

BASE DE DATOS 3

ESCUELA TECNOLÓGICA INSTITUTO TÉCNICO CENTRAL

SISTEMA DE ANÁLISIS DEPORTIVO CONSUMIDO

PRESENTADO POR:

Matthew Espinosa Santiago
Carvajal

PRESENTADO A:

Brayan Sebastián Yepes

BOGOTÁ DC

MAYO 2025

JUSTIFICACIÓN DEL PROYECTO

En el mundo del análisis deportivo actual, especialmente en el fútbol, tomar decisiones basadas en datos ya no es una opción, sino una necesidad. Entrenadores, analistas y equipos buscan cada vez más apoyo en herramientas tecnológicas que les permitan interpretar el rendimiento en tiempo real. Sin embargo, acceder a datos confiables, estructurados y actualizados puede ser complicado, costoso o poco accesible para muchos.

Este proyecto surge precisamente para cubrir esa necesidad. La idea es desarrollar una solución que automatice la recolección, almacenamiento y análisis de datos futbolísticos a partir de una API pública y gratuita: football-data.org. A través de esta fuente, el sistema podrá extraer información sobre partidos, equipos y estadísticas clave, para luego procesarla y presentarla de forma clara mediante visualizaciones y análisis comparativos. Esto no solo ahorra tiempo al evitar el ingreso manual de datos, sino que además garantiza que la información provenga de una fuente confiable y actualizada.

La solución será desarrollada con herramientas accesibles pero robustas: Python como lenguaje principal, Flask para la construcción de la API backend, y una base de datos relacional (SQLite o PostgreSQL) para almacenar la información. Todo esto se desplegará en una instancia EC2 de AWS en el SO de Linux, lo que asegura un entorno escalable, económico y autónomo.

Además, este proyecto tiene potencial para crecer en el futuro. La arquitectura está pensada para poder adaptarse fácilmente a otros deportes, ligas o tipos de análisis, e incluso podría integrarse con sistemas de scouting, modelos de predicción o plataformas de rendimiento en tiempo real.

2. Descripción del Proyecto

El proyecto consiste en el desarrollo de un sistema que automatiza la recolección, almacenamiento y exposición de datos estadísticos del fútbol, utilizando como fuente la API pública football-data.org. El núcleo del sistema será una API REST creada con **Flask**, que actuará como intermediario entre la fuente de datos y los usuarios finales.

Los datos obtenidos —como resultados de partidos, información de equipos y estadísticas clave (goles, posesión, xG, entre otros)— serán almacenados en una base de datos relacional, utilizando **PostgreSQL** como motor principal. Una vez estructurados, estos datos estarán disponibles tanto para consulta a través de la API como para visualización en dashboards interactivos.

El sistema ofrecerá las siguientes funcionalidades principales:

- Recolección de datos históricos y actuales sobre partidos, equipos y estadísticas relevantes.
- Almacenamiento y normalización de los datos recolectados para facilitar su análisis posterior.
- Exposición de la información a través de una API propia, permitiendo su consumo por otras aplicaciones o servicios.
- Visualización de datos mediante dashboards dinámicos que faciliten la interpretación de tendencias y comparativas.
- Soporte para análisis externos, permitiendo que otras herramientas se conecten al sistema para realizar consultas o procesamientos adicionales.
- En conjunto, esta solución busca simplificar el acceso a datos deportivos estructurados, habilitando su uso en proyectos de análisis, visualización y toma de decisiones basadas en información real y actualizada.

3. Objetivos

Objetivo General: Desarrollar una API REST funcional y documentada, que consuma, almacene y exponga datos deportivos reales, utilizando estándares de diseño y despliegue profesional.

Objetivos Específicos:

- Integrar y consumir datos desde una API (football-data.org).
- Crear una base de datos PostgreSQL para almacenamiento.
- Desarrollar en Flask para exposición de los datos.
- Desplegarla en una instancia EC2 de AWS.
- Implementar paneles de visualización para el análisis de datos deportivos.
- Establecer mecanismos de seguridad y autenticación básicos.

4. Alcance Incluye:

- Conexión con una API.
- Base de datos.
- Backend con Flask.
- Despliegue inicial en AWS EC2.
- Panel básico de visualización y documentación técnica o dashboard.

No incluye:

- Aplicaciones móviles.
- Funcionalidades de login avanzado o autenticación múltiple.
- Inteligencia artificial o predicción automática de resultados.

5. Cronograma Tentativo (por sprints)

Sprint	Semana	Entregables principales
1	1-2	Acta, WBS, ERD, roles, BPMN, configuración inicial
2	3-4	Conexión con API y pruebas en local
3	5-6	Modelado base de datos y endpoints
4	7-8	Visualización, integración y seguridad
5	9-10	Pruebas finales, presentación y despliegue completo

6. Requisitos del Proyecto • API pública football-data.org.

- PostgreSQL 13 o superior.
- Python 3.10, Flask, SQLAlchemy, requests.
- AWS EC2 Ubuntu Server 22.04.
- Herramientas de apoyo: Trello, GitHub, DBeaver, ngrok, Insomnia, Jira.

7. Roles y Responsabilidades

Nombre	Rol	Responsabilidad principal
Santiago Carvajal	Líder del Proyecto	Dirección, gestión, pruebas, desarrollo backend y despliegue
Matthew Espinosa	DBA y Arquitecto	Diseño BD, Desarrollo Básico, pruebas y soporte técnico

8. Supuestos

- El API football-data.org mantendrá su disponibilidad durante el proyecto ya que es algo público.
- El equipo contará con acceso estable a Internet y servicios de AWS ya que es nuestro entorno y pues todavía no nos cobran a ninguno.

9. Restricciones

- No saber de lo que nos piden.
- No saber usar la IA.
- No hacer nada, pues ya que el internet lo tiene todo.

10. Aprobación del Proyecto

Nombre	Rol	Firma	Fecha
Santiago Carvajal	Líder del Proyecto	San\$tiag'Carvaj,IF	26/05/2025
Matthew Espinosa	DBA / Arquitecto	Ma1hewEspin'sa	26/05/2025

WBS DEL PROYECTO DE ANÁLISIS DEPORTIVO CON API

1. Inicio del Proyecto

- 1.1 Definición del problema
- 1.2 Justificación
- 1.3 Objetivos generales y específicos
- 1.4 Acta de constitución del proyecto

2. Planeación

- 2.1 Asignación de roles (Santiago - Backend, Matthew - Visualización)
- 2.2 Identificación de herramientas (API, Python, Flask, BD, EC2)

3. Análisis y Diseño

- 3.1 Consulta a la API
- 3.2 Definición de endpoints y estructura de datos

4. Desarrollo e Implementación

- 4.1 Conexión con API externa usando Python
- 4.2 Creación y conexión con base de datos (SQLite o PostgreSQL)
- 4.3 Desarrollo del backend con Flask
- 4.4 Procesamiento de datos y limpieza

5. Visualización y Consumo de Datos

- 5.1 Desarrollo de interfaz visual o dashboard (local o web)
- 5.2 Visualización de métricas: goles, posesión, xG, etc.
- 5.3 Exportación o visualización de resultados

6. Pruebas y Ajustes

- 6.1 Pruebas de conexión API y rendimiento
- 6.2 Validación de integridad de los datos en BD
- 6.3 Validación de visualización y lógica de negocio

7. Documentación y Entrega

- 7.1 Capturas de evidencia
- 7.2 Informe final del proyecto
- 7.3 Presentación del producto terminado

DEFINICIÓN DE USUARIOS DEL SISTEMA

1. Administrador del Sistema

Descripción:

Usuario con conocimientos técnicos, generalmente parte del equipo de desarrollo o mantenimiento del sistema.

Responsabilidades:

- Configurar la conexión con la API externa.
- Administrar la base de datos (crear, eliminar, actualizar).
- Controlar el acceso al sistema.
- Ejecutar mantenimientos y actualizaciones del backend.

Permisos:

- Acceso completo al sistema
- Control de API keys y tokens
- Modificación de scripts y endpoints
- Visualización y edición total de los datos

2. Analista Deportivo

Descripción:

Profesional que interpreta los datos recolectados por el sistema para generar informes, análisis y recomendaciones.

Responsabilidades:

- Consultar estadísticas históricas y actuales.
- Interpretar métricas como goles, posesión, tarjetas, rendimiento ofensivo/defensivo, etc.
- Exportar gráficos o tablas para presentaciones.
- Detectar patrones o tendencias en el desempeño de equipos.

Permisos:

- Acceso completo a visualizaciones y consultas
- Filtros por fechas, equipos, ligas

3. Tester o Evaluador

Descripción:

Usuario encargado de validar el correcto funcionamiento del sistema en cuanto a rendimiento, integridad de los datos y experiencia de usuario.

Responsabilidades:

- Validar que los datos del API se integren correctamente.
- Verificar que las visualizaciones estén bien renderizadas.
- Reportar errores o bugs encontrados en el sistema.

Permisos:

- Acceso a todos los módulos funcionales

4. Usuario Final (Observador)

Descripción:

Persona que simplemente accede a la plataforma para visualizar resultados de manera rápida y sencilla, sin conocimientos técnicos.

Responsabilidades:

- Navegar por el sistema de visualización.
- Observar estadísticas clave de equipos o partidos.
- Obtener información para fines personales, académicos o recreativos.

Permisos:

- Solo lectura de datos y visualizaciones

5. Docente o Evaluador Académico

Descripción:

Profesor o tutor que revisa el avance del proyecto para evaluación académica.

Responsabilidades:

- Consultar documentación del proyecto.
- Probar funcionalidades clave del sistema.
- Verificar coherencia entre lo planeado y lo ejecutado.

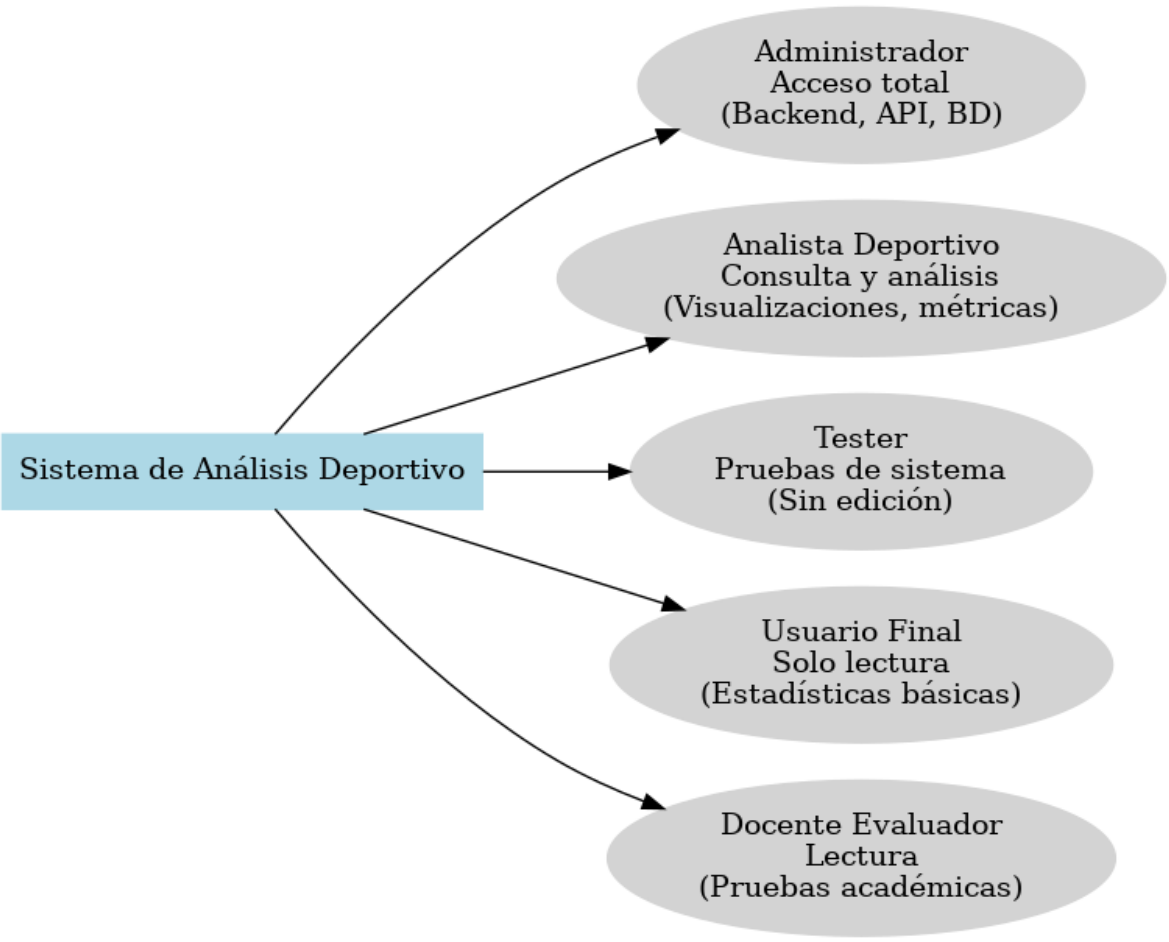
Permisos:

- Acceso a documentación, endpoints principales y dashboard

TABLA DE USUARIOS:

Tipo de Usuario	Descripción	Permisos Clave
Administrador del Sistema	Encargado de configurar, mantener y actualizar el sistema	Acceso completo, puede modificar datos y configuración.
Analista Deportivo	Analiza los datos y genera conclusiones estadísticas.	Acceso a visualizaciones, filtros y exportaciones.
Tester o Evaluador	Valida el correcto funcionamiento del sistema.	Acceso de prueba, sin modificar datos.
Usuario Final (Observador)	Accede a los resultados de forma pasiva.	Solo lectura básica de resultados.
Docente o Evaluador Académico	Evalúa el proyecto con fines académicos.	Lectura limitada, acceso a documentación y dashboard.

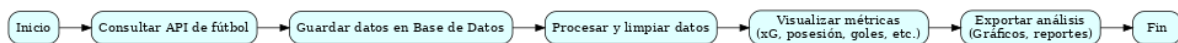
DIAGRAMA:



MODELO ENTIDAD RELACION:



DIAGRAMA BPMN:



ERD normalizado:

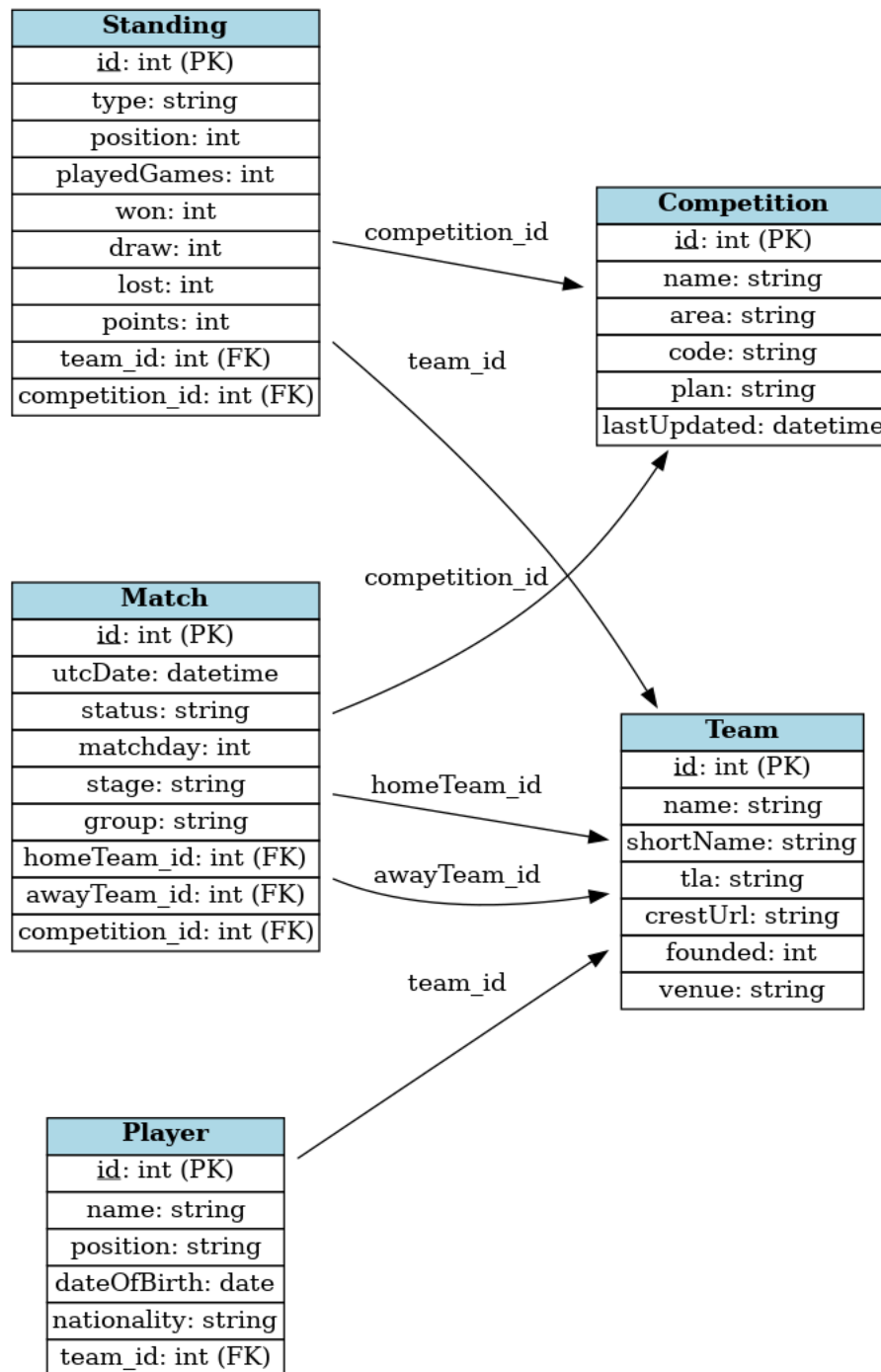


DIAGRAMA DE CLASES:

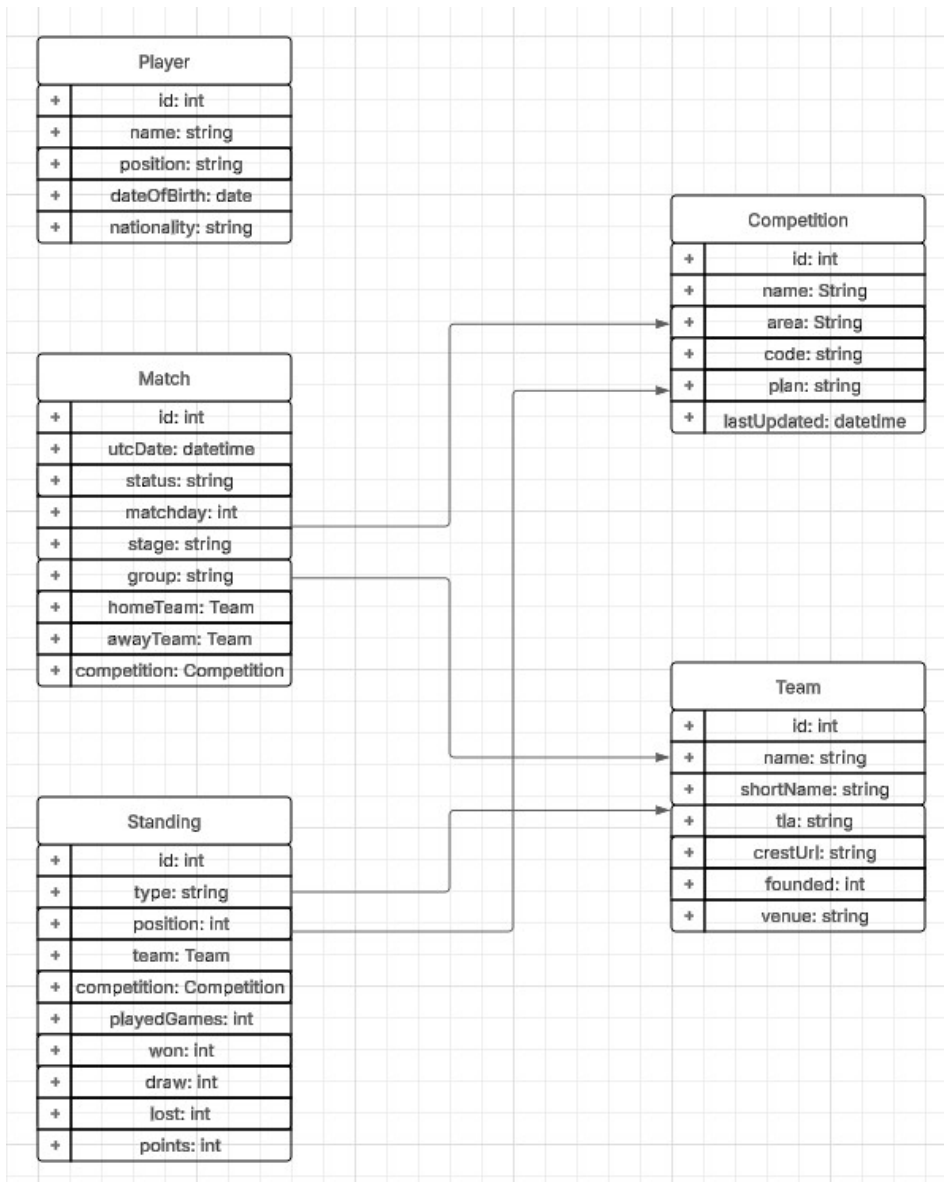
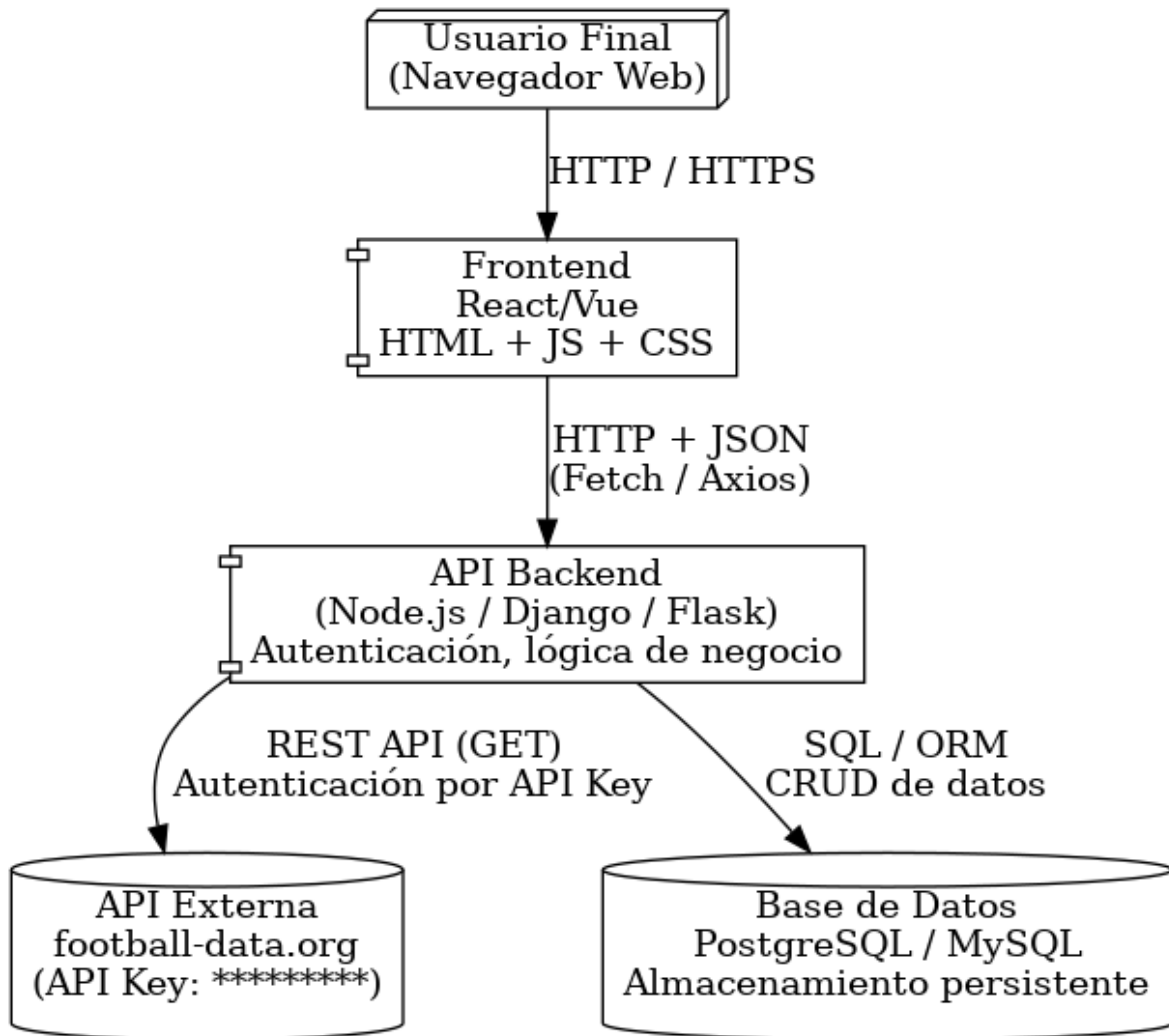
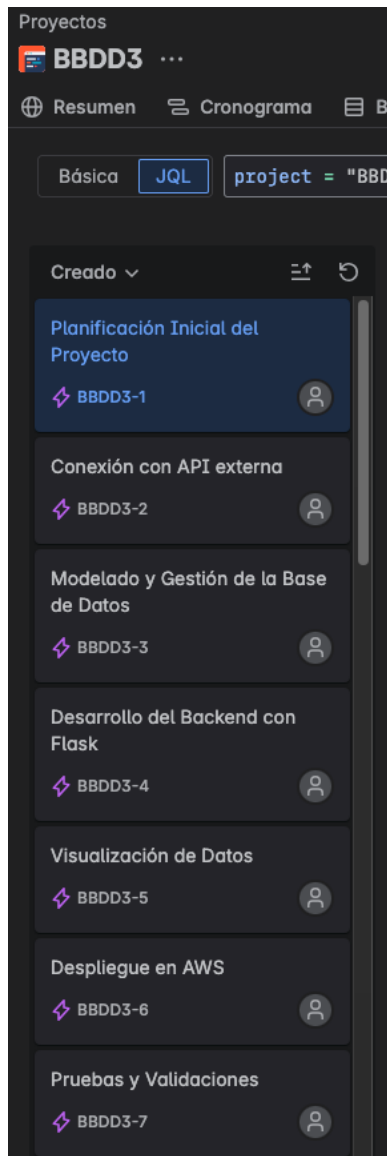


DIAGRAMA DE DESPLIEGUE:



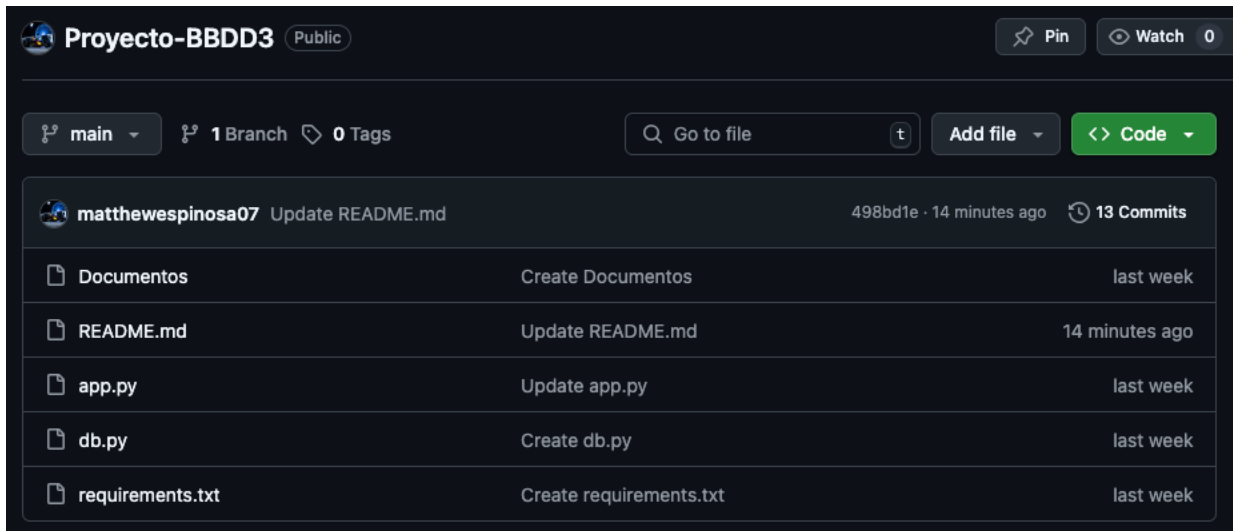
Documentación inicial en Jira

Link: <https://prproyectobdts8a.atlassian.net/jira/software/projects/BBDD3/issues/BBDD3-1?jql=project%20%3D%20%22BBDD3%22%20ORDER%20BY%20created%20ASC>



Commits en GitHub

Link: <https://github.com/matthewespinosa07/Proyecto-BBDD3/tree/main>



The screenshot shows the GitHub interface for the repository 'Proyecto-BBDD3'. At the top, the repository name is displayed with a 'Public' badge. To the right are buttons for 'Pin' and 'Watch' (0). Below this, the current branch is 'main', with '1 Branch' and '0 Tags' indicated. A search bar 'Go to file' and buttons 'Add file' and 'Code' are present. The commit history section shows a list of commits by user 'matthewespinosa07'. The most recent commit is 'Update README.md' with hash '498bd1e' from '14 minutes ago', which has '13 Commits' linked to it. Below this, a table lists the files changed in this commit: 'Documentos' (Create), 'README.md' (Update), 'app.py' (Update), 'db.py' (Create), and 'requirements.txt' (Create). Each file entry includes the action taken and the time since the change.

File	Action	Time
Documentos	Create	last week
README.md	Update	14 minutes ago
app.py	Update	last week
db.py	Create	last week
requirements.txt	Create	last week