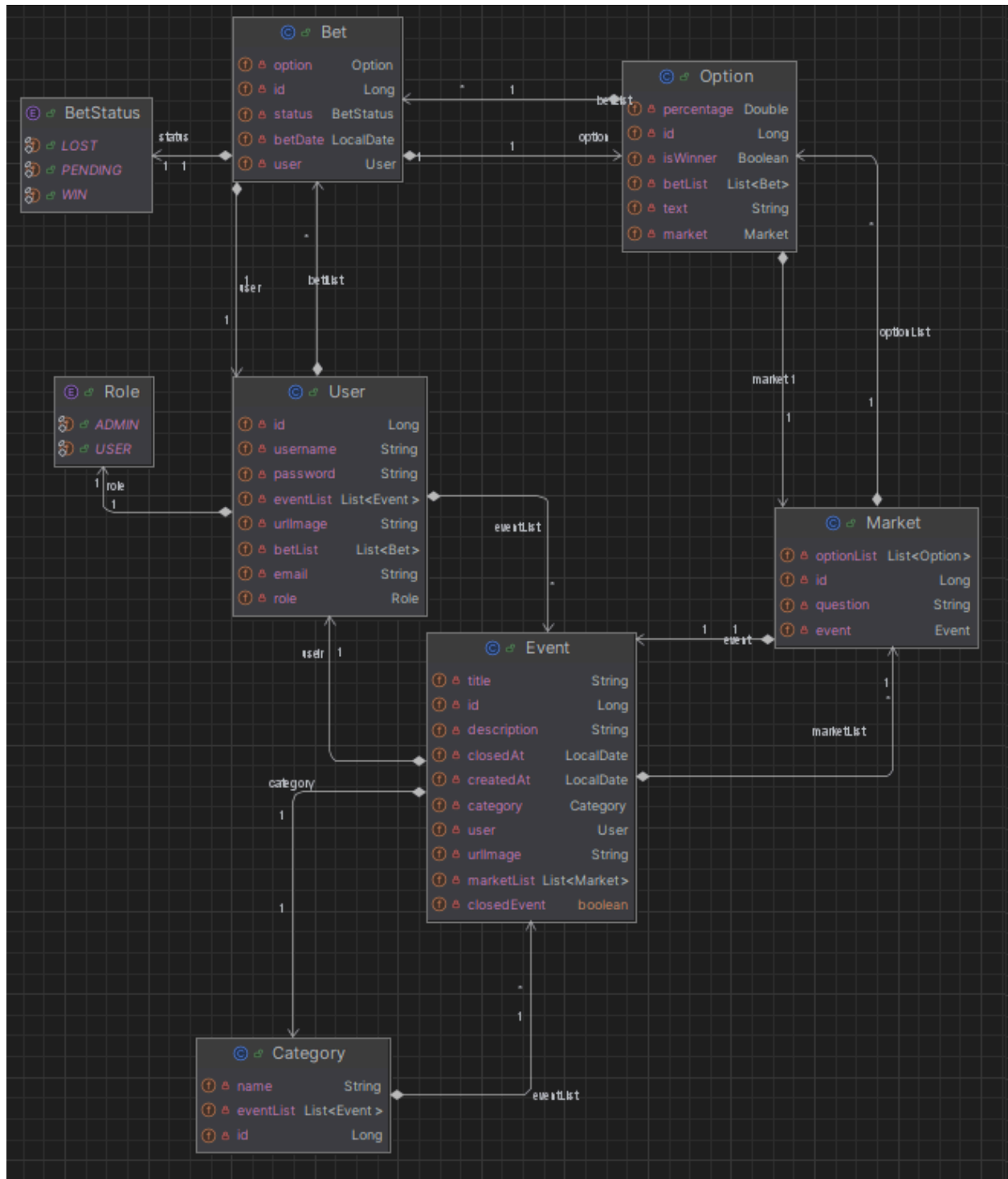
En cuanto al rendimiento, se han seguido buenas prácticas básicas, como:

- Evitar consultas innecesarias a la base de datos.
- Utilizar relaciones **LAZY** cuando es apropiado para evitar la carga excesiva de datos.
- Estructurar correctamente los controladores y servicios para mantener el código limpio, escalable y fácilmente optimizable en el futuro.

V. Documentación

a. Documentación Técnica






b. Documentación de Usuario Final

He creado un enlace a un archivo PDF que contiene la **Documentación de Usuario Final**. En este manual se explica de forma clara y detallada cómo utilizar la web, incluyendo imágenes explicativas que acompañan cada paso del proceso. El objetivo


es facilitar la comprensión y el uso correcto de todas las funcionalidades disponibles, tanto para usuarios nuevos como para quienes deseen resolver dudas específicas sobre su funcionamiento.

 Manual de Uso - Votium App (Versión Beta).pdf

c. Manual de Instalación y Configuración

He creado este manual con el objetivo de facilitar la instalación y configuración tanto de la API como de la aplicación web del proyecto. El contenido está disponible en dos formatos para mayor comodidad:

[Guia_Ejecucion_VotiumApp.md](#)

 Guia_Ejecucion_VotiumApp.pdf

Ambos documentos incluyen todos los pasos necesarios, desde la preparación del entorno hasta la ejecución final de los servicios. Se recomienda seguir las instrucciones en el orden indicado para evitar errores y asegurar un correcto funcionamiento.

VI. Mantenimiento y Evolución

a. Identificación de Posibles Mejoras y Evolución del Proyecto

Mediante iba realizando mi proyecto de Votium App he identificado varias mejoras y nuevas implementaciones con el objetivo de mejorar la seguridad, usabilidad y rendimiento, además de seguir aumentando el valor al usuario y preparar la app para seguir escalando.

Estas mejoras que he identificado son:

1. Recuperación de Contraseña mediante Correo Electrónico

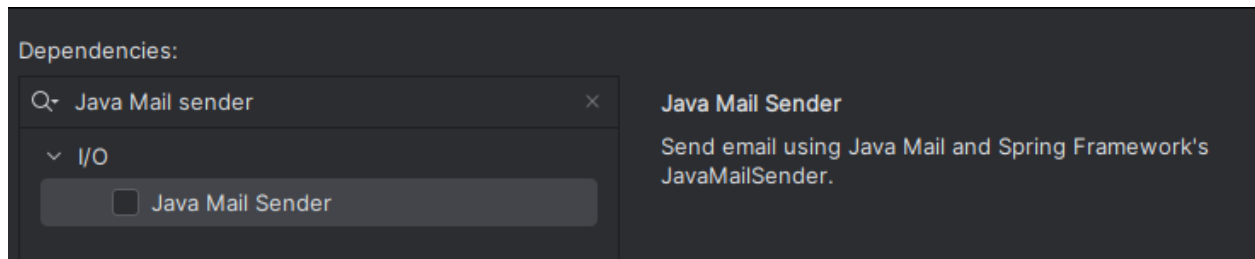
Permitir que un usuario pueda recuperar su contraseña. Esto se haría a través de un correo electrónico que contiene un enlace para restablecerla, aumentando así la accesibilidad y la autonomía del usuario en la gestión de su cuenta y la seguridad.

Posible implementación:

- Uso de tokens con JWT temporales para validar la solicitud de cambio de contraseña.
- Envío automático de correos con enlaces personalizados.

Dependencias necesarias:

- `spring-boot-starter-mail` para el envío de correos electrónicos.



2. Two-Factor Authentication - 2FA

Implementar una segunda capa de verificación al iniciar sesión. El usuario, tras introducir su contraseña, deberá verificar su identidad mediante un código generado en una app externa, como Google Authenticator por ejemplo, o mediante correo electrónico. Esto proporciona mayor seguridad a los usuarios.

Posible implementación:

- Generación de códigos temporales (habitualmente de 6 dígitos) válidos durante unos segundos.
- Integración con aplicaciones de autenticación como Google Authenticator.
- Validación del código introducido por el usuario antes de permitir el acceso completo a la aplicación.

Dependencias necesarias:

- Librerías de generación de TOTP (Time-based One-Time Password), como **Google Authenticator Java Library** o **Passlib**.
- Opcional: uso de códigos por correo como método de 2FA básico, en combinación con `spring-boot-starter-mail`.



3. Paginación de Entidades

Implementar paginación para evitar mostrar todos los elementos de una entidad (como eventos o categorías) de una sola vez. Esto mejora tanto el rendimiento del sistema como la experiencia del usuario, especialmente en dispositivos móviles o listas extensas.

Posible implementación:

- Dividir los datos en páginas de tamaño configurable (por ejemplo, 10 elementos por página).
- Añadir controles de navegación (anterior, siguiente, ir a página) en el frontend.
- Permitir ordenación y filtrado para facilitar la búsqueda dentro de las entidades paginadas.

Dependencias necesarias:

- `spring-boot-starter-data-jpa` (Ya añadida en el proyecto)

Paginación



b. Actualizaciones y Mejoras Futuras

Creo que he tenido una buena experiencia creando este proyecto y le tengo bastante cariño. Por eso, me gustaría, una vez entregado el proyecto con sus funcionalidades iniciales, seguir mejorándolo y añadiendo nuevas actualizaciones, tal como comenté en el punto anterior.

Por ello me gustaría mejorar si tuviera más tiempo:

1. Estadísticas Dinámicas de Resultados

Crear paneles visuales con gráficos interactivos para mostrar los resultados de las encuestas o votaciones.

2. Accesibilidad y Adaptación a Dispositivos Móviles

Mejorar ciertas interfaces para dispositivos móviles y así facilitar el acceso desde smartphones o tablets.

3. Permitir el inicio de sesión con otros métodos

Permitir iniciar sesión usando cuentas de Google, GitHub, etc., utilizando protocolos estándar como OAuth2.

VII. Conclusiones

a. Evaluación del Proyecto

Estoy bastante contento con el resultado de mi APP Votium. Ha sido un proyecto desafiante y enriquecedor en muchos sentidos, donde he podido aplicar mis conocimientos en Spring Boot, así como aprender nuevas técnicas de desarrollo profundizando más en el mundo Backend de Java. Durante el proceso, he enfrentado obstáculos interesantes que me han complicado el avance pero me han ayudado a crecer como desarrollador y a reforzar ciertos puntos.

El proyecto ha cumplido con los objetivos que me había propuesto inicialmente, y el resultado final ha superado mis expectativas. Votium, a mi parecer, es una aplicación funcional e intuitiva, que permite gestionar encuestas y eventos de forma eficiente.

Estoy especialmente satisfecho con la organización, creo que he sabido organizarme para crear todo lo que tenía en mente en el tiempo reducido que teníamos, y esto realmente me pone muy feliz. Creo que aún hay cosas a mejorar y poder añadir nuevas funcionalidades que hagan al usuario tener una mejor experiencia junto a una aplicación más segura.

En general, considero que la experiencia ha sido muy positiva y me siento orgulloso de lo que he logrado. Además, este proyecto me ha motivado aún más a seguir aprendiendo y perfeccionando mis habilidades como desarrollador de software.

b. Cumplimiento de Objetivos y Requisitos

He conseguido cumplir todos los objetivos iniciales que me había propuesto para este proyecto. He podido desarrollar por completo el CRUD de las entidades involucradas, como los eventos, mercados, opciones de votación y las propias votaciones, asegurando que la aplicación sea robusta y completa en cuanto a funcionalidad.

Además, logré implementar con éxito toda la lógica de votación, garantizando que cada usuario pueda emitir un único voto por mercado y/o opción, lo que era uno de los requisitos clave y más complejos. También he añadido estadísticas detalladas y gráficos (charts) que ofrecen una visión clara y atractiva de los resultados de las votaciones y de los eventos en general.

Por último, también implemente un panel de administrador para gestionar aquellas entidades que son restringidas a los usuarios normales, como pueden ser las categorías de eventos o los usuarios.

Estoy muy satisfecho con estos logros, ya que considero que aportan un gran valor a la aplicación y permiten una experiencia mucho más completa e intuitiva para los usuarios.

c. Lecciones aprendidas y Recomendaciones para Futuros Proyectos

Este proyecto me ha enseñado muchas cosas importantes para mi futuro como desarrollador. La planificación y la organización han resultado ser claves para mantener el proyecto enfocado y avanzar de forma constante. Además, he aprendido a tener en cuenta la importancia de la validación de datos y las restricciones, ya que fueron fundamentales para asegurar que la lógica de votación y el sistema de votos únicos funcionaran correctamente.

También he reforzado mi capacidad para integrar funcionalidades como los gráficos y estadísticas, lo cual me ha permitido ofrecer un resultado más visual y completo.

Para futuros proyectos, recomendaría tener en cuenta lo siguiente:

- Dedicar un tiempo inicial a planificar bien la arquitectura y las funcionalidades básicas para evitar posibles reestructuraciones más adelante, como me ocurrió con la fecha de cierre, que tuve que añadir posteriormente para poder guardar la hora también.

- Empezar a utilizar sistemas de control de versiones como Git, ya que permiten llevar un seguimiento claro de los cambios realizados, facilitan la colaboración y hacen posible volver atrás en caso de errores o experimentos fallidos. Sé que puede parecer complicado al principio, pero es una herramienta muy valiosa para gestionar ramas y mantener el proyecto siempre en un estado funcional.
- Dedicar tiempo a escribir pruebas unitarias para las funcionalidades clave del proyecto, ya que ayudan a identificar rápidamente errores o fallos en la lógica y aseguran que los cambios futuros no afecten negativamente a las partes ya implementadas. Esto refuerza la estabilidad y minimiza el riesgo de cometer errores que ya estaban previstos.
- Finalmente, no tener miedo de explorar nuevas tecnologías o librerías para enriquecer el proyecto, como fue el caso con los gráficos y la lógica compleja de votación.

En resumen, ha sido una experiencia muy importante y emocionante, que sin duda me motiva a seguir mejorando como desarrollador y a encarar nuevos desafíos con más confianza.

VIII. Bibliografía/Webgrafía y/o Referencias

a. Fuentes utilizadas en el Proyecto

- www.chatgpt.com - Uso de Inteligencia Artificial para realización de logos
- www.freepik.es - Creación de favicon con editor de FreePik