

# **DOCUMENTACION DE PROYECTO BOOTCAMPS**

Autores

**Leonardo David Díaz León**  
**Juan Sebastián Tirado Verbel**

Docente

**Boris Espitia**

Asesor

**Emilio**

Curso

**Programación**

**Montería, Colombia**  
**2024**

## Contenido

|   |    |
|---|----|
| • Introducción .....  | 3  |
| Propósito .....   | 3  |
| Arquitectura del Proyecto .....                                 | 4  |
| 1. Tecnologías Utilizadas .....                                 | 4  |
| 2. Diagrama de Arquitectura .....                               | 4  |
| Instalación y Configuración .....                               | 7  |
| ▪ Requisitos Previos .....                                      | 7  |
| ▪ Configuración del Entorno.....                                | 8  |
| ▪ Configuración de la Base de Datos .....                       | 9  |
| Base de Datos.....  | 10 |
| Base de Datos: Estructura y Diseño .....                        | 10 |
| Diseño del Esquema de Base de Datos .....                       | 10 |
| Relaciones entre Tablas .....                                   | 11 |
| Ejemplo de Consulta SQL .....                                   | 11 |
| Funcionalidades Principales .....                               | 15 |
| Funcionalidades del Sistema .....                               | 15 |
| Creación de Torneos.....  | 15 |
| Sistema de Estadísticas .....                                   | 16 |
| Interfaz de Usuario .....                                       | 16 |
| Interfaz Web del Sistema KICK OFF .....                         | 16 |
| Diseño Responsivo .....   | 17 |
| Problemas Conocidos y Limitaciones .....                        | 23 |
| Limitaciones Identificadas.....                                 | 23 |
| Rendimiento .....   | 23 |
| Seguridad .....   | 24 |
| Usabilidad .....  | 24 |
| Disponibilidad.....   | 24 |
| Futuras Mejoras.....  | 25 |
| Lista de funcionalidades planeadas para versiones futuras ..... | 25 |
| Posibles mejoras adicionales incluyen.....                      | 26 |
| Resumen de funcionalidades adicionales.....                     | 26 |
| Créditos y Licencia .....                                       | 27 |

- **Introducción**

### **Propósito**

El proyecto **KICK OFF** es una aplicación web innovadora y eficiente, diseñada específicamente para facilitar la gestión de torneos deportivos. Esta plataforma está pensada para ser una herramienta integral dirigida a administradores y organizadores de eventos deportivos, brindándoles una solución práctica y robusta para registrar equipos, programar partidos y llevar un control detallado de los resultados y clasificaciones. El objetivo fundamental de KICK OFF es simplificar y optimizar el proceso de organización de torneos, ofreciendo una interfaz intuitiva y fácil de usar, con acceso flexible y personalizable para adaptarse a las distintas necesidades de los usuarios.

**Nombre del Proyecto:** Organizador de Torneos – **KICK OFF**

**Descripción:** **KICK OFF** es una plataforma web orientada a la organización y gestión eficiente de torneos deportivos, especialmente enfocados en el fútbol. Permite a los administradores y organizadores crear y gestionar torneos de manera sencilla, desde el registro de equipos y la programación de partidos, hasta la recopilación y visualización de resultados y clasificaciones. La plataforma está diseñada para optimizar la coordinación de eventos deportivos y mejorar la experiencia de los participantes y espectadores..

**Objetivo:** El objetivo de KICK OFF es desarrollar una plataforma web que permita a los usuarios organizar, gestionar y participar en torneos de fútbol de manera flexible y eficiente. Se busca ofrecer una solución que contemple tanto opciones de acceso gratuito, con funcionalidades básicas, como planes premium que brinden características avanzadas y un mayor control sobre la gestión del torneo. La meta es crear una experiencia personalizada y accesible para todos los tipos de usuarios, desde organizadores amateurs hasta profesionales de la gestión deportiva.

## Arquitectura del Proyecto

El proyecto KICK OFF está construido bajo una arquitectura cliente-servidor, donde la interacción de los usuarios se realiza a través de una interfaz web, y la lógica de negocio y la gestión de datos se procesan en el servidor. Este enfoque garantiza una separación clara entre el frontend y el backend, optimizando el flujo de información y la escalabilidad de la aplicación.

### 1. Tecnologías Utilizadas

#### Backend:

- **Python:** Se utiliza como lenguaje de programación principal para el desarrollo del servidor, proporcionando una base sólida y versátil para manejar la lógica de negocio de la aplicación.
- **Flask:** Un microframework de Python que permite construir aplicaciones web de manera sencilla y eficiente. Flask se encarga de recibir las solicitudes del cliente, procesar la lógica de negocio y enviar respuestas adecuadas.

#### Base de Datos:

- **SQLite:** Una base de datos ligera y fácil de configurar que se utiliza para almacenar y gestionar de manera eficiente la información de los torneos, equipos, jugadores y resultados. SQLite se integra perfectamente con Flask, facilitando las operaciones de lectura y escritura.

#### Frontend:

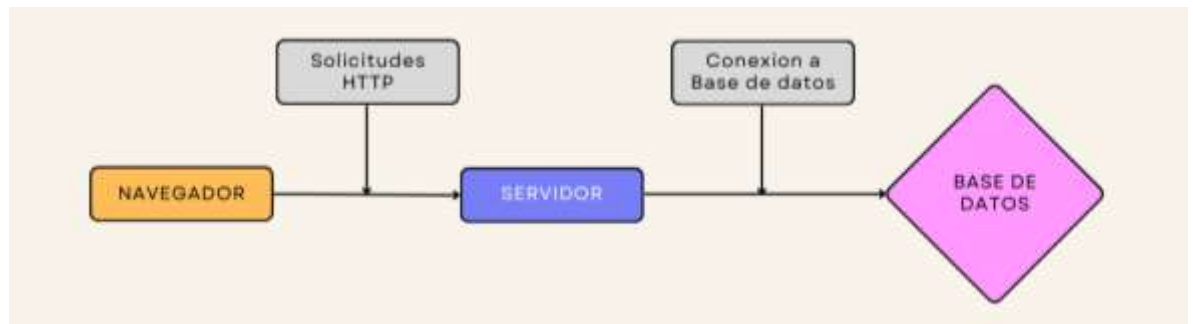
- **HTML, CSS, JavaScript:** Se utilizan para crear la interfaz de usuario de la aplicación, proporcionando una experiencia interactiva y amigable. HTML se emplea para estructurar la página web, CSS para estilizar y JavaScript para agregar funcionalidades interactivas y dinámicas.

#### Framework de Estilo:

- **Bootstrap:** Un framework de diseño front-end que ayuda a desarrollar una interfaz web responsiva y estéticamente agradable de manera rápida y eficiente. Bootstrap se utiliza para facilitar la creación de un diseño moderno y adaptable a diferentes dispositivos.

### 2. Diagrama de Arquitectura

El siguiente diagrama ilustra cómo fluye la información entre los componentes de la arquitectura cliente-servidor:



- **Cliente (Navegador):** El usuario interactúa con la interfaz web para enviar solicitudes, como el registro de equipos, la visualización de resultados o la programación de partidos.
- **Servidor (Flask):** Recibe las solicitudes del cliente, procesa la lógica de negocio necesaria y se encarga de las operaciones de lectura y escritura en la base de datos.
- **Base de Datos (SQLite):** Almacena y gestiona la información de los torneos, equipos y resultados, permitiendo al servidor acceder y actualizar los datos de manera eficiente.

Esta arquitectura modular y bien definida permite que KICK OFF proporcione una plataforma robusta y escalable para la gestión de torneos deportivos, asegurando una experiencia de usuario fluida y un rendimiento optimizado.

- **Estructura del Proyecto:** Presenta la organización de los archivos y carpetas.

app

- \_\_pycache\_\_
  - app.cpython-312.pyc
  - config.cpython-312.pyc
- models
  - \_\_pycache\_\_
  - database
  - db.py

- routes
  - \_\_pycache\_\_
  - \_\_init\_\_.py
  - main.py
- static
  - css
    - style.css
  - img
    - escudos
    - avatar.svg
    - img-torneo.svg
  - js
    - login.js
    - logout.js
    - partidos.js
    - registroEquipos.js
    - registroJugadores.js
    - registroTorneo.js
    - script.js
    - singUp.js
- templates
  - dashboard.html
  - dashboardLanding.html
  - equipos.html
  - fixtures.html
  - index.html
  - layout.html
  - login.html

- registerTorneo.html
- registroEquipos.html
- registroJugadores.html
- singUp.html
- torneo.html
- verEquipo.html
- utils
  - \_\_pycache\_\_
    - funtions.cpython-312.pyc
    - funtions.cpython-313.pyc
  - funtions.py
- \_\_init\_\_.py
- app.py
- config.py
- README.md

## Instalación y Configuración

Para poner en marcha el proyecto KICK OFF y garantizar su funcionamiento adecuado, es necesario seguir algunos pasos de instalación y configuración. A continuación, se describe el proceso detallado:

### **Instalación y Configuración del Proyecto KICK OFF**

Para poner en marcha el proyecto KICK OFF y garantizar su funcionamiento adecuado, es necesario seguir algunos pasos de instalación y configuración. A continuación, se describe el proceso detallado:

### **Requisitos Previos**

Antes de comenzar con la instalación y configuración, asegúrate de tener lo siguiente en tu sistema:

- **Python:** Debes instalar Python en tu computadora. Se recomienda utilizar Python 3.7 o superior. Puedes verificar si Python está instalado ejecutando el siguiente comando en la terminal o línea de comandos:
- `python --version`

Si Python no está instalado, descárgalo e instálalo desde [python.org](https://python.org).

- **pip:** Es el gestor de paquetes de Python y debe estar incluido en la instalación de Python. Verifica su instalación con:
- `pip --version`

Si `pip` no está instalado, puedes instalarlo siguiendo las instrucciones en la [documentación oficial](#).

## Configuración del Entorno

Para mantener un entorno de desarrollo limpio y manejar dependencias de manera eficiente, se recomienda usar un entorno virtual. Sigue estos pasos:

1. **Crea un entorno virtual:** Navega al directorio de tu proyecto y ejecuta:
2. `python -m venv venv`
3. **Activa el entorno virtual:**
  - En Windows:
  - `venv\Scripts\activate`
  - En macOS y Linux:
  - `source venv/bin/activate`

**Instala las dependencias:** Crea un archivo `requirements.txt` con las siguientes líneas (o asegúrate de tenerlo con las dependencias necesarias):

```
Flask==2.0.3
Flask-SQLAlchemy==2.5.1
```

Instala las dependencias con:

```
pip install -r requirements.txt
```

También puedes instalar Flask y SQLite directamente si prefieres hacerlo manualmente:

```
pip install Flask Flask-SQLAlchemy
```



## Configuración de la Base de Datos

La base de datos SQLite se utilizará para almacenar la información de los torneos. Para configurarla:

**Crea el archivo de la base de datos:** En tu proyecto, crea un archivo de base de datos SQLite llamado `databaseTorneo.db` (o cualquier otro nombre que prefieras).

**Conecta Flask con SQLite:** En tu archivo de configuración de Flask (`app.py` o `config.py`), añade la siguiente línea para configurar la conexión a la base de datos:

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///databaseTorneo.db'
```

**Inicializa la base de datos:** Puedes crear las tablas necesarias ejecutando un script de inicialización de la base de datos en Python:

```
from app import db
# Crear todas las tablas definidas en el modelo
db.create_all()
```

Asegúrate de que tu aplicación tenga el siguiente código de inicialización:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///databaseTorneo.db'
db = SQLAlchemy(app)

if __name__ == '__main__':
    with app.app_context():
        db.create_all() # Crea las tablas en la base de datos
```

**Verifica la conexión y estructura:** Inicia tu aplicación y verifica que la base de datos se haya creado correctamente y que las tablas estén en su lugar. Puedes usar una herramienta de administración de SQLite como **DB Browser for SQLite** para inspeccionar la base de datos y asegurarte de que la estructura se haya implementado correctamente.

# Base de Datos

## Base de Datos: Estructura y Diseño

El sistema KICK OFF utiliza **SQLite** como base de datos para gestionar de manera eficiente la información relacionada con equipos, jugadores, partidos y resultados. SQLite es ideal para este proyecto debido a su ligereza, facilidad de integración con Flask y su capacidad para manejar operaciones de lectura y escritura de forma rápida y segura.

## Diseño del Esquema de Base de Datos

El esquema de la base de datos está diseñado con tablas interrelacionadas que permiten almacenar la información de manera estructurada. Esta estructura facilita tanto la consulta de datos como su actualización, asegurando que la plataforma funcione de manera eficiente y sin redundancias. A continuación, se describen las principales tablas y sus relaciones:

### 1. Tabla de Equipos

- **Campos:** id\_equipo, nombre\_equipo, ciudad, estado, fecha\_creacion
- **Relaciones:** Relacionada con la tabla de jugadores a través de id\_equipo, y con la tabla de partidos para registrar los equipos participantes.

### 2. Tabla de Jugadores

- **Campos:** id\_jugador, nombre\_jugador, apellido\_jugador, edad, posicion, id\_equipo
- **Relaciones:** Relacionada con la tabla de equipos a través de id\_equipo para asignar jugadores a sus respectivos equipos.

### 3. Tabla de Partidos

- **Campos:** id\_partido, id\_equipo\_local, id\_equipo\_visitante, fecha\_partido, ubicacion
- **Relaciones:** Relacionada con las tablas de equipos para registrar los equipos participantes y con la tabla de resultados para almacenar los resultados de cada partido.

### 4. Tabla de Resultados

- **Campos:** id\_resultado, id\_partido, goles\_equipo\_local, goles\_equipo\_visitante, comentarios
- **Relaciones:** Relacionada con la tabla de partidos para asociar los resultados con su partido correspondiente.

### 5. Tabla de Torneos

- **Campos:** id\_torneo, nombre\_torneo, fecha\_inicio, fecha\_fin, tipo\_torneo

- **Relaciones:** Relacionada con las tablas de equipos y partidos para gestionar los equipos participantes y los partidos programados durante el torneo.

### Relaciones entre Tablas

Las tablas están interconectadas mediante claves foráneas que garantizan la integridad referencial y la coherencia de los datos. Por ejemplo:

- La **tabla de jugadores** tiene una clave foránea `id_equipo` que se relaciona con la **tabla de equipos**, asignando jugadores a un equipo específico.
- La **tabla de partidos** utiliza las claves foráneas `id_equipo_local` e `id_equipo_visitante` para vincular los equipos que participan en un partido.
- La **tabla de resultados** se asocia con la **tabla de partidos** mediante la clave foránea `id_partido`, asegurando que cada resultado corresponda a un partido específico.

### Ventajas del Diseño de Esquema

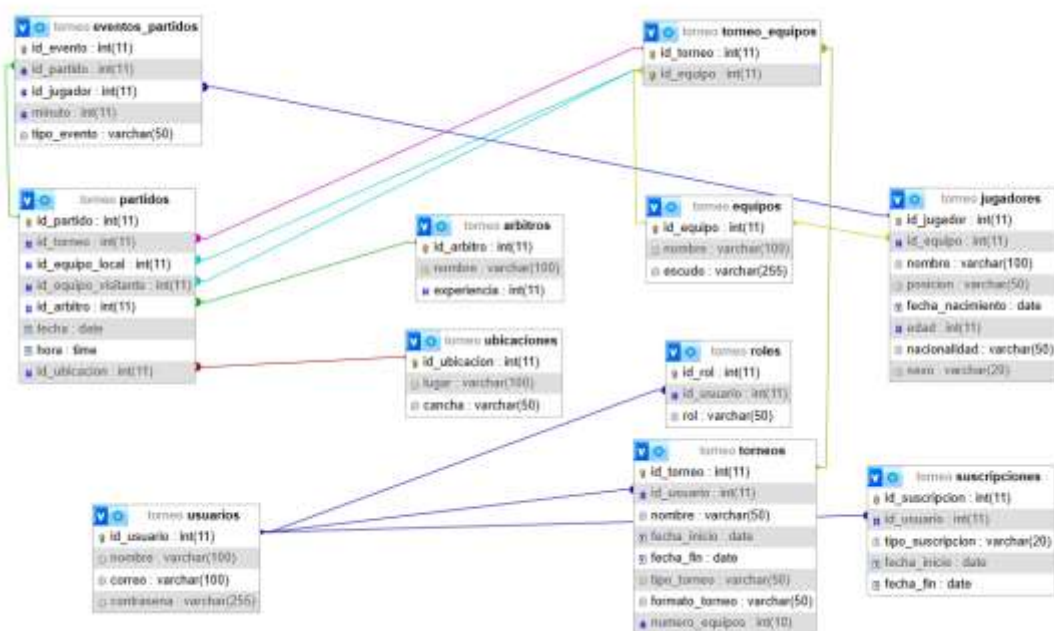
- **Estructura Normalizada:** La base de datos está diseñada siguiendo principios de normalización para evitar redundancia de datos y garantizar la integridad de la información.
- **Consultas Eficientes:** La interrelación de las tablas permite realizar consultas complejas de manera eficiente, como la obtención de la lista de jugadores por equipo, los resultados de un torneo específico o los detalles de los partidos programados.
- **Flexibilidad para Actualizaciones:** Las actualizaciones de los datos se pueden realizar de manera sencilla y consistente gracias a la estructura modular de las tablas y sus relaciones.

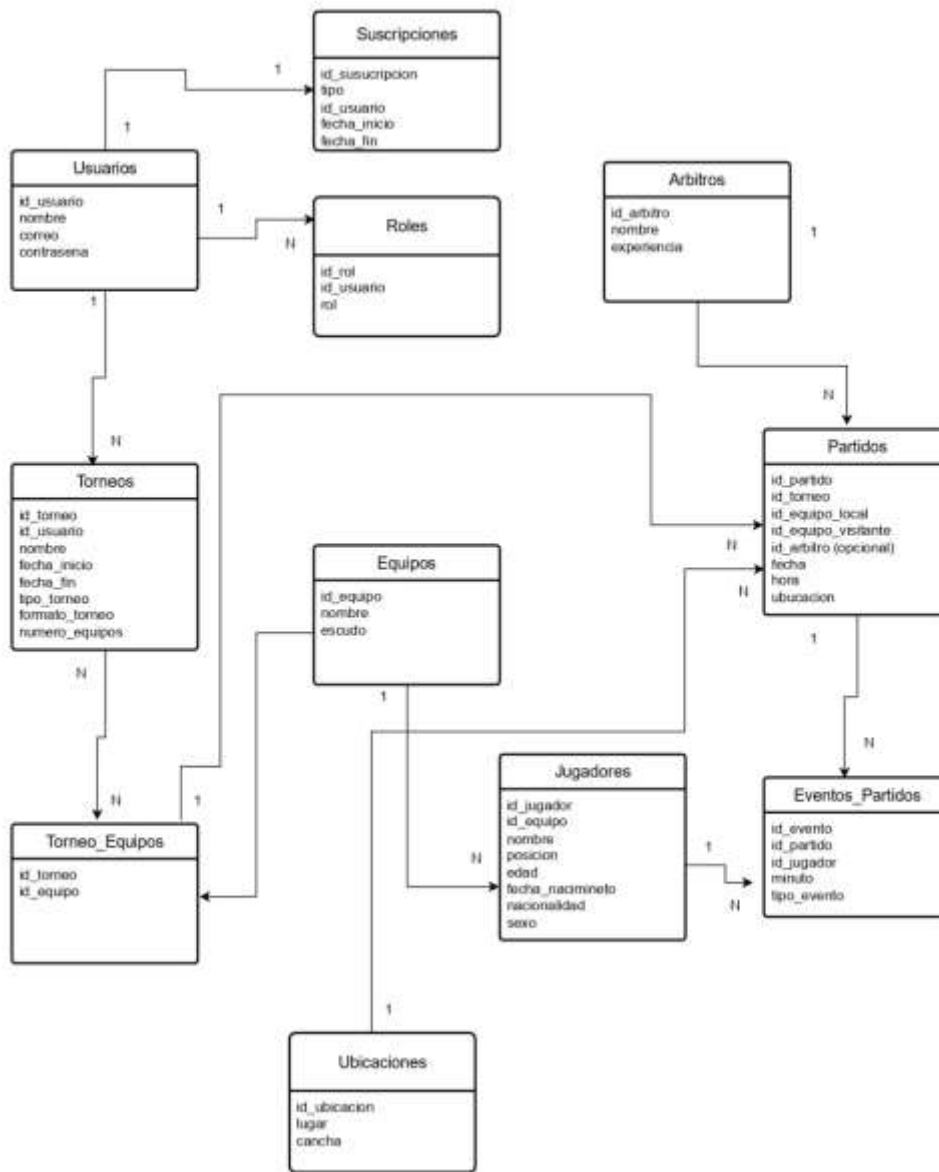
### Ejemplo de Consulta SQL

Aquí hay un ejemplo de una consulta SQL que podría realizarse en este esquema para obtener los resultados de un torneo específico:

```
SELECT t.nombre_torneo, p.fecha_partido, e1.nombre_equipo AS equipo_local,
e2.nombre_equipo AS equipo_visitante,
r.goles_equipo_local, r.goles_equipo_visitante
FROM torneos t
JOIN partidos p ON t.id_torneo = p.id_torneo
JOIN equipos e1 ON p.id_equipo_local = e1.id_equipo
JOIN equipos e2 ON p.id_equipo_visitante = e2.id_equipo
JOIN resultados r ON p.id_partido = r.id_partido
WHERE t.id_torneo = ?;
```

12







## Funcionalidades Principales

### Funcionalidades del Sistema

El sistema está diseñado para ofrecer una solución integral para la organización y gestión de torneos deportivos, permitiendo a los administradores y participantes interactuar de manera eficiente y segura. A continuación, se detalla un desglose de las funcionalidades principales del sistema:

#### Gestión de Usuarios

El módulo de gestión de usuarios permite un control completo sobre la creación y administración de cuentas, así como la asignación de permisos y el acceso a distintas funcionalidades de la plataforma.

- **Registro e inicio de sesión:** Los usuarios pueden registrarse en la plataforma mediante un proceso de registro sencillo, con la opción de iniciar sesión de forma segura. Esta funcionalidad asegura que cada usuario tenga acceso exclusivo a sus datos y configuraciones.
- **Asignación de roles (Superadministrador, Administrador de Torneo, Espectador):** Se pueden asignar distintos roles con niveles de acceso y permisos específicos. Los superadministradores tienen control total sobre la plataforma, mientras que los administradores de torneo gestionan los eventos y los espectadores acceden solo a la información de los torneos.
- **Gestión de planes (Gratis y Premium):** Los usuarios tienen la opción de elegir entre planes de acceso gratuito o premium, cada uno ofreciendo diferentes características y beneficios. Los planes premium pueden incluir funcionalidades avanzadas como estadísticas detalladas, personalización de torneos y más.

#### Creación de Torneos

Este módulo permite a los organizadores crear y gestionar torneos con flexibilidad y control, adaptándose a distintas necesidades y tipos de eventos deportivos.

- **Personalización de torneos:** Los organizadores pueden configurar los detalles específicos de sus torneos, como nombre, fecha, ubicación, y otras características esenciales que definen el evento.
- **Selección de modalidad de juego:** Se puede seleccionar la modalidad de juego más adecuada para cada torneo, como partidos de eliminación directa, ligas, torneos de grupos, etc.

- **Configuración de formato de torneo:** La plataforma permite establecer el formato de competición, especificando el número de rondas, la estructura de los partidos y otros aspectos clave.
- **Gestión de equipos y jugadores:** Los organizadores pueden agregar y gestionar equipos y jugadores, asignando miembros a los equipos y actualizando la información de los participantes conforme avanzan las fases del torneo.

## Sistema de Estadísticas

El sistema de estadísticas ofrece a los administradores y a los participantes la capacidad de analizar y visualizar el rendimiento y los resultados de los torneos.

- **Generación de estadísticas:** La plataforma genera estadísticas detalladas, que pueden incluir rendimiento de equipos, puntuaciones, análisis de partidos, y más. Esto permite a los organizadores evaluar y hacer un seguimiento del progreso del torneo.
- **Visualización restringida:** La visualización de estadísticas y datos específicos está limitada de acuerdo con los permisos de acceso de los usuarios, protegiendo la confidencialidad y la integridad de la información.
- **Control de acceso por roles:** Los accesos a las funciones de análisis y a la visualización de estadísticas están controlados según el rol del usuario, asegurando que solo aquellos con los permisos adecuados puedan acceder a la información sensible.

## Interfaz de Usuario

### Interfaz Web del Sistema KICK OFF

El sistema KICK OFF ha sido diseñado con una interfaz web intuitiva y fácil de usar, que facilita la interacción de los usuarios con la plataforma. La interfaz está organizada de manera que los administradores y organizadores de torneos puedan acceder rápidamente a las funciones principales y gestionar los eventos deportivos con eficiencia. A continuación, se describen las secciones clave de la interfaz web:

### Páginas Principales de la Interfaz

1. **Sección de Gestión de Torneos**
  - **Funcionalidades:** Permite a los usuarios ver la lista de torneos activos y pasados, así como crear, editar y eliminar torneos.



- **Características:** Cada torneo puede incluir detalles como nombre, fechas de inicio y fin, modalidad de juego y estado del torneo.
- **Usabilidad:** Botones y formularios fáciles de usar para agregar nuevos torneos y actualizar información existente.

## 2. Sección de Registro de Equipos y Jugadores

- **Funcionalidades:** Facilita el registro de nuevos equipos y jugadores, asignando jugadores a los equipos correspondientes y gestionando sus datos.
- **Características:** Formularios de entrada de datos con campos para nombre del equipo, nombre del jugador, posición, edad y otros detalles relevantes.
- **Usabilidad:** Interfaces limpias y formularios que permiten la adición de múltiples jugadores y equipos en una sola operación.

## 3. Sección de Visualización de Estadísticas y Resultados

- **Funcionalidades:** Presenta estadísticas de los partidos, clasificaciones y resultados de torneos de manera clara y comprensible.
- **Características:** Gráficos, tablas y resúmenes de los partidos, con opciones de filtrado para ver estadísticas de equipos, jugadores y torneos específicos.
- **Usabilidad:** Sección interactiva con opciones para ordenar y filtrar los datos por diferentes criterios, como fecha, equipo, o tipo de partido.

## Diseño Responsivo

El sistema está diseñado con un enfoque de **diseño responsivo**, lo que garantiza que la plataforma sea completamente funcional y se vea bien en diferentes dispositivos y tamaños de pantalla. Esto permite a los usuarios acceder a la aplicación desde computadoras de escritorio, tabletas y teléfonos móviles sin problemas de visualización ni de usabilidad.

### Características del Diseño Responsivo:

- **Adaptabilidad:** Los elementos de la interfaz, como menús, botones y tablas, se ajustan automáticamente al tamaño de la pantalla del dispositivo.
- **Compatibilidad:** Utilización de tecnologías como **HTML5**, **CSS3** y **Bootstrap** para garantizar que el contenido se adapte y mantenga una presentación adecuada en todos los dispositivos.
- **Experiencia de Usuario (UX):** Elementos interactivos que se rediseñan de manera eficiente para mejorar la navegación y la interacción en pantallas pequeñas y grandes.

Inicio



## Login



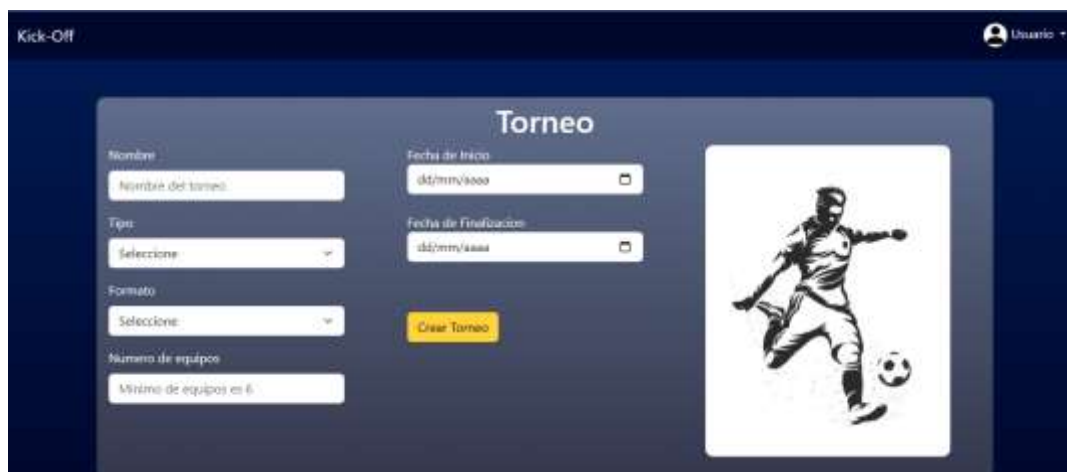
## Registro



## dashboard



register-torneo







| LOGO    | HOME     | FIXTURES    | RESULTS | TABLE     | PLAYERS |
|---------|----------|-------------|---------|-----------|---------|
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |
| 5/09/24 | 07:00 pm | Real Madrid | 2 VS 1  | Liverpool |         |

LOGO

HOME

FIXTURES

RESULTS

TABLE

PLAYERS

| Position | Club   | Played | Won | Drawn | Lost | GF | GA | GD | Points | Form   |
|----------|--|--------|-----|-------|------|----|----|----|--------|--|
| 1        |  Liverpool                  | 11     | 9   | 1     | 1    | 21 | 6  | 15 | 28     | <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> |
| 2        |  Manchester City            | 11     | 7   | 2     | 2    | 22 | 13 | 9  | 23     | <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> |
| 3        |  Chelsea                   | 11     | 5   | 4     | 2    | 21 | 13 | 8  | 19     | <span>L</span> <span>L</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> |
| 4        |  Arsenal                  | 11     | 5   | 4     | 2    | 18 | 12 | 6  | 19     | <span>W</span> <span>W</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> |
| 5        |  Nottingham Forest        | 11     | 5   | 4     | 2    | 15 | 10 | 5  | 19     | <span>L</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>W</span> <span>L</span> |
| 6        |  Brighton And Hove Albion | 11     | 5   | 4     | 2    | 19 | 15 | 4  | 19     | <span>W</span> <span>W</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> <span>L</span> |

## Problemas Conocidos y Limitaciones

El proyecto presenta varias áreas clave a mejorar y optimizar para asegurar su funcionalidad, escalabilidad y seguridad en el uso. A continuación, se analizan los aspectos mencionados de manera más amplia y detallada:

### Limitaciones Identificadas

#### 1. Soporte Multilingüe

Actualmente, la plataforma no es compatible con múltiples idiomas, lo que restringe su accesibilidad a usuarios que no hablan el idioma predeterminado. La inclusión de un sistema de localización permitiría traducir tanto la interfaz como el contenido dinámico, ampliando el alcance global de la aplicación.

#### 2. Interfaz en Dispositivos Móviles

Aunque funcional, la interfaz actual requiere optimización para pantallas pequeñas. Esto incluye un diseño responsivo mejorado y pruebas exhaustivas en diferentes resoluciones de dispositivos móviles para garantizar una experiencia de usuario óptima.

#### 3. Configuración Manual de Equipos

El ajuste manual del número de equipos por torneo es ineficiente. Un sistema dinámico que permita al administrador definir los equipos con mayor flexibilidad reduciría significativamente el tiempo necesario para configurar un torneo.

### Rendimiento

Un buen desempeño es esencial para una aplicación exitosa. Los objetivos clave incluyen:

- **Tiempo de Carga de Página (< 3 segundos):**

Se debe implementar optimización de scripts, uso de almacenamiento en caché, compresión de archivos y bases de datos bien estructuradas para mantener el tiempo de carga por debajo del umbral de 3 segundos.

- **Soporte para Usuarios Concurrentes (500 usuarios):**

La infraestructura debe soportar un tráfico simultáneo significativo. Esto puede lograrse mediante balanceadores de carga, servidores escalables y optimización del backend para manejo de múltiples solicitudes.

- **Escalabilidad:**  
Es crucial diseñar la arquitectura pensando en el futuro, permitiendo la incorporación de nuevos módulos y servicios sin afectar el rendimiento actual.

## Seguridad

La protección de los datos y la confianza del usuario son esenciales:

- **Encriptación de Datos:**  
Todo dato sensible, como contraseñas y datos personales, debe estar encriptado utilizando estándares modernos como AES-256.
- **Autenticación de Doble Factor (2FA):**  
Este método añade una capa adicional de seguridad, asegurando que solo los usuarios legítimos puedan acceder a sus cuentas.
- **Protección Contra Amenazas Comunes (OWASP Top 10):**  
Se debe implementar validación de entradas, protección contra inyecciones SQL y sanitización de datos para prevenir ataques comunes.
- **Cumplimiento de Normativas:**  
La aplicación debe alinearse con regulaciones como el GDPR (en Europa) o leyes locales de protección de datos según el área de operación.

## Usabilidad

La experiencia de usuario es un factor crítico para la aceptación y retención:

- **Interfaz Intuitiva:**  
Diseñar una experiencia de usuario clara, con navegación sencilla y elementos visuales intuitivos.
- **Compatibilidad con Navegadores Modernos:**  
La aplicación debe ser probada y funcional en navegadores como Chrome, Firefox, Edge y Safari.
- **Accesibilidad (WCAG 2.1 Nivel AA):**  
Incluir características de accesibilidad como soporte para lectores de pantalla, alto contraste y navegación por teclado para personas con discapacidades.

## Disponibilidad

Para garantizar una experiencia confiable y consistente, los objetivos de disponibilidad incluyen:



- **Uptime (99.5%):**  
El sistema debe estar disponible la mayor parte del tiempo, utilizando infraestructura redundante para minimizar interrupciones.
- **Recuperación Ante Fallos (< 30 minutos):**  
Implementar procedimientos y herramientas para una rápida recuperación en caso de fallas, como backups automáticos y monitoreo en tiempo real.

## Futuras Mejoras

El propósito de estas futuras mejoras es enriquecer la experiencia del usuario, optimizar la funcionalidad del sistema y expandir las capacidades de la plataforma para atender de manera más efectiva las necesidades de los usuarios y organizadores de torneos deportivos. A través de estas actualizaciones, se busca hacer la plataforma más accesible, funcional y adaptable, además de facilitar su uso en distintos contextos y regiones.

### Lista de funcionalidades planeadas para versiones futuras

1. **Soporte para múltiples idiomas:**
  - Implementar un sistema de localización y traducción para que la plataforma pueda ser utilizada en diferentes idiomas, permitiendo un acceso más amplio a una audiencia global y ofreciendo una mejor experiencia a los usuarios de diversas regiones.
2. **Integración con plataformas de pagos para inscripción:**
  - Integrar métodos de pago seguros y eficientes para la inscripción de participantes en los torneos. Esto incluye opciones de pago en línea a través de tarjetas de crédito, débito, y sistemas de pago digitales como PayPal y otros servicios de pago móviles.
3. **Aplicación móvil complementaria:**
  - Desarrollar una aplicación móvil que complemente la plataforma web, permitiendo a los usuarios gestionar y seguir sus torneos desde dispositivos móviles. La aplicación ofrecerá una interfaz amigable y funcional que permita a los organizadores y participantes estar al tanto de los eventos y actualizaciones en tiempo real.

Posibles mejoras adicionales incluyen

**4. Soporte para personalización de torneos por categorías o modalidades:**

- Permitir a los organizadores personalizar sus torneos de acuerdo a distintas categorías o modalidades de juego, adaptándose a diferentes tipos de deportes y formatos de competición, lo que brindará mayor flexibilidad en la organización de eventos.

**5. Implementación de notificaciones por correo electrónico para eventos importantes:**

- Integrar un sistema de notificaciones automáticas por correo electrónico que alerte a los participantes y organizadores sobre eventos clave, como cambios en el calendario, actualizaciones de resultados, recordatorios de inscripción y otros eventos importantes.

**6. Integración con una API de mapas para ubicar sedes de partidos:**

- Incorporar un sistema de localización que permita a los usuarios ver y obtener direcciones hacia las sedes de los partidos, mejorando la accesibilidad y la planificación de los desplazamientos de los participantes y espectadores.

**7. Desarrollo de API para futuras integraciones:**

- Crear una API robusta que facilite la integración de la plataforma con otros sistemas y aplicaciones, como plataformas de redes sociales para compartir resultados y actualizaciones en tiempo real, o herramientas de análisis de rendimiento de jugadores y equipos.

Resumen de funcionalidades adicionales

- **Integración de sistemas de pago:** Asegurar que la plataforma pueda aceptar distintos métodos de pago, brindando comodidad y seguridad para la inscripción en torneos.
- **Soporte multilenguaje:** Expandir la accesibilidad de la plataforma a nivel internacional con la adición de múltiples idiomas.
- **Desarrollo de API para futuras integraciones:** Crear una infraestructura flexible que permita la interacción con otras aplicaciones y plataformas, fomentando la interoperabilidad y la expansión de funcionalidades.

Estas mejoras están diseñadas para hacer de la plataforma una herramienta más completa y adaptable, brindando una experiencia enriquecida a los organizadores y participantes de torneos deportivos, optimizando tanto la gestión como la participación en los eventos.

## Créditos y Licencia

Créditos: Proyecto desarrollado por Leonardo David Díaz León y Juan Sebastián Tirado Verbel.

El proyecto fue desarrollado por Leonardo David Díaz León y Juan Sebastián Tirado.

El uso y distribución del proyecto están sujetos a la licencia especificada en el archivo de licencia del repositorio.

Github: <https://github.com/JuanT20/Torneo>

Figma: <https://www.figma.com/design/yACg2KKf6fkEyBEP3P0lLw/TORNEO?node-id=0-1&t=ySZestCipyNzOaYl-1>