



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**FutboStats
Análisis y estadísticas de
fútbol**



Presentado por Miguel Ángel Extremo
Cabornero
en Universidad de Burgos — 9 de junio de 2024
Tutor: César Represa Pérez

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	5
Apéndice B Especificación de Requisitos	15
B.1. Introducción	15
B.2. Objetivos generales	16
B.3. Catálogo de requisitos	16
B.4. Especificación de requisitos	18
Apéndice C Especificación de diseño	31
C.1. Introducción	31
C.2. Diseño de datos	31
C.3. Diseño procedimental	34
C.4. Diseño arquitectónico	38
C.5. Diseño de interfaces	41
Apéndice D Documentación técnica de programación	43
D.1. Introducción	43
D.2. Estructura de directorios	43

<i>Índice general</i>	II
D.3. Manual del programador	45
D.4. Compilación, instalación y ejecución del proyecto	47
D.5. Pruebas del sistema	55
Apéndice E Documentación de usuario	58
E.1. Introducción	58
E.2. Requisitos de usuarios	58
E.3. Instalación	58
E.4. Manual del usuario	59
Apéndice F Anexo de sostenibilización curricular	73
F.1. Introducción	73
Bibliografía	75

Índice de figuras

A.1. Compatibilidad entre licencias	10
B.1. Diagrama de casos de uso	19
C.1. Ejemplo de petición a API-FOOTBALL	32
C.2. Ejemplo de petición a la API de Flask	33
C.3. Diagrama de flujo de obtención de datos a API-FOOTBALL . .	35
C.4. Diagrama de flujo de inicio de sesión en FutboStats	36
C.5. Diagrama de flujo de obtención de datos en la API de Flask . .	37
C.6. Diagrama de flujo de uso de ImageAI	38
C.7. Arquitectura cliente-servidor [15]	40
C.8. Prototipo de Inicio	41
C.9. Prototipo general con header, body y footer	41
C.10. Prototipo de la vista de ImageAI	42
D.1. Ejecución de <i>ng version</i> para comprobar la instalación.	47
D.2. Clonar repositorio de GitHub.	48
D.3. Importar proyecto en Visual Studio Code	49
D.4. Creación de la carpeta node modules con las dependencias del proyecto Angular	50
D.5. Creación y activación del entorno virtual en Python.	51
D.6. Listar dependencias instaladas utilizando el comando <i>pip list</i> . .	51
D.7. Comprobación de la instalación de <i>Ffmpeg</i>	53
D.8. Ejecución de la aplicación Angular mediante el comando <i>ng serve -o</i> en el <i>Angular CLI</i>	54
D.9. Ejecución de la aplicación Angular desde Visual Studio Code. .	54
D.10. Ejecución de la aplicación Flask utilizando <i>python run.py</i>	55
E.1. Inicio de FutboStats	59

E.2. Tutoriales en Inicio de FutboStats	60
E.3. Búsqueda de partidos en FutboStats	61
E.4. Búsqueda de partidos en FutboStats sin resultados.	61
E.5. Búsqueda de equipos en FutboStats.	62
E.6. Búsqueda de competiciones en FutboStats.	63
E.7. Paso 1 de iniciar sesión en FutboStats: entrar en iniciar sesión. .	64
E.8. Paso 2 de iniciar sesión en FutboStats: elegir la cuenta de Google.	65
E.9. Paso 3 de iniciar sesión en FutboStats: autenticado con éxito. .	65
E.10. Cerrar sesión en FutboStats.	66
E.11. Paso 1 para utilizar SportAI: entrar en 'SportsAI'.	67
E.12. Paso 2 para utilizar SportAI: pulsar o arrastrar un vídeo '.mp4'.	67
E.13. Paso 3 para utilizar SportAI: visualizar el vídeo generado por imageAI	68
E.14. Visualización de predicción de puntos esperados y real.	69
E.15. Representación gráfica de los puntos esperados a lo largo de las jornadas.	70
E.16. Visualización tablas del modelo de goles esperados xG	70
E.17. Representación gráfica del modelo de goles esperados xG.	71
E.18. Visualización de las estadísticas del mundial 2022	71
E.19. Seleccionar el jugador para visualizar su gráfico.	72

Índice de tablas

A.1. Costes de personal	6
A.2. Costes de <i>hardware</i>	7
A.3. Costes de <i>software</i>	7
A.4. Costes varios	7
A.5. Costes totales	8
A.6. Ingresos esperados por publicidad y colaboraciones	8
A.7. Dependencias del proyecto	10
A.8. Resumen de la licencia MIT	12
A.9. Resumen de la licencia CC-BY-4.0	13
A.10. Fuentes de imágenes de terceros	13
A.11. Resumen de las licencias del proyecto	14
B.1. CU-1 Consultar datos de fútbol	20
B.2. CU-1.1 Consultar equipos	21
B.3. CU-1.1.1 Obtener datos de los jugadores del equipo	22
B.4. CU-1.2 Buscar partidos de fútbol	23
B.5. CU-1.3 Buscar clasificaciones	24
B.6. CU-2 Consultar estadísticas y predicciones de fútbol	25
B.7. CU-2.1 Consultar predicciones de puntos esperados	26
B.8. CU-2.2 Consultar modelo de goles esperados	27
B.9. CU-2.3 Consultar estadísticas del Mundial 2022	28
B.10. CU-2.4 Utilizar modelo de tracking en vídeos de fútbol	29
B.11. CU-3 Autenticación	30

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La fase de planificación constituye un punto importante en cualquier proyecto. En esta fase se estima el trabajo, el tiempo y el dinero que va a suponer realizar el proyecto. Se analizan todas las partes que va a tener el proyecto previamente para conocer al máximo posible los recursos necesarios. La fase de planificación la podemos dividir en:

- Planificación temporal.
- Estudio de viabilidad.

En la planificación temporal, se desarrolla un calendario de tiempos donde se estima el tiempo necesario para la realización de cada una de las partes del proyecto o *sprints*. Se establece una fecha fija de inicio del proyecto y una fecha de finalización esperada. Debemos tener en cuenta los requisitos que se deben cumplir para poder empezar a trabajar en cualquiera de las tareas.

El estudio de viabilidad del proyecto se puede dividir en dos apartados:

- Viabilidad económica.
- Viabilidad legal.

La viabilidad económica de un proyecto supone estimar sus costes y beneficios que puede suponer realizar el proyecto. La viabilidad legal supone analizar

todas aquellas leyes que puedan afectar al proyecto como el uso de licencias y la Ley de Protección de Datos en el caso del *software*.

A.2. Planificación temporal

Al inicio del proyecto se planteó utilizar una metodología ágil como Scrum para la gestión del proyecto. Se han aplicado las siguientes características Scrum en el desarrollo del proyecto:

- Se aplicó una estrategia de desarrollo incremental a través de iteraciones (*sprints*) y revisiones.
- La duración media de los *sprints* fue de entre dos y cuatro semanas.
- Al finalizar cada *sprint* se entregaba una parte del producto funcional(incremento).
- Se realizaban reuniones para revisar las tareas realizadas en el *sprint* y se planificaba el siguiente *sprint*.
- En la planificación del *sprint* se generaba una pila de tareas a realizar.
- Estas tareas se estimaban y priorizaban en un tablero canvas.

A continuación, se describen los diferentes *sprints* que se han realizado:

Sprint 0 (19/02/24 - 18/03/24)

La reunión de planificación de este *sprint* marcó el comienzo del proyecto. Unas semanas atrás en una reunión se había planteado una idea inicial del proyecto. En esta nueva reunión se profundizó en las funcionalidades que iba a tener la aplicación web y se establecieron algunas pautas y objetivos. Los objetivos fueron:

- Profundizar y formalizar los objetivos del proyecto.
- Investigar sobre que frameworks o lenguajes de programación podría utilizar para llevar a cabo el desarrollo del *front-end*.
- Investigar sobre que frameworks o lenguajes de programación podría utilizar para llevar a cabo el desarrollo del *back-end*.

- Investigar acerca del funcionamiento de ImageAI y crear un pequeño código de ejemplo [10].
- Investigar acerca de las métricas utilizadas en la estadística del fútbol.
- Aprender a utilizar la librería de Python de StatsbombPy [14].

Se dió un tiempo de cuatro semanas para preparar este *sprint* y presentar los avances al tutor.

Sprint 1 (19/03/24 - 09/04/24)

Una vez decididas las herramientas de desarrollo del proyecto en el anterior *sprint* (Angular y Flask) se pasó al desarrollo de la web.

Los objetivos de este *sprint* fueron los siguientes:

- Se realizó un curso online de Angular para obtener los conocimientos necesarios y que no se disponían al empezar el proyecto [6].
- Realizar prototipos de la aplicación web antes de iniciar su desarrollo en Angular.
- Iniciar el desarrollo en Angular creando una página visual y fácil de utilizar para el usuario.
- Aplicar ImageAI a un vídeo de fútbol de ejemplo.
- Implementar un login con Google en la aplicación.
- Implementar la protección de rutas en la aplicación.
- Comenzar a desarrollar la parte del *back-end* en Flask.
- Investigar sobre como realizar peticiones a una API externa (API-FOOTBALL).
- Desarrollar pantallas de la aplicación en las que el usuario puede consultar datos fácilmente.

Las tareas en las que se descompusieron los objetivos se pueden ver en: [Sprint 1](#). Este *sprint* tuvo una duración de tres semanas.

Sprint 2 (10/04/24 - 17/04/24)

Los objetivos de *sprint* fueron:

- Continuar el desarrollo *front-end* en Angular.
- Continuar con el desarrollo del *back-end* en Flask.
- Investigar los endpoints de API-FOOTBALL para llevar a cabo las funcionalidades deseadas.
- Creación de gráficos en Flask con datos de StatsBombPy.

Las tareas en las que se descompuso el *sprint* se pueden ver en: [Sprint 2](#)

Sprint 3 (18/04/24 - 30/04/24)

Los objetivos del *sprint* fueron:

- Continuar el desarrollo *front-end* en Angular.
- Corregir errores de la interfaz de usuario de la web.
- Implementar los gráficos generados en Flask en la aplicación web.
- Creación de gráficos en Flask con datos de StatsBombPy.

Las tareas en las que se descompuso el *sprint* se pueden ver en: [Sprint 3](#)

Sprint 4 (1/05/24 - 15/05/24)

Los objetivos del *sprint* fueron:

- Desplegar la aplicación *front-end* en Netlify.
- Desplegar la aplicación *back-end* en Render.
- Implementación de ImageAI en la aplicación web.
- Hacer responsive las vistas para poder utilizar la aplicación en móvil.
- Desarrollar la memoria del proyecto.

Este *sprint* tuvo una duración de dos semanas. Las tareas en las que se descompuso el *sprint* se pueden ver en: [Sprint 4](#)

Sprint 5 (16/05/24 - 1/06/24)

Los objetivos del *sprint* fueron:

- Corrección de aspectos confusos y mejoras visuales de la web.
- Hacer responsive más vistas para poder utilizar la aplicación en móvil.
- Finalizar la memoria del proyecto.
- Iniciar los anexos del proyecto.

Este *sprint* tuvo una duración de dos semanas. Las tareas en las que se descompuso el *sprint* se pueden ver en: [Sprint 5](#)

Sprint 6 (02/06/24 - 12/06/24)

Los objetivos del *sprint* fueron:

- Finalizar anexos del proyecto.
- Correcciones de memoria y anexos.
- Testing con Postman.
- Crear máquina virtual con el proyecto en local.

Este *sprint* tuvo una duración de diez días hasta la entrega del proyecto completo. Las tareas en las que se descompuso el *sprint* se pueden ver en: [Sprint 6](#)

A.3. Estudio de viabilidad

Viabilidad económica

En el siguiente apartado se analizarán los costes y beneficios que hubieran supuesto desarrollar este proyecto en un entorno empresarial real.

Costes

Los costes del proyecto se pueden desglosar en las siguientes categorías.

Costes de personal:

El proyecto ha sido llevado a cabo por un desarrollador empleado a tiempo completo durante cuatro meses. Se considera el siguiente salario:

Concepto	Coste
Salario mensual neto	1.000€
Retención IRPF (19 %)	235,30€
Seguridad Social (28,3 %)	514,90€
Salario mensual bruto	1.750,20€
Total 4 meses	7.000,8€

Tabla A.1: Costes de personal.

Para realizar los cálculos en la tabla proporcionada, se han utilizado los siguientes porcentajes para determinar las diferentes deducciones y componentes del salario [13].

La retención IRPF (Impuesto sobre la Renta de las Personas Físicas) varía según las circunstancias, pero para este proyecto se ha supuesto un 19 %. La contribución a la Seguridad Social se ha calculado con los siguientes porcentajes:

- Contingencias comunes: 23,60 %
- Desempleo de tipo general : 4,70 %
- Fondo de Garantía Salarial (FOGASA): 0,20 %
- Formación profesional: 0,60 %
- Total Seguridad Social: $23,60\% + 4,70\% + 0,20\% + 0,60\% = 28,3\%$

Costes de hardware:

En este apartado se revisan los costes de los dispositivos hardware utilizados en el desarrollo del proyecto. Se considera que la amortización para estos dispositivos es de 5 años y han sido utilizados 4 meses.

Concepto	Coste	Coste amortizado
Ordenador de sobremesa	1.000€	66,67€
Móvil	400€	26,67€
Total	1.400€	93,34€

Tabla A.2: Costes de *hardware*.**Costes de software:**

En este apartado se revisan los costes de las licencias software no gratuitas utilizadas en el desarrollo del proyecto. Se considera que la amortización del software es de 2 años.

Las herramientas utilizadas que requieren licencia son Windows 10, Adobe Photoshop (utilizado para la edición del logo), Adobe Premiere (utilizado para editar, recortar vídeos grabados de la aplicación).

Concepto	Coste	Coste amortizado
Adobe Premiere	240€	40,00€
Adobe Photoshop	120€	20,00€
Windows 10	279€	46,52€
Total	639€	106,52€

Tabla A.3: Costes de *software*.**Costes varios:**

En este apartado se citan el resto de costes del proyecto:

Concepto	Coste
Dominio de futbostats	31,90€
Precio por plan de API-FOOTBALL	25€
Internet	150€
Alquiler de la oficina	500€
Memoria impresa y pendrives	50€
Total	756,90€

Tabla A.4: Costes varios.

Costes totales:

El sumatorio de todos los costes es el siguiente:

Concepto	Coste
Personal	7.000,8€
Hardware	93,34€
Software	106,52€
Varios	756,90€
Total	7.957,56€

Tabla A.5: Costes totales.

Beneficios

La aplicación web desarrollada se distribuirá inicialmente de forma gratuita, permitiendo a los usuarios acceder a estadísticas de fútbol y realizar búsquedas de datos sin coste alguno. A medida que la base de usuarios crezca, se buscará monetizar la aplicación mediante la introducción de publicidad en la web. Se establecerán colaboraciones con marcas y patrocinadores interesados en llegar a una audiencia apasionada por el fútbol.

Concepto	Ingresos esperados
Publicidad por impresiones	2.000,00€
Publicidad por clics	1.500,00€
Colaboraciones con marcas	3.000,00€
Patrocinios	2.500,00€
Total mensual	9.000,00€

Tabla A.6: Ingresos esperados por publicidad y colaboraciones.

Viabilidad legal

En esta sección se expondrán los temas relacionados con las licencias de software, documentación y otros programas utilizados. Una licencia es un contrato que permite a una persona usar, copiar, distribuir, estudiar y modificar ciertos bienes, generalmente no tangibles o intelectuales, como marcas, patentes o software libre. El otorgante conserva la propiedad de

estos bienes, mientras que la persona que obtiene la licencia puede utilizarlos, a menudo a cambio de un pago [16].

Software

Voy a exponer cuál han sido las licencias de las dependencias utilizadas en mi proyecto para llevar a cabo su desarrollo.

Dependencia	Versión	Descripción	Licencia
Angular	17.2.1	Framework para construir aplicaciones web.	MIT
Flask	3.0.2	Microframework de desarrollo web para Python.	BSD-3-Clause
OAuth2	17.0.1	Biblioteca para la implementación del protocolo OAuth2.	MIT
Flex-Layout	15.0.0	Biblioteca para la creación de layouts flexibles en Angular.	MIT
Animate.css	4.1.1	Biblioteca de animaciones CSS listas para usar.	MIT
Moment.js	2.30.1	Biblioteca para manipulación y formateo de fechas en JavaScript.	MIT
Sweetalert2	11.10.6	Biblioteca para mostrar alertas atractivas en JavaScript.	MIT
Angular Material	17.2.1	Componentes de interfaz de usuario para Angular.	MIT
StatsBombPy	1.12.0	API de Python para acceder a los datos de StatsBomb.	MIT
MplSoccer	1.2.2	Biblioteca para la visualización de datos de fútbol en Matplotlib.	MIT
Pandas	1.5.3	Biblioteca de manipulación y análisis de datos en Python.	BSD-3-Clause

Dependencia	Versión	Descripción	Licencia
Matplotlib	3.7.5	Biblioteca de visualización de gráficos en Python.	PSF
NumPy	1.24.4	Biblioteca para computación numérica en Python.	BSD
SciPy	1.10.1	Biblioteca de algoritmos y herramientas matemáticas en Python.	BSD
Scikit-learn	1.3.2	Biblioteca para aprendizaje automático en Python.	BSD-3-Clause
ImageAI	3.0.3	Biblioteca para la implementación de inteligencia artificial en imágenes.	MIT
Ffmpeg	4.4	Herramienta para la manipulación de multimedia.	GPLv3
PostMan	8.6.2	Plataforma para el desarrollo de APIs.	Apache-2.0

Tabla A.7: Dependencias del proyecto.

Por lo tanto, tenemos que escoger una licencia para nuestro proyecto que sea compatible con Apache-2.0, MIT, BSD, PSF y GPLv3. En el siguiente gráfico muestro la compatibilidad entre estas licencias y su grado de permisividad.



Figura A.1: Compatibilidad entre licencias

La licencia más restrictiva como podemos ver que es GPLv3 que posee el programa Ffmpeg.

La forma de monetización del proyecto se realizará mediante la integración de anuncios en la aplicación y colaboraciones con marcas.

Teniendo en cuenta las licencias que usan las dependencias de mi proyecto, la licencia que más se ajusta ajusta a FutboStats es *MIT License*. Esta licencia es permisiva y compatible con las licencias de las dependencias del proyecto y permiten la incorporación de anuncios y colaboraciones comerciales sin complicaciones legales [11].

Derechos en MIT:

- La licencia permite utilizar el software para fines comerciales.
- La licencia permite distribuir copias del software.
- La licencia permite modificar el software.
- La licencia permite el uso privado del software.

Condiciones en MIT:

- Para utilizar los derechos otorgados por la licencia MIT, es necesario incluir la nota de copyright y una copia de la licencia en todas las copias o partes sustanciales del software.

Limitaciones en MIT:

- El software se proporciona sin garantías ni condiciones de ningún tipo, ya sean expresas o implícitas.
- No se ofrecen garantías con el uso del software.

He considerado la licencia MIT como la mejor para mi proyecto porque es permisiva y simple. Permite usar, copiar, modificar, fusionar, publicar, distribuir, vender copias del software. Además tiene alta compatibilidad con otras licencias, siendo totalmente compatible con las licencias que tienen las dependencias utilizadas en mi proyecto.

En cambio, en otras licencias como es Apache License 2.0 he encontrado que posee una alta incompatibilidad con la licencia GPLv2.

La licencia GPLv3 es menos permisiva y no es compatible con algunas licencias como MIT y BSD, por lo que esta licencia queda descartada para ser usada en mi proyecto.

Derechos	Condiciones	Limitaciones
Uso comercial.	Incluir la licencia en todas las copias o partes sustanciales del software.	Limitación de responsabilidad.
Distribución.	Nota sobre la licencia y copyright.	Sin garantías.
Modificación.		
Uso privado.		

Tabla A.8: Resumen de la licencia MIT.

En resumen, dado que mi proyecto es una aplicación que se ofrece de forma gratuita a los usuarios para consultar datos y estadísticas, y teniendo en cuenta las licencias de las dependencias y programas (que incluyen MIT, BSD, Apache-2.0, PSF y GPLv3), la licencia MIT es la más adecuada por ser permisiva y compatible con otras licencias usadas en el proyecto y por permitir la incorporación de anuncios y colaboraciones comerciales.

Documentación

La documentación del proyecto ha sido realizada en LaTex. Para la documentación he decidido no utilizar MIT y optar por otro tipo de licencias que se encuentran más enfocadas en licenciar este tipo de material. Se ha elegido la *Creative Commons Attribution-ShareAlike 4.0 International (CC-BY-SA 4.0)* [2]. Esta licencia establece lo siguiente:

Derechos	Condiciones	Limitaciones
Uso comercial.	Nota sobre la licencia y copyright.	Limitación de responsabilidad.
Distribución.	Indicar modificaciones realizadas.	Sin garantías.
Modificación.		No proporciona derechos sobre marcas registradas.
Uso privado.		No proporciona derechos sobre patentes.

Tabla A.9: Resumen de la licencia CC-BY-4.0.

Imágenes y videos

Todas las imágenes y vídeos utilizados en la documentación son propios del proyecto y tienen la misma licencia que la documentación (CC-BY-4.0). No se ha utilizado ninguna imagen o vídeo de terceros.

Por otro lado, en la aplicación se ha utilizado una fuente de imágenes de terceros:

Fuente	Descripción	Licencia
FlatIcon	Plataforma que proporciona íconos vectoriales gratuitos.	Flaticon Basic License

Tabla A.10: Fuentes de imágenes de terceros.

Resumen

En la siguiente tabla muestro un resumen de las licencias que posee el proyecto.

Recurso	Licencia
Software	MIT
Documentación	CC-BY-4.0
Imágenes	CC-BY-4.0
Vídeos	CC-BY-4.0

Tabla A.11: Resumen de las licencias del proyecto.

Apéndice B

Especificación de Requisitos

B.1. Introducción

Este apéndice recoge la especificación de requisitos que define el comportamiento del sistema desarrollado. Conforma un documento que detalla todas las necesidades y expectativas de los usuarios y el equipo de desarrollo. Este documento sirve como una guía fundamental para el desarrollo del proyecto y asegura que todas las partes involucradas comprendan y acuerden lo que se espera del proyecto desarrollado.

Se han seguido las recomendaciones del estándar IEEE 830-1998, el cual establece que una especificación de requisitos de calidad debe ser [7]:

- **Correcta:** todos los requisitos deben ser precisos y exactos.
- **Completa:** todos los requisitos necesarios para el software deben estar incluidos.
- **Inequívoca:** cada requisito debe tener una única interpretación y esta debe ser clara.
- **Consistente:** no debe haber conflictos entre los requisitos.
- **Clasificada por importancia:** los requisitos deben estar ordenados según su prioridad.
- **Verificable:** debe ser posible verificar que el software cumple con cada requisito.
- **Modificable:** la especificación debe estar estructurada de tal manera que los cambios sean fáciles de realizar.

- **Rastreable:** cada requisito debe ser identificable y rastreable a lo largo del ciclo de vida del desarrollo del *software*.

B.2. Objetivos generales

El proyecto persigue los siguientes objetivos generales:

- Desarrollar una aplicación web de fútbol y estadísticas que permita a los usuarios consultar datos y estadísticas de forma interactiva.
- Facilitar al usuario la interpretación de los algoritmos y estadísticas mediante gráficos y tablas.
- Implementar en una aplicación web un sistema de *tracking* para visualizar el movimiento de los jugadores en el campo.
- Proporcionar al usuario información actualizada sobre estadísticas.
- Desarrollar una interfaz de usuario intuitiva.

B.3. Catálogo de requisitos

A continuación, se enumeran los requisitos específicos derivados de los objetivos generales del proyecto.

Requisitos funcionales

- **RF-1 Consultar datos de fútbol:** el usuario debe poder consultar datos de fútbol sobre partidos, equipos y clasificaciones fácilmente.
 - **RF-1.1 Consultar equipos:** el usuario debe poder consultar equipos de fútbol.
 - **RF-1.1.1 Obtener información del equipo:** la aplicación debe poder mostrar información detallada de los equipos.
 - **RF-1.1.2 Obtener datos de los jugadores del equipo:** la aplicación debe poder mostrar información de los jugadores de los equipos.
 - **RF-1.2 Buscar partidos de fútbol:** el usuario debe poder buscar partidos de fútbol.

- **RF-1.2.1 Buscar resultados de partidos:** la aplicación debe poder mostrar los resultados de los partidos.
- **RF-1.3 Buscar clasificaciones:** el usuario debe poder buscar clasificaciones.
 - **RF-1.3.1 Ver tabla de posiciones:** la aplicación debe poder mostrar la tabla de posiciones.
- **RF-2 Consultar estadísticas y predicciones de fútbol:** el usuario debe poder consultar estadísticas y predicciones de fútbol, incluyendo puntos esperados, modelo de goles esperados y estadísticas del Mundial 2022.
 - **RF-2.1 Consultar predicciones de puntos esperados:** el usuario debe poder consultar predicciones de puntos esperados.
 - **RF-2.2 Consultar modelo de goles esperados:** el usuario debe poder consultar el modelo de goles esperados.
 - **RF-2.3 Consultar estadísticas del Mundial 2022:** el usuario debe poder consultar estadísticas del Mundial 2022.
 - **RF-2.4 Utilizar modelo de *tracking* en vídeos de fútbol:** el usuario debe poder aplicar *ImageAI* en vídeos de fútbol.
- **RF-3 Autenticación:** el usuario debe poder iniciar sesión en la aplicación.
 - **RF-3.1 Iniciar sesión:** el usuario debe poder iniciar sesión en la aplicación con Google.

Requisitos no funcionales

- **RNF-1 Rendimiento:**
 - **RNF-1.1:** La aplicación debe cargar las páginas de búsqueda en un tiempo menor a 3 segundos.
 - **RNF-1.2:** Las consultas a la API-FOOTBALL deben responder en un tiempo menor a 5 segundos.
- **RNF-2 Seguridad:**
 - **RNF-2.1:** La autenticación de usuarios debe ser segura y cumplir con los estándares de autenticación, como OAuth2.

- **RNF-3 Usabilidad:**

- **RNF-3.1:** La interfaz de usuario debe ser intuitiva y fácil de usar.
 - **RNF-3.2:** La aplicación debe ser accesible desde dispositivos móviles y de escritorio.

- **RNF-4 Escalabilidad:**

- **RNF-4.1:** La aplicación debe poder manejar varios usuarios de forma concurrente sin degradación significativa en el rendimiento.

- **RNF-5 Mantenibilidad:**

- **RNF-5.1:** El código de la aplicación debe seguir las mejores prácticas de desarrollo para facilitar el mantenimiento y las actualizaciones futuras.
 - **RNF-5.2:** La aplicación debe tener documentación técnica actualizada para desarrolladores.

- **RNF-6 Compatibilidad:**

- **RNF-6.1:** La aplicación debe ser compatible con los navegadores web más utilizados (Chrome, Firefox, Safari, Edge).

- **RNF-7 Fiabilidad:**

- **RNF-7.1:** La aplicación debe tener una gran disponibilidad.
 - **RNF-7.2:** Debe existir un sistema de *backup* para restaurar datos en caso de fallo.

- **RNF-8 Internacionalización:**

- **RNF-8.1:** La aplicación debe estar preparada para soportar varios idiomas, modificando textos, imágenes, unidades de medida, etc..

B.4. Especificación de requisitos

En esta sección se mostrará el diagrama de casos de uso resultante y se desarrollará cada uno de ellos.

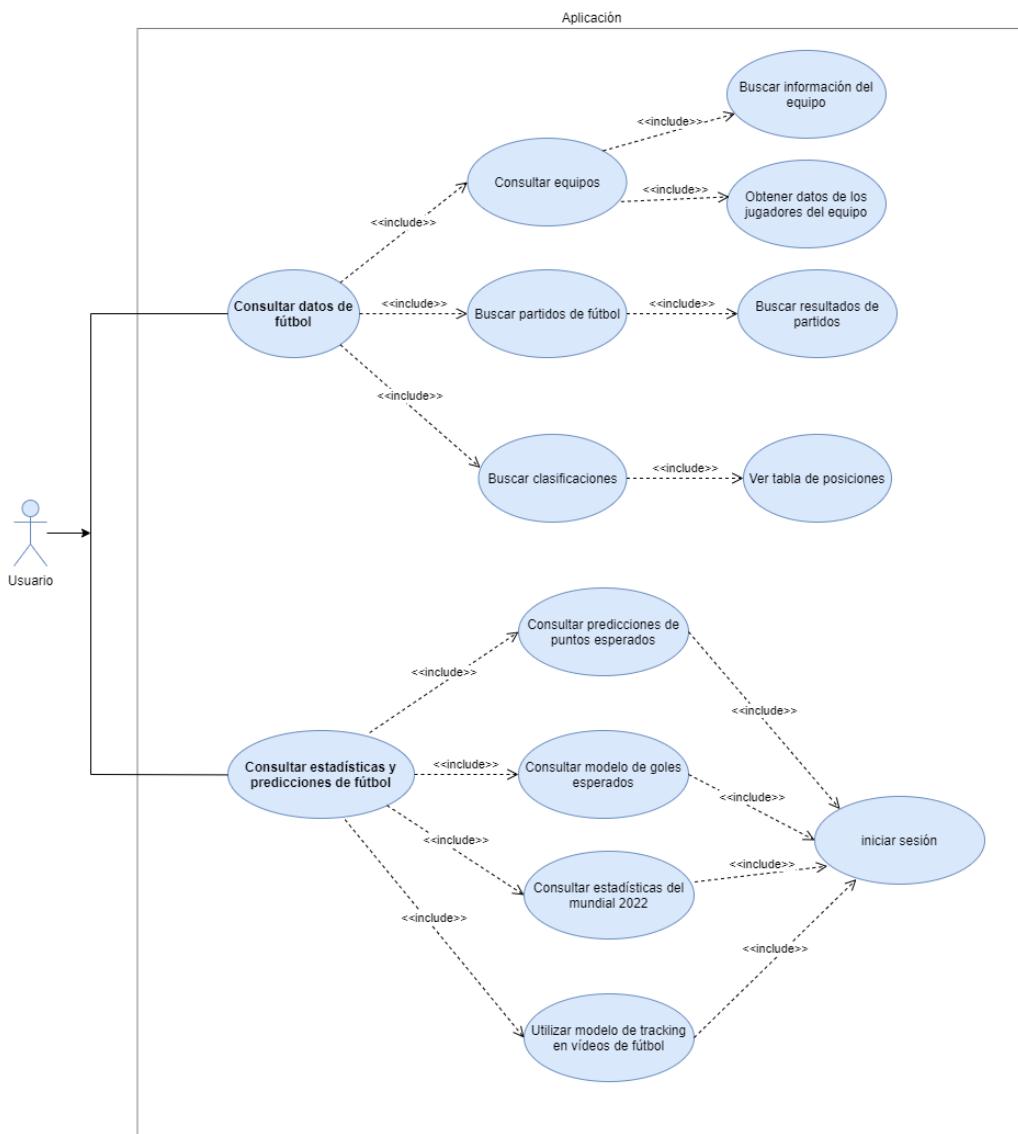


Figura B.1: Diagrama de casos de uso

Diagrama de casos de uso

Actores

El único actor que se contempla en los casos de uso se corresponderá con la figura del usuario.

CU-1 Consultar datos de fútbol	
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-1
Descripción	El usuario puede consultar datos sobre partidos, equipos y clasificaciones.
Precondición	El usuario debe tener acceso a la aplicación.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona una de las opciones para consultar datos de fútbol.2. El sistema muestra opciones para consultar datos sobre partidos, equipos y clasificaciones.
Postcondición	El usuario obtiene la información solicitada.
Excepciones	Error en la conexión con la API.
Importancia	Alta

Tabla B.1: CU-1 Consultar datos de fútbol.

CU-1.1	Consultar equipos
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-1.1
Descripción	El usuario puede consultar información detallada de los equipos de fútbol.
Precondición	El usuario debe tener acceso a la aplicación.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de consultar equipos.2. El usuario introduce el nombre de un equipo.3. El sistema muestra la información del equipo introducido por el usuario.
Postcondición	El usuario visualiza la información del equipo seleccionado.
Excepciones	<ol style="list-style-type: none">1. El equipo seleccionado no tiene información disponible.2. Error en la conexión con la API.
Importancia	Alta

Tabla B.2: CU-1.1 Consultar equipos.

CU-1.1.1 Obtener datos de los jugadores del equipo	
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-1.1.2
Descripción	La aplicación muestra información de los jugadores del equipo seleccionado.
Precondición	El usuario debe haber seleccionado un equipo.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona un equipo.2. El sistema muestra la lista de jugadores del equipo.
Postcondición	El usuario visualiza la lista de jugadores y su información.
Excepciones	Información no disponible para los jugadores del equipo.
Importancia	Media

Tabla B.3: CU-1.1.1 Obtener datos de los jugadores del equipo.

CU-1.2		Buscar partidos de fútbol
Versión	1.0	
Autor	Miguel Ángel Extremo Cabornero	
Requisitos asociados	RF-1.2	
Descripción	El usuario puede buscar información sobre partidos de fútbol.	
Precondición	El usuario debe tener acceso a la aplicación.	
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de buscar partidos de fútbol.2. El sistema muestra un formulario de búsqueda de partidos.3. El usuario ingresa los criterios de búsqueda y envía el formulario.4. El sistema muestra los resultados de la búsqueda.	
Postcondición	El usuario visualiza los partidos de fútbol que coinciden con los criterios de búsqueda.	
Excepciones	No se encuentran partidos que coincidan con los criterios de búsqueda.	
Importancia	Alta	

Tabla B.4: CU-1.2 Buscar partidos de fútbol.

CU-1.3	Buscar clasificaciones
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-1.3
Descripción	El usuario puede buscar clasificaciones de fútbol.
Precondición	El usuario debe tener acceso a la aplicación.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de buscar clasificaciones.2. El usuario comienza a escribir el nombre de una liga.3. El sistema muestra sugerencias de ligas al usuario.
Postcondición	El usuario visualiza la clasificación de la liga.
Excepciones	No se encuentran clasificaciones disponibles.
Importancia	Alta

Tabla B.5: CU-1.3 Buscar clasificaciones.

CU-2		Consultar estadísticas y predicciones de fútbol
Versión	1.0	
Autor	Miguel Ángel Extremo Cabornero	
Requisitos asociados	RF-2	
Descripción		El usuario puede consultar estadísticas y predicciones de fútbol, incluyendo puntos esperados, modelo de goles esperados y estadísticas del Mundial 2022.
Precondición		El usuario debe tener acceso a la aplicación y haber iniciado sesión.
Acciones		<ol style="list-style-type: none">1. El usuario selecciona la opción de consultar estadísticas y predicciones.2. El sistema muestra opciones para consultar puntos esperados, modelo de goles esperados y estadísticas del Mundial 2022.
Postcondición		El usuario obtiene las estadísticas y predicciones solicitadas.
Excepciones		Error en la conexión con la API.
Importancia		Alta

Tabla B.6: CU-2 Consultar estadísticas y predicciones de fútbol.

CU-2.1 Consultar predicciones de puntos esperados	
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-2.1
Descripción	El usuario puede consultar predicciones de puntos esperados para los equipos de fútbol.
Precondición	El usuario debe tener acceso a la aplicación y haber iniciado sesión.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de consultar predicciones de puntos esperados.2. El sistema muestra las predicciones de puntos esperados para las ligas.3. El sistema muestra un gráfico de la posición de los equipos en cada jornada.
Postcondición	El usuario visualiza las predicciones de puntos esperados.
Excepciones	No se encuentran predicciones disponibles.
Importancia	Alta

Tabla B.7: CU-2.1 Consultar predicciones de puntos esperados.

CU-2.2 Consultar modelo de goles esperados	
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-2.2
Descripción	El usuario puede consultar el modelo de goles esperados.
Precondición	El usuario debe tener acceso a la aplicación y haber iniciado sesión.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de consultar modelo de goles esperados.2. El sistema muestra el modelo de goles esperados..
Postcondición	El usuario visualiza el modelo de goles esperados.
Excepciones	No se encuentra el modelo de goles esperados.
Importancia	Alta

Tabla B.8: CU-2.2 Consultar modelo de goles esperados.

CU-2.3	Consultar estadísticas del Mundial 2022
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-2.3
Descripción	El usuario puede consultar estadísticas del Mundial 2022.
Precondición	El usuario debe tener acceso a la aplicación y haber iniciado sesión.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de consultar estadísticas del Mundial 2022.2. El sistema muestra las estadísticas del Mundial 2022.3. El usuario puede elegir de que jugador quiere visualizar el gráfico de estadísticas del Mundial 2022.4. El usuario puede comprender los gráficos fácilmente.
Postcondición	El usuario visualiza las estadísticas del Mundial 2022.
Excepciones	No se encuentran estadísticas disponibles.
Importancia	Media

Tabla B.9: CU-2.3 Consultar estadísticas del Mundial 2022.

CU-2.4 Utilizar modelo de tracking en vídeos de fútbol	
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-2.4
Descripción	El usuario puede utilizar el modelo de tracking en vídeos de fútbol.
Precondición	El usuario debe tener acceso a la aplicación y haber iniciado sesión.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de utilizar modelo de tracking (ImageAI).2. El sistema solicita al usuario que cargue un vídeo de fútbol.3. El usuario carga el vídeo.4. El sistema procesa el vídeo utilizando el modelo de ImageAI.5. El sistema muestra el vídeo procesado al usuario.
Postcondición	El usuario visualiza el vídeo con el tracking aplicado.
Excepciones	Error en el procesamiento del vídeo.
Importancia	Alta

Tabla B.10: CU-2.4 Utilizar modelo de tracking en vídeos de fútbol.

CU-3	Autenticación
Versión	1.0
Autor	Miguel Ángel Extremo Cabornero
Requisitos asociados	RF-3
Descripción	El usuario puede iniciar sesión en la aplicación.
Precondición	El usuario debe tener una cuenta de Google y acceso a la aplicación.
Acciones	<ol style="list-style-type: none">1. El usuario selecciona la opción de iniciar sesión.2. El sistema muestra un botón de inicio de sesión.3. El usuario inicia sesión con su cuenta de Google.4. El sistema verifica que la cuenta es correcta e inicia sesión.
Postcondición	El usuario está autenticado y puede acceder a funcionalidades protegidas.
Excepciones	Credenciales incorrectas o error en la autenticación.
Importancia	Alta

Tabla B.11: CU-3 Autenticación.

Apéndice C

Especificación de diseño

C.1. Introducción

Este apéndice describe el diseño detallado de la aplicación FutboStats. FutboStats es una aplicación web que permite a los usuarios buscar información sobre partidos de fútbol, equipos, y competiciones, así como consultar estadísticas avanzadas y predicciones. Este documento cubre el diseño de datos, el diseño procedural y el diseño arquitectónico de la aplicación.

C.2. Diseño de datos

FutboStats no utiliza una base de datos persistente, sino que se basa en la interacción con API-FOOTBALL y la API en Flask para obtener los datos necesarios. Los datos se manejan de la siguiente manera:

Estructura de solicitudes y respuestas de API-FOOTBALL

Las solicitudes a la API-FOOTBALL se estructuran en formato JSON y contienen parámetros como el nombre del equipo, nombre de la liga o fecha del partido. Estas solicitudes requieren incluir una clave de API (*API key*) y un host de API (*API host*) en los encabezados para autenticación. Cada endpoint de la API tiene parámetros establecidos, algunos de los cuales son opcionales. Para acceder a API-FOOTBALL, es necesario incluir en cada solicitud los siguientes encabezados:

- **API key:** Clave única proporcionada al registrarse en la API-FOOTBALL, utilizada para autenticar las solicitudes.
- **API host:** Identificador del host de la API, que especifica el servidor al cual se envían las solicitudes.

Cada endpoint de la API-FOOTBALL acepta diferentes parámetros, que pueden ser obligatorios u opcionales. Algunos de los parámetros comunes incluyen:

- **Identificador del equipo:** identificador del equipo para el cual se desean obtener datos.
- **Identificador de la liga:** identificador de la liga para la cual se quieren obtener estadísticas o clasificaciones.
- **Fecha:** fecha del partido para el cual se desean obtener los resultados o detalles.
- **Season:** año de la temporada por la que se desea filtrar la información.

A continuación, en la figura C.1 se muestra un ejemplo de cómo se estructura una solicitud en formato JSON realizada mediante la herramienta Postman:

The screenshot shows a Postman request configuration for a GET request to the API endpoint `https://api-football-v1.p.rapidapi.com/v3/standings?league=140&season=2023`. The Headers tab is selected, displaying three checked items: `Connection` (value: `keep-alive`), `X-RapidAPI-Key` (value: `your-api-key`), and `X-RapidAPI-Host` (value: `api-football-v1.p.rapidapi.com`). The Body tab shows the JSON response received from the API, which includes parameters, errors, results, paging information, and a response array containing a single league object with the value `"league": "140"`.

```

1
2   "get": "standings",
3   "parameters": {
4     "league": "140",
5     "season": "2023"
6   },
7   "errors": [],
8   "results": 1,
9   "paging": {
10     "current": 1,
11     "total": 1
12   },
13   "response": [
14     {
15       "league": {
16         "id": "140"
17       }
18     }
19   ]
20 }

```

Figura C.1: Ejemplo de petición a API-FOOTBALL

Como podemos ver se realiza una petición GET al *endpoint* '/standings' para obtener una clasificación en concreto. Le pasamos como parámetros el identificador de liga 140(es el identificador de La Liga española) y la season 2023(año de la clasificación). Para que la petición tenga éxito tenemos que establecer en los headers el campo 'X-RapidAPI-Key' con la Api-key que da API-FOOTBALL al registrarse y el campo 'X-RapidAPI-Host' con el host correspondiente que en mi caso es 'api-football-v1.p.rapidapi.com'. Como podemos observar en el body observamos el resultado en JSON de la petición GET y el status de la petición que ha sido un 200 OK.

Estructura de solicitudes y respuestas de la API en Flask

La API diseñada en Flask contiene varios *endpoints*. Algunos devuelven un JSON con la información del modelo de goles esperados o con la información de las tablas de los puntos esperados. En cambio, otros devuelven imágenes generadas en Python utilizando los datos de StatsBomb.

En la figura C.2 muestro un ejemplo de petición a la API creada en Flask:

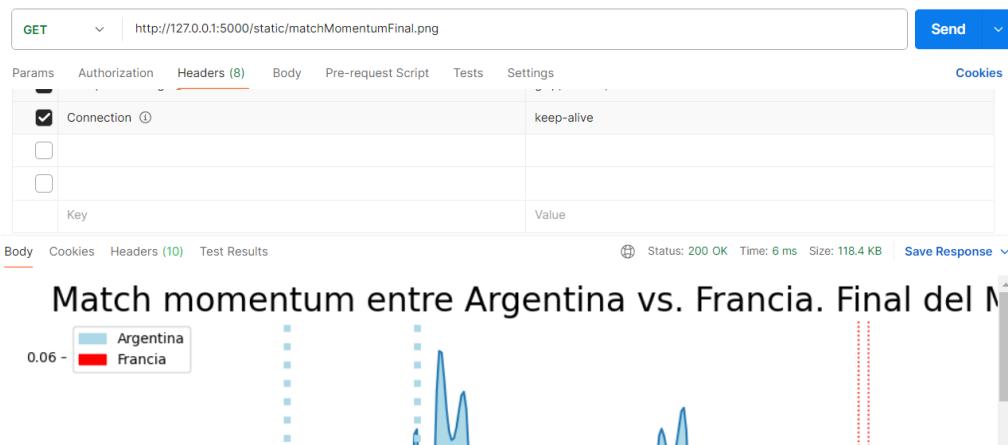


Figura C.2: Ejemplo de petición a la API de Flask

Como podemos ver en la imagen, se realiza una petición a un *endpoint* que devuelve una imagen. En este caso basta con hacer la petición al *endpoint* correcto sin utilizar una Api-key.

Manejo de datos en memoria

Los datos recibidos de la API se almacenan temporalmente en la memoria del cliente (*front-end*) para su visualización y en el servidor (*back-end*) para el procesamiento de predicciones y seguimiento de jugadores con ImageAI. Cuando se realiza una petición a la API-FOOTBALL desde el *front-end* desarrollado en Angular, los datos recibidos se almacenan en memoria en el cliente. Esto permite que la aplicación web pueda mostrar rápidamente la información solicitada por el usuario sin necesidad de realizar repetidas consultas a la API. Los datos en memoria se gestionan mediante servicios de Angular que mantienen el estado de la aplicación.

Cuando se realiza una petición al *back-end*, los datos también se manejan en memoria temporalmente. Además, los datos procesados, como gráficos e imágenes, se guardan en las carpetas 'static' y 'uploads'. La carpeta static almacena imágenes de los gráficos generados, mientras que la carpeta uploads se utiliza para almacenar los vídeos procesados.

En resumen, los datos de la API-FOOTBALL se almacenan temporalmente en la memoria del frontend Angular para una rápida visualización y se manejan en memoria en el *back-end* Flask para el procesamiento de predicciones y vídeos, guardando los resultados en las carpetas 'static' y 'uploads'.

C.3. Diseño procedural

En esta sección se describen los flujos de trabajo y procedimientos clave en FutboStats.

Flujo de búsqueda de datos en API-FOOTBALL

1. El usuario ingresa los criterios de búsqueda (fecha, nombre del equipo, nombre de la liga).
2. El *front-end* envía una solicitud a API-FOOTBALL a través de los servicios de Angular.
3. La API-FOOTBALL devuelve los datos a Angular.
4. Angular procesa los datos y los envía a la vista.
5. Se muestran los resultados al usuario.

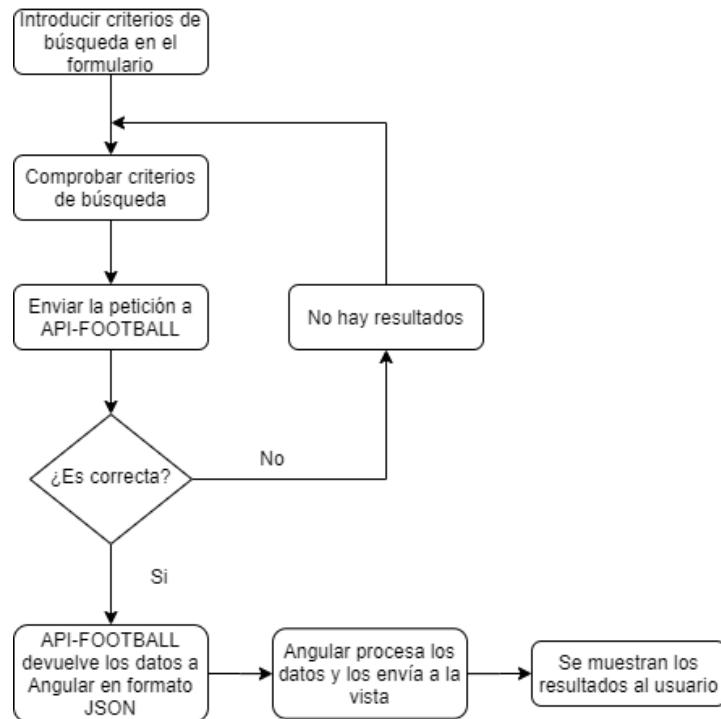


Figura C.3: Diagrama de flujo de obtención de datos a API-FOOTBALL

Proceso de autenticación

1. El usuario selecciona la opción de iniciar sesión.
2. El *front-end* redirige al usuario al proveedor de autenticación (Google).
3. El usuario se autentica y se redirige de vuelta al *front-end* con un token.
4. El *front-end* valida el token utilizando las bibliotecas de autenticación de Google.
5. Se crea una sesión para el usuario.
6. El usuario puede acceder a las funcionalidades protegidas de la aplicación.

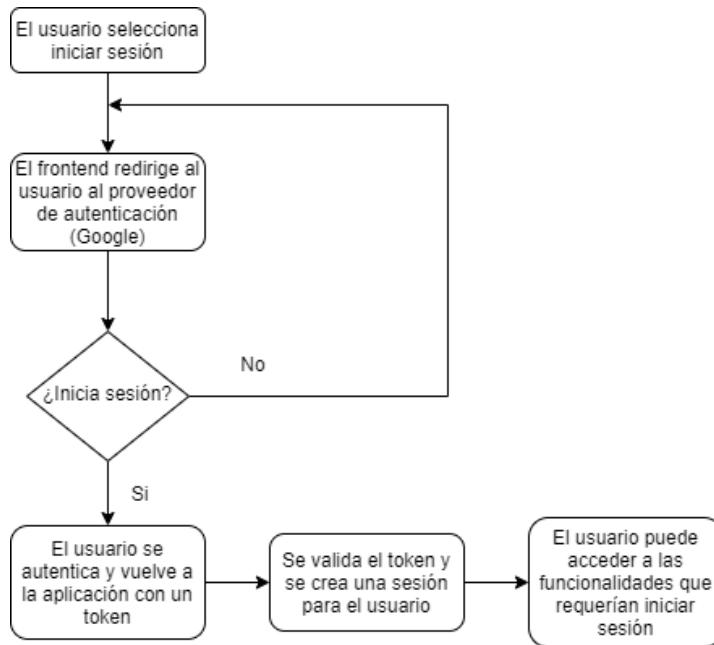


Figura C.4: Diagrama de flujo de inicio de sesión en FutboStats

Flujo de predicciones

1. El usuario inicia sesión y accede a la sección de predicciones.
2. El *front-end* envía una solicitud al *back-end* para obtener los datos de las predicciones(goales esperados, puntos esperados, gráficos del mundial).
3. El *back-end* realiza cálculos utilizando los datos de la librería StatsBombPy.
4. El *back-end* devuelve los resultados del modelo al *front-end*.
5. El *front-end* muestra las predicciones al usuario en forma de gráficos y tablas.

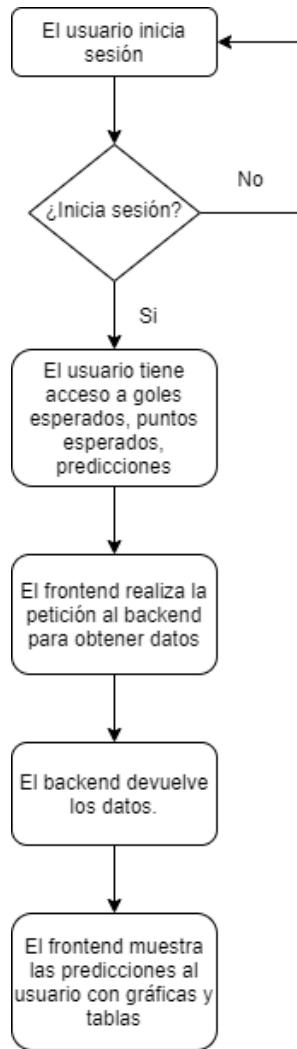


Figura C.5: Diagrama de flujo de obtención de datos en la API de Flask

Procesamiento de vídeos para tracking

1. El usuario inicia sesión y accede a la sección de SportsAI.
2. El usuario sube un vídeo a la aplicación.
3. El *front-end* envía el vídeo al *back-end*.
4. El *back-end* procesa el vídeo utilizando el modelo de *tracking* de ImageAI.
5. El *back-end* genera un vídeo procesado con el *tracking* aplicado.

6. El *back-end* envía el vídeo procesado al *front-end*.
7. El *front-end* muestra el vídeo procesado al usuario.

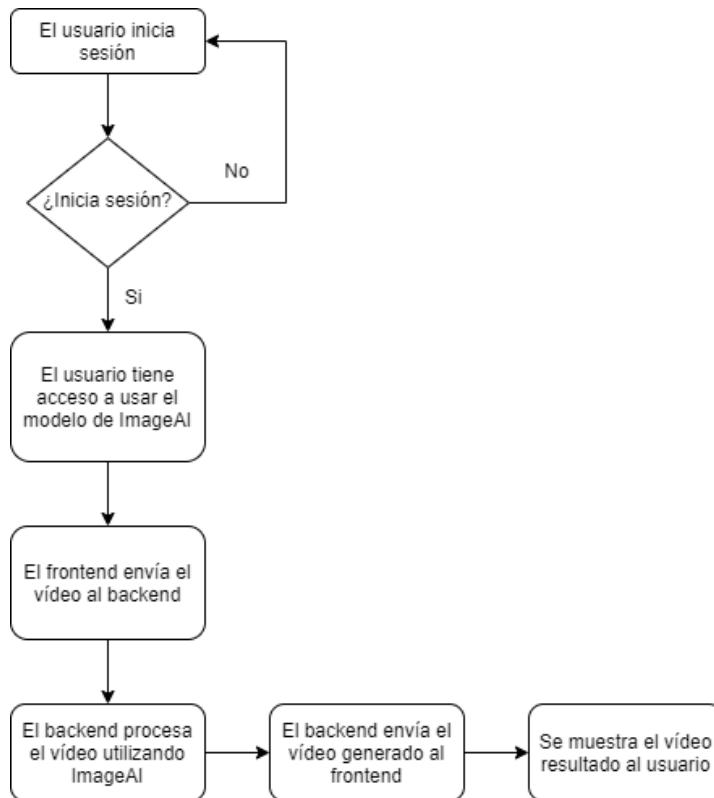


Figura C.6: Diagrama de flujo de uso de ImageAI

C.4. Diseño arquitectónico

FutboStats utiliza una arquitectura cliente-servidor porque divide claramente las responsabilidades entre el *front-end* y el *back-end*, se comunican a través de peticiones HTTP, y facilita la interacción con servicios externos, el procesamiento de datos y la independencia de plataforma [3]. El *front-end* está desarrollado en Angular y el *back-end* en Flask. La aplicación interactúa con la API-FOOTBALL para obtener los datos necesarios.

Descripción general de la arquitectura

La arquitectura de FutboStats se divide en dos capas principales: el *front-end* y el *back-end*. Tiene las siguientes características de una arquitectura cliente-servidor:

- **Separación de responsabilidades:** en FutboStats, el *front-end* (cliente) está desarrollado en Angular y se encarga de la interfaz de usuario y los servicios que hacen las peticiones a API-FOOTBALL, gestionando las interacciones del usuario y la presentación de los datos. El *back-end* (servidor), desarrollado en Flask, maneja la lógica de negocio y el procesamiento de los datos, gráficos, imágenes y videos.
- **Comunicación a través de HTTP:** la arquitectura cliente-servidor implica que el cliente y el servidor se comuniquen a través de peticiones HTTP. En FutboStats, el frontend realiza peticiones HTTP al backend para obtener datos de predicciones, modelos, gráficos y este responde con la información pedida.
- **Independencia de plataforma:** esta arquitectura permite que el cliente y el servidor se ejecuten en plataformas diferentes. El *front-end* puede ser accedido desde cualquier navegador web (desplegado en Netlify), mientras que el *back-end* puede estar alojado en un servidor remoto(desplegado en Render). Esta separación facilita la escalabilidad y el mantenimiento de la aplicación.
- **Procesamiento de datos en el servidor:** el procesamiento de predicciones, generación de gráficos y el tracking de videos, se realizan en el *back-end*. Esto descarga al cliente de tareas intensivas en recursos y permite una mayor eficiencia y seguridad en el manejo de datos.

En la figura C.7 se muestra la arquitectura cliente-servidor.

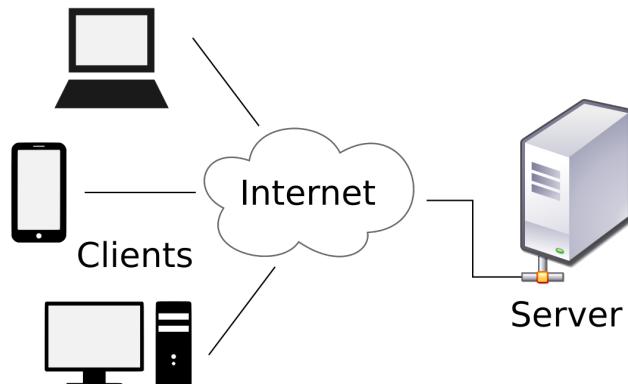


Figura C.7: Arquitectura cliente-servidor [15]

Componentes principales

Frontend (Angular):

- Componente de búsqueda: permite a los usuarios buscar partidos, equipos y clasificaciones mediante la interacción del usuario.
- Componente de visualización: muestra los resultados de búsqueda y estadísticas mediante tablas y tarjetas.
- Componente de autenticación: gestiona el inicio de sesión de los usuarios.
- Servicios que manejan las peticiones a API-FOOTBALL.

Backend (Flask):

- Módulo de predicciones: calcula las predicciones de puntos esperados y goles esperados.
- Generación de gráficos: se generan gráficos con los datos de StatsBombPy que son guardados en la carpeta 'static'.
- Módulo de procesamiento de videos: procesa los videos utilizando ImageAI.

C.5. Diseño de interfaces

En el diseño de la interfaz se han realizado varios prototipos inicialmente utilizando la aplicación *JustInMind*. Estos primeros prototipos sirvieron para definir la paleta de colores que iba a usar la aplicación (verde, blanco y negro) y la distribución de las opciones del menú.

Las siguientes figuras muestran algunos prototipos que fueron desarrollados con *JustInMind*:



Figura C.8: Prototipo de Inicio

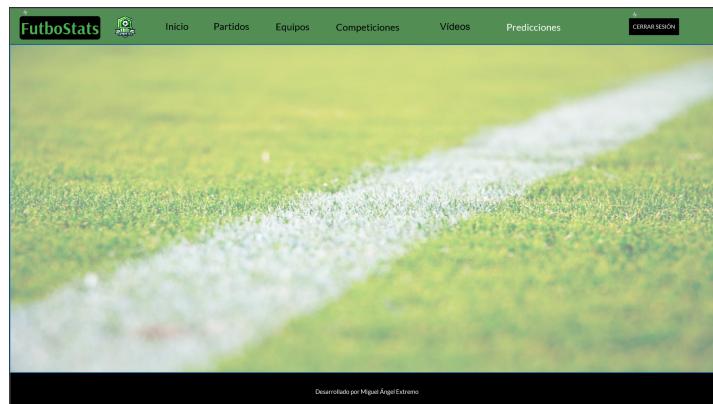


Figura C.9: Prototipo general con header, body y footer



Figura C.10: Prototipo de la vista de ImageAI

Apéndice D

Documentación técnica de programación

D.1. Introducción

Esta sección describe la documentación técnica de programación, incluyendo la instalación del entorno de desarrollo, la estructura de directorios, la estructura de la aplicación y su compilación.

D.2. Estructura de directorios

El repositorio del proyecto se distribuye de la siguiente manera:

- /:
 - .gitignore: Archivo que especifica qué archivos y directorios deben ser ignorados por Git.
 - LICENSE: Copia de la licencia MIT.
 - README.md: archivo que describe la estructura y detalles del proyecto.
- /Códigos/Angular/proyecto/:
 - **Proyecto Angular:** Contiene el código del proyecto Angular junto con los ficheros de configuración del proyecto y el archivo de dependencias.

- **/dist/**: Archivos para la puesta en producción de la aplicación Angular.
- **/src/**:
 - **Código fuente**: Contiene el código fuente de la aplicación Angular junto con su estructura general de ficheros.
 - **/assets/img/**: Imágenes y vídeos utilizados en la aplicación web.
 - **/app/**:
 - ◊ **Módulos compartidos**: Componentes compartidos como el `header` y el `footer`.
 - ◊ **Servicios**: Servicios de la aplicación que se encargan de realizar las llamadas a las APIs.
 - ◊ **Componente login**: Inicio de sesión con Google.
 - ◊ **Interfaces**: Conjunto de interfaces utilizadas en el desarrollo.
 - ◊ **Guards**: Protección de rutas en Angular.
 - ◊ **Módulo aplicacion**: Contiene el resto de componentes de la aplicación.
 - ◊ **/aplicacion/pages**: Componentes que forman las vistas de la aplicación:
 - ◊ **Competiciones**: Vista de competiciones.
 - ◊ **Equipos**: Vista de equipos.
 - ◊ **Goles Esperados**: Vista de goles esperados.
 - ◊ **Inicio**: Vista de inicio.
 - ◊ **Mundial**: Vista del mundial.
 - ◊ **Partidos**: Vista de partidos.
 - ◊ **Predicciones**: Vista de predicciones.
 - ◊ **Sport-IA**: Vista de sport-IA.
 - ◊ Cada directorio contiene sus respectivos ficheros HTML, CSS y TypeScript.
- **/Códigos/Flask/**:
 - **Aplicación Flask**: Contiene la aplicación de Flask.
 - **/static/**: Guarda las imágenes generadas.
 - **/uploads/**: Guarda los vídeos procesados por imageAI.

- **Distintos ficheros JSON:** Contienen los resultados de los modelos de predicción.
- **/flaskApp/:**
 - **Ficheros Python:** Desarrollados para generar el modelo de goles esperados, modelo de puntos esperados, gráficas del mundial y procesado de vídeo con imageAI.
 - **routes.py:** Contiene las rutas definidas en la aplicación Flask.
- **/Memoria/:**
 - **Documentación del proyecto:** Contiene la memoria y los anexos del proyecto en LaTeX y en formato .pdf.
 - **/img/:** Imágenes utilizadas en la documentación.
- **/Prototipos/:**
 - **Prototipos 1.0.vp:** Fichero que contiene los prototipos realizados con la aplicación JustInMind.

D.3. Manual del programador

El siguiente manual tiene como objetivo servir de referencia para futuros desarrolladores que trabajen en la aplicación. En él, se detalla como montar el entorno de desarrollo, obtener el código fuente del proyecto, instalarlo, compilarlo y ejecutarlo en local.

Entorno de desarrollo

Para trabajar con el proyecto se necesita tener instalados los siguientes programas:

- Visual Studio Code.
- Node.js
- Git
- Python 3.8
- Angular

Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito, multiplataforma desarrollado por Microsoft. Se ha convertido en una de las herramientas de desarrollo más populares debido a su versatilidad (permite una amplia variedad de lenguajes de programación), rendimiento y amplio conjunto de características. Su integración con Git facilita el control de versiones, revisión del código y la colaboración en equipo. Podemos descargar Visual Studio Code desde su página oficial [8] eligiendo el sistema operativo y la arquitectura del ordenador y, posteriormente, seguir el asistente de instalación.

Node.js

Node.js es un entorno de ejecución JavaScript que permite a los desarrolladores ejecutar código JavaScript del lado del servidor. Node.js incluye npm, que es la herramienta de gestión de paquetes más utilizada en el ecosistema de JavaScript. Angular y sus dependencias se distribuyen a través de npm, por lo que es esencial tener Node.js instalado para poder instalar Angular y sus bibliotecas asociadas. Se puede descargar la versión de Node.js desde su página oficial y seguir los pasos del instalador [9].

Git

Para poder hacer uso del repositorio en GitHub es necesario tener instalado el gestor de versiones Git. Este programa nos permitirá clonar el repositorio, movernos por sus ramas. Se puede obtener desde [4].

Python

Python es un lenguaje de programación muy popular para realizar cualquier tipo de programa, o aplicación utilizando microframeworks como Flask. Antes de instalar Flask, es necesario tener instalada la versión de Python 3.8.0 debido a que algunas de las dependencias utilizadas requieren de esta versión de Python para funcionar correctamente.

Para instalar Python 3.8.0 se puede hacer desde [12] y seguir los pasos del instalador.

Angular

Angular es un framework de aplicaciones web de código abierto desarrollado por Google. Está diseñado para facilitar la creación de aplicaciones web dinámicas de una sola página. Angular permite a los desarrolladores

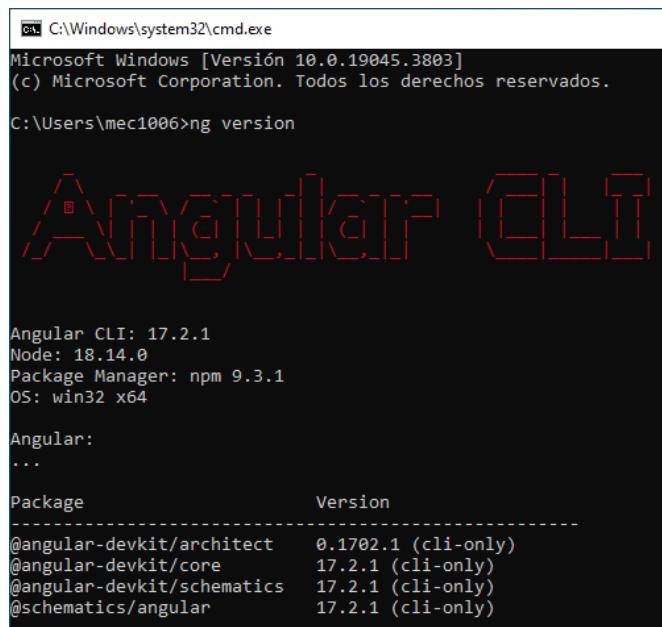
construir aplicaciones web modernas con una arquitectura robusta y un código mantenible.

Para instalar Angular, es necesario tener Node.js y npm instalados en tu sistema, debido a que el Angular CLI (*Command Line Interface*) se distribuye a través de npm.

Una vez instalado Node.js y npm (se instala al instalar Node.js), podremos pasar a instalar Angular en nuestro sistema utilizando el comando '*npm install -g @angular/cli*' en una ventana de comandos.

Una vez instalado, podremos comprobar que se ha instalado con éxito ejecutando el comando '*ng version*' como se puede ver en la figura.

Podemos obtener más información en la página oficial de Angular [1].



The screenshot shows a Windows command prompt window titled 'cmd' with the path 'C:\Windows\system32\cmd.exe'. The title bar also displays 'Microsoft Windows [Versión 10.0.19045.3803]' and '(c) Microsoft Corporation. Todos los derechos reservados.'. The command 'C:\Users\mec1006>ng version' is entered at the prompt. The output shows the Angular CLI logo followed by the following text:

```
Angular CLI: 17.2.1
Node: 18.14.0
Package Manager: npm 9.3.1
OS: win32 x64

Angular:
...
Package          Version
-----
@angular-devkit/architect    0.1702.1 (cli-only)
@angular-devkit/core         17.2.1 (cli-only)
@angular-devkit/schematics   17.2.1 (cli-only)
@schematics/angular          17.2.1 (cli-only)
```

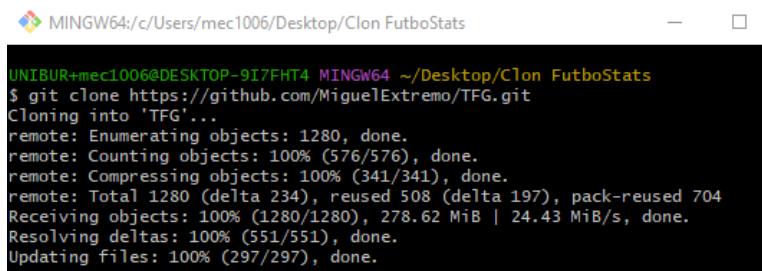
Figura D.1: Ejecución de *ng version* para comprobar la instalación.

D.4. Compilación, instalación y ejecución del proyecto

Obtención del código fuente del repositorio

Para el desarrollo de la aplicación se ha utilizado un repositorio Git hospedado en GitHub. Para obtener una copia de este hay que proceder de la siguiente manera:

1. Abrir la terminal Git Bash.
2. Desplazarse utilizando el comando cd al directorio donde deseemos copiar el repositorio.
3. Introducir el siguiente comando:
`git clone https://github.com/MiguelExtremo/TFG.git`
4. Se iniciará la descarga del repositorio, cuando finalice se dispondrá de una copia de este.



The screenshot shows a terminal window with the following text:

```
MINGW64:/c/Users/mec1006/Desktop/Clon_FutboStats
UNIBUR+mec1006@DESKTOP-9I7FHT4 MINGW64 ~/Desktop/Clon_FutboStats
$ git clone https://github.com/MiguelExtremo/TFG.git
Cloning into 'TFG'...
remote: Enumerating objects: 1280, done.
remote: Counting objects: 100% (576/576), done.
remote: Compressing objects: 100% (341/341), done.
remote: Total 1280 (delta 234), reused 508 (delta 197), pack-reused 704
Receiving objects: 100% (1280/1280), 278.62 MiB | 24.43 MiB/s, done.
Resolving deltas: 100% (551/551), done.
Updating files: 100% (297/297), done.
```

Figura D.2: Clonar repositorio de GitHub.

Para más información sobre el proceso de clonar repositorios se puede consultar [5].

Importar proyecto en Visual Studio Code

Una vez clonado el repositorio y con el código fuente de la aplicación, lo importamos en Visual Studio Code siguiendo los siguientes pasos:

1. Abrir Visual Studio Code
2. Menú File > Open folder
3. Buscamos el directorio donde hemos clonado el repositorio.
4. Abrimos el directorio en Visual Studio Code.

Una vez importado el proyecto en Visual Studio Code pasaremos a explicar como instalar las dependencias de Angular y las dependencias de Flask.

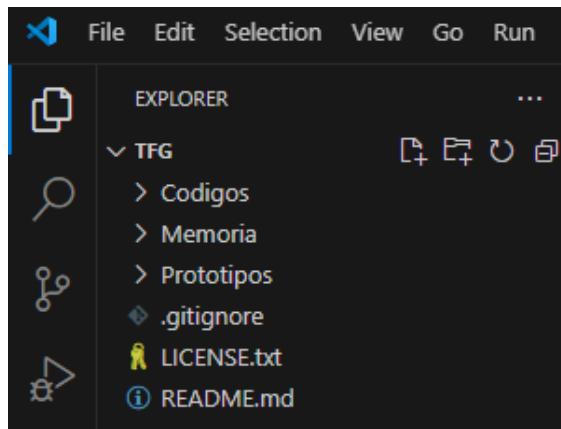


Figura D.3: Importar proyecto en Visual Studio Code

Instalar dependencias de Angular

Para instalar las dependencias de Angular necesitamos generar la carpeta 'node modules'. Este directorio es el más importante en proyectos Angular, ya que contiene todos los módulos y paquetes que el proyecto necesita para funcionar. Cada vez que se instalan dependencias nuevas en un proyecto Angular usando npm, se descargan y se almacenan en la carpeta 'node modules'.

Todos los paquetes y módulos de terceros especificados en el archivo 'package.json' del proyecto se descargan y almacenan en la carpeta 'node modules'. Esto incluye bibliotecas de Angular, utilidades de desarrollo, herramientas de construcción, y cualquier otra dependencia que la aplicación requiera.

Para crear la carpeta node modules se deben seguir los siguientes pasos:

1. Abrir la terminal en la raíz del proyecto Angular.
2. Ejecutar el comando
npm install
3. Se descargarán todas las dependencias listadas en 'package.json' y creará la carpeta 'node modules'.

Este comando puede tardar varios minutos en terminar de ejecutarse. Una vez termine podremos ver la carpeta node modules creada en la raíz del proyecto.

Nombre	Fecha de modificación
dist	02/06/2024 16:50
e2e	02/06/2024 16:50
node_modules	02/06/2024 17:18
src	02/06/2024 16:50
.editorconfig	02/06/2024 16:50
.gitignore	02/06/2024 16:50
angular	02/06/2024 16:50
karma.conf	02/06/2024 16:50
package	02/06/2024 16:50
package-lock	02/06/2024 16:50
README	02/06/2024 16:50
tsconfig.app	02/06/2024 16:50
tsconfig	02/06/2024 16:50
tsconfig.spec	02/06/2024 16:50
tslint	02/06/2024 16:50

Figura D.4: Creación de la carpeta node modules con las dependencias del proyecto Angular

Creación entorno virtual para la aplicación Flask

Crear un entorno virtual es una buena práctica para mantener las dependencias del proyecto aisladas de otras en el sistema. Para crear un entorno virtual y activarle se siguen los siguientes pasos:

1. Abrir una ventana de comandos en el directorio donde queramos crear el entorno virtual.
2. Ejecutar el comando:
python -m venv venv
3. El comando creará un entorno virtual en un directorio llamado 'venv'.
4. Una vez creado nos movemos con cd a venv/Scripts/activate.
5. Veremos que el prompt cambia para indicar que el entorno virtual está activado.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask>python -m venv venv
C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask>cd venv\Scripts\activate
El nombre del directorio no es válido.

C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask>cd venv
C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask\venv>cd Scripts
C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask\venv\Scripts>activate
(venv) C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask\venv\Scripts>
```

Figura D.5: Creación y activación del entorno virtual en Python.

Instalar las dependencias del proyecto Flask

Con el entorno virtual activado seguimos los siguientes pasos:

1. Navegamos con cd al directorio donde se encuentra el fichero 'requirements.txt'.
2. Ejecutamos el siguiente comando para instalar las dependencias:
pip install -r requirements.txt
3. Como resultado habremos instalado las dependencias del proyecto de Flask.
4. Ejecutar comando 'pip list' para comprobar que se han instalado las dependencias.

Este proceso de instalación de las dependencias del proyecto Flask puede tardar unos minutos.

Package	Version
absl-py	2.1.0
annotated-types	0.6.0
asgiref	3.7.2
asttokens	2.4.1
astunparse	1.6.3
attrs	23.2.0
backcall	0.2.0

Figura D.6: Listar dependencias instaladas utilizando el comando *pip list*.

Otras instalaciones: Ffmpeg

Para poder utilizar la herramienta ImageAI sin problemas e integrarla en una aplicación web es necesario instalar la herramienta Ffmpeg. Es un programa utilizado para codificar los formatos de los vídeos que el usuario introduce. Si no se instala este programa, el vídeo procesado por ImageAI y devuelto a la aplicación web no podrá visualizarse correctamente debido a que la codificación de el vídeo no es compatible con los navegadores. Para instalar la herramienta solo hay que seguir los siguientes pasos:

1. Entrar en <https://ffmpeg.org/download.html>.
2. Elegimos nuestro sistema operativo, Windows en este caso y hacemos click en *Windows builds from gyan.dev*.
3. Accedemos a <https://www.gyan.dev/ffmpeg/builds/>
4. Hacemos click en el *ffmpeg-git-full.7z* y se descargará un archivo comprimido con todo lo necesario.
5. Descomprimimos el archivo descargado y lo movemos a el disco *C*:
6. Accedemos al directorio bin/ffmpeg.exe y copiamos la ruta a ese fichero.
7. Abrimos las variables de entorno del sistema.
8. Seleccionamos PATH > Editar > Nuevo.
9. Añadimos la ruta del fichero *ffmpeg.exe*.
10. En una terminal escribimos el comando *ffmpeg* para verificar que se ha instalado correctamente.
11. Se debe añadir a las variables de entorno del sistema para que así cualquier *Script* externo sea capaz de encontrar este programa.

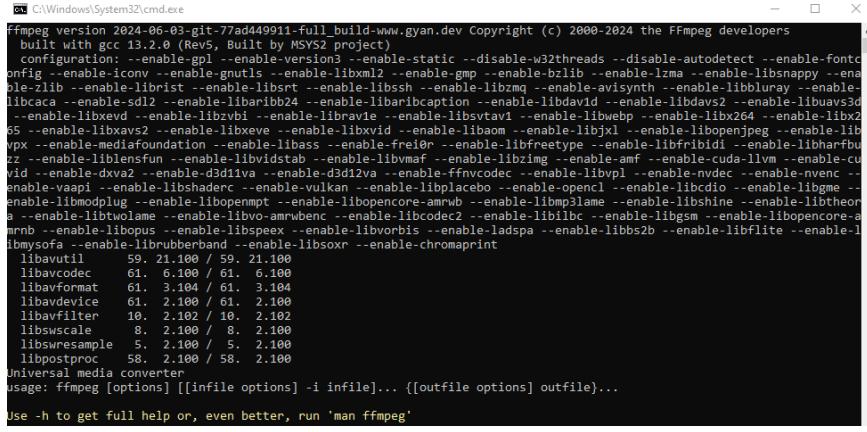


Figura D.7: Comprobación de la instalación de *Ffmpeg*

Ejecutar aplicación en local

Ejecución de aplicación Angular

La ejecución de la aplicación Angular se puede hacer de dos maneras.

1. Mediante el Angular CLI (Command Line Interface).
 2. Arrancar manualmente mediante el fichero *package.json* del proyecto.

Para realizar la ejecución mediante el Angular CLI seguiremos los siguientes pasos:

1. Abrir terminal de comandos (cmd).
 2. Movernos a la raíz del proyecto Angular.
 3. Ejecutar el comando `ng serve -o`. Con este comando ejecutaremos la aplicación y se abrirá directamente en el navegador en el puerto `localhost:4200`.

```

Windows PowerShell
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Angular\proyecto>ng serve -o
✓ Browser application bundle generation complete.

Initial chunk files
+ vendor.js | Names | Raw size
+ styles.css, styles.js | vendor | 6.11 MB
+ main.js | styles | 1.55 MB
+ polyfills.js | main | 622.98 kB
+ runtime.js | polyfills | 337.87 kB
+ runtime.js | runtime | 12.96 kB

+ Initial total | 8.61 MB

Lazy chunk files
+ node_modules_angular_animations_fesm2022_browser_mjs.js | Names | Raw size
+ angular-animations-browser | angular-animations-browser | 177.40 kB

Build at: 2024-06-03T15:15:13.005Z - Hash: 94a0befc13847c01 - Time: 23525ms

Warning: C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Angular\proyecto\src\app\aplicacion\pages\partidos\partidos.component.ts depends on 'moment'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.

```

Figura D.8: Ejecución de la aplicación Angular mediante el comando *ng serve -o* en el *Angular CLI*.

Para realizar la ejecución de la aplicación Angular desde Visual Studio Code seguiremos los siguientes pasos:

1. Abrir el proyecto en Visual Studio Code.
2. Abrir el fichero *package.json*.
3. Buscar la línea donde se encuentra el script *ng serve -o*.
4. Poner el cursor encima de *start* y dar a *Run Script*.

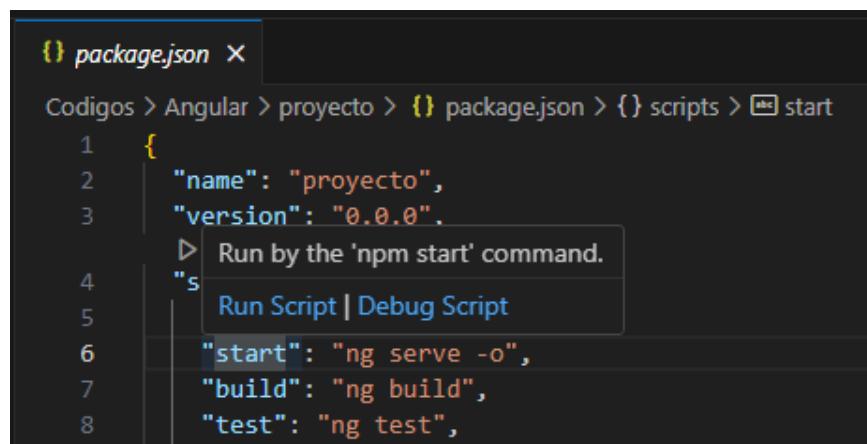
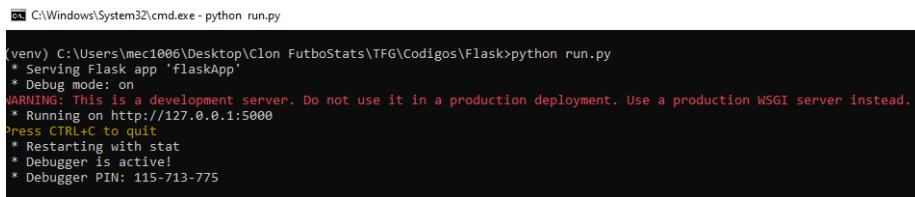


Figura D.9: Ejecución de la aplicación Angular desde Visual Studio Code.

Ejecución de aplicación Flask

Para ejecutar la aplicación Flask usaremos el entorno virtual creado anteriormente. Para ello seguiremos los siguientes pasos:

1. Abrir terminal.
2. Movernos a la carpeta donde se encuentra el entorno virtual.
3. Activar el entorno virtual en *Flask/Scripts/activate*.
4. Una vez activado el entorno virtual, nos movemos a la carpeta que contiene el fichero *run.py* en *Flask/run.py*.
5. Ejecutamos la aplicación utilizando el comando *python run.py*.
6. La aplicación quedará disponible en *http://127.0.0.1:5000*.



```
C:\Windows\System32\cmd.exe - python run.py
(venv) C:\Users\mec1006\Desktop\Clon_FutboStats\TFG\Codigos\Flask>python run.py
 * Serving Flask app 'flaskApp'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 115-713-775
```

Figura D.10: Ejecución de la aplicación Flask utilizando *python run.py*

D.5. Pruebas del sistema

Se han realizado pruebas de validación, pruebas de seguridad y pruebas de sistema para comprobar el correcto funcionamiento de la aplicación.

Pruebas de validación

Se han realizado validaciones de los formularios de consulta de datos de la siguiente manera:

- Verificar que los formularios permiten seleccionar ligas, años, fechas, nombre del equipo y el resultado obtenido es correcto.
- Asegurar que los formularios manejan entradas incorrectas (se muestran mensajes de error en el caso de que no se introduzcan correctamente los parámetros de búsqueda).

- Validar que la información mostrada después de enviar el formulario es correcta y coincide con los datos esperados.

Se han realizado comprobaciones en la funcionalidad de subida y procesado de vídeos:

- Probar funcionalidad de carga de vídeos mediante el explorador de archivos.
- Validar que los vídeos subidos son procesados correctamente y que se devuelve el vídeo procesado por ImageAI.
- Verificar que se manejen adecuadamente errores en la carga de archivos (formatos no soportados, archivos demasiado grandes).

Se han realizado pruebas respecto a las generación de gráficos y estadísticas:

- Validar la correcta generación y visualización de gráficos generados en Python (mapas de calor, mapas de pases, mapas de posiciones, etc).
- Verificar que los modelos de goles esperados y puntos esperados generan resultados aproximados y coherentes con los datos proporcionados.
- Verificar que los gráficos se actualizan correctamente al cambiar el jugador seleccionado.

Pruebas de seguridad

Se han llevado a cabo pruebas de autenticación, manejo de datos sensibles (claves API), autorización de API.

- Verificar que el sistema requiere autenticación para acceder a las funcionalidades restringidas.
- Asegurar que los usuarios solo pueden acceder a los datos y funcionalidades para los que tienen permiso.
- Asegurar que datos como las claves API están protegidas y no se exponen al usuario.
- Verificar que las API utilizadas están correctamente autenticadas y autorizadas.

Otras pruebas de sistema realizadas

- Medir el tiempo de respuesta para la consulta de datos y el envío de gráficos al *front-end*.
- Probar la aplicación en diferentes navegadores (Chrome, Firefox, Edge, Safari) y dispositivos (escritorio, portátil, móvil) para comprobar su compatibilidad.
- Asegurar que la interfaz de usuario se muestra y funciona correctamente en todos los entornos probados.
- Evaluar la facilidad de uso de la interfaz de usuario, asegurándose de que los usuarios pueden navegar y realizar consultas de datos sin problemas.
- Asegurar que los diferentes componentes del sistema (*front-end* en Angular, *back-end* en Flask, procesamiento de vídeos con ImageAI, generación de gráficos con Python) se integran y funcionan correctamente juntos.
- Probar el flujo completo desde la entrada de datos en el formulario hasta la visualización de resultados y gráficos.

Estas pruebas ayudan a garantizar que la aplicación web sea robusta, segura y ofrezca una experiencia de usuario fluida y confiable.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice se detallan los requerimientos de la aplicación y las indicaciones sobre cómo usarla correctamente. El manual de usuario aquí descrito también se aporta en formato vídeo para que se pueda ver de una forma dinámica como funciona la aplicación en tiempo real.

E.2. Requisitos de usuarios

Los requisitos para poder hacer uso de la aplicación son:

- Tener acceso a internet estable para poder utilizar la aplicación.
- Tener un navegador web actualizado (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).
- Contar con un dispositivo móvil o un ordenador (portátil o de sobremesa).
- Tener una cuenta de Google para iniciar sesión y acceder así a las funcionalidades protegidas de la aplicación.

E.3. Instalación

Para usar la aplicación FutboStats solamente es necesario que el usuario tenga un navegador web instalado en su dispositivo. En el navegador web introducir la url de la web: <https://futbostats.netlify.app/>

E.4. Manual del usuario

En esta sección se describe el uso de las diferentes funcionalidades de la aplicación para que un usuario sin previo conocimiento de uso pueda utilizarla sin mayores problemas. Se aportarán imágenes ilustrativas con ejemplos reales de uso de la aplicación.

Inicio

El apartado de 'Inicio' de FutboStats supone una carta de presentación al usuario. Describe de forma resumida las funcionalidades, expone que la aplicación se puede utilizar en móvil, ya que, es responsive y se dispone de pequeños tutoriales integrados en la aplicación que permiten enseñar al usuario como utilizar las distintas funcionalidades haciendo el uso de la aplicación más fácil e intuitiva para el usuario.

Pasos para utilizar la página de 'Inicio':

1. Pulsar en la opción del menú llamada 'Inicio'
2. Consultar vídeos explicativos de las funcionalidades de la aplicación.

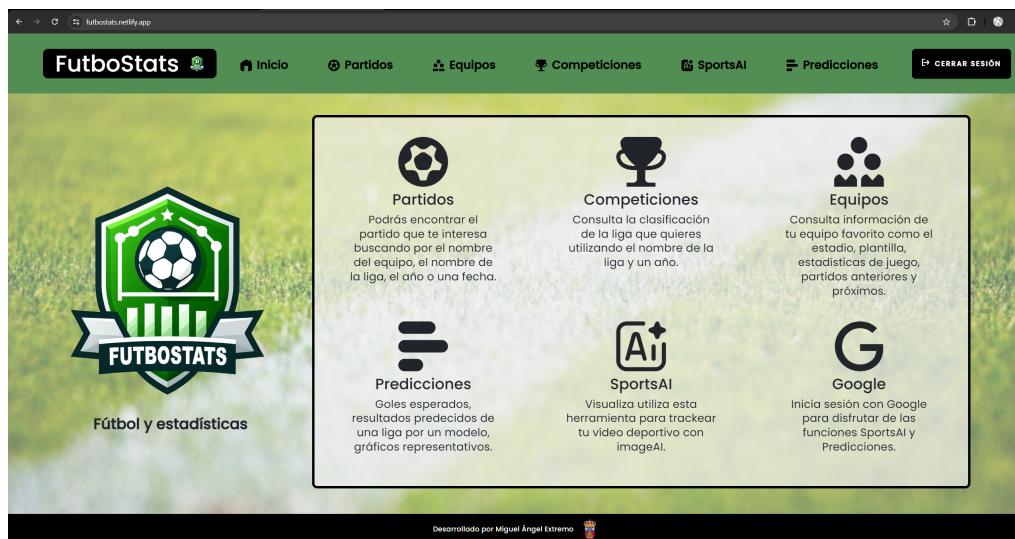


Figura E.1: Inicio de FutboStats

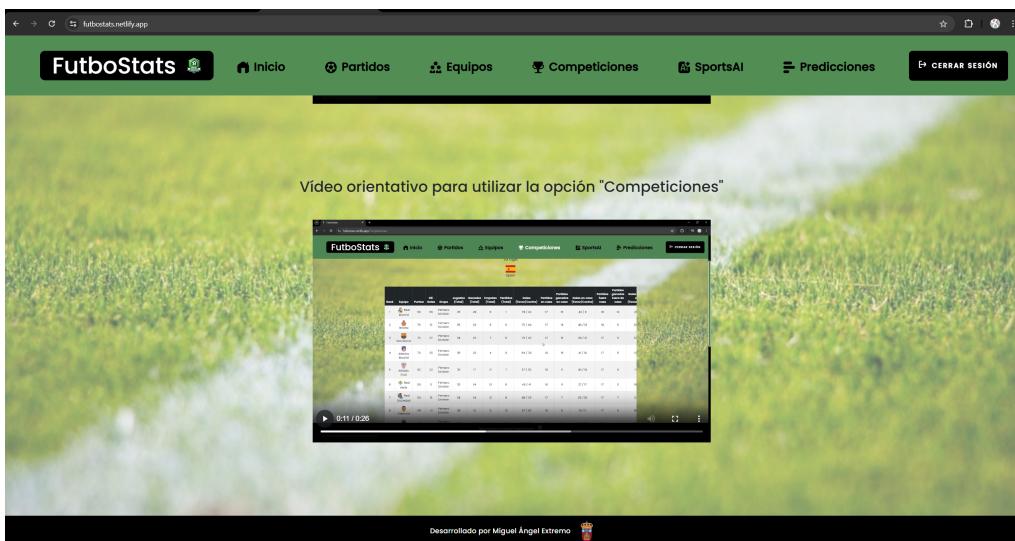


Figura E.2: Tutoriales en Inicio de FutboStats

Buscar partidos

El apartado 'Partidos' de FutboStats proporciona al usuario una forma sencilla de buscar partidos introduciendo unos parámetros de búsqueda. Para buscar partidos el usuario debe:

1. Pulsar en el menú superior la opción 'Partidos'.
2. Introducir el nombre de una liga. El sistema genera una lista de sugerencias al usuario a medida que este escribe.
3. Introducir el nombre de un equipo.
4. Introducir una fecha para filtrar los partidos.
5. Para realizar la búsqueda de partidos es obligatorio introducir el nombre de la liga o el nombre del equipo.
6. La fecha es un parámetro opcional para ofrecer un mayor filtrado.
7. Por último, dar al botón buscar.
8. El usuario visualiza los resultados de la búsqueda.
9. En el caso de no haber partidos con esos filtros, se le informa al usuario mediante un mensaje.

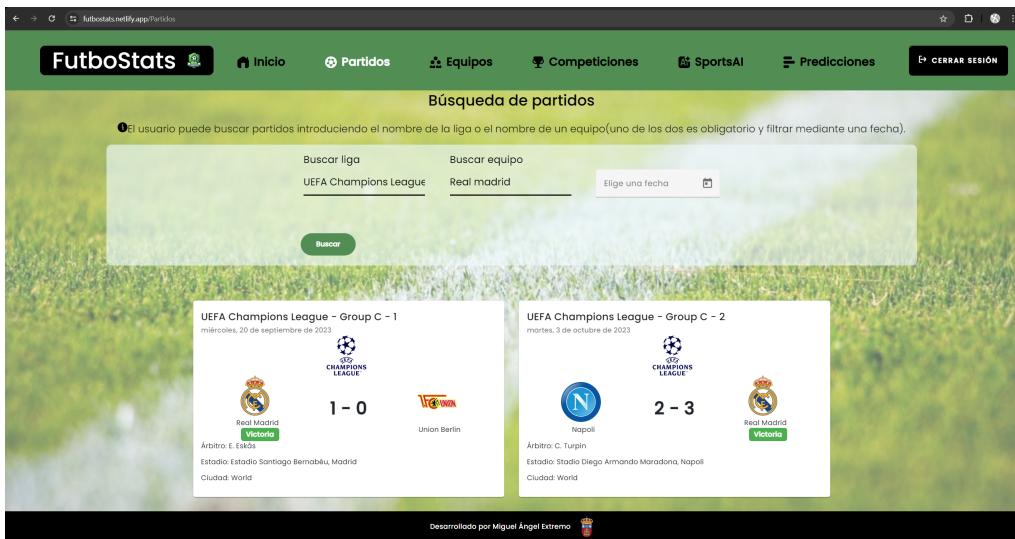


Figura E.3: Búsqueda de partidos en FutboStats

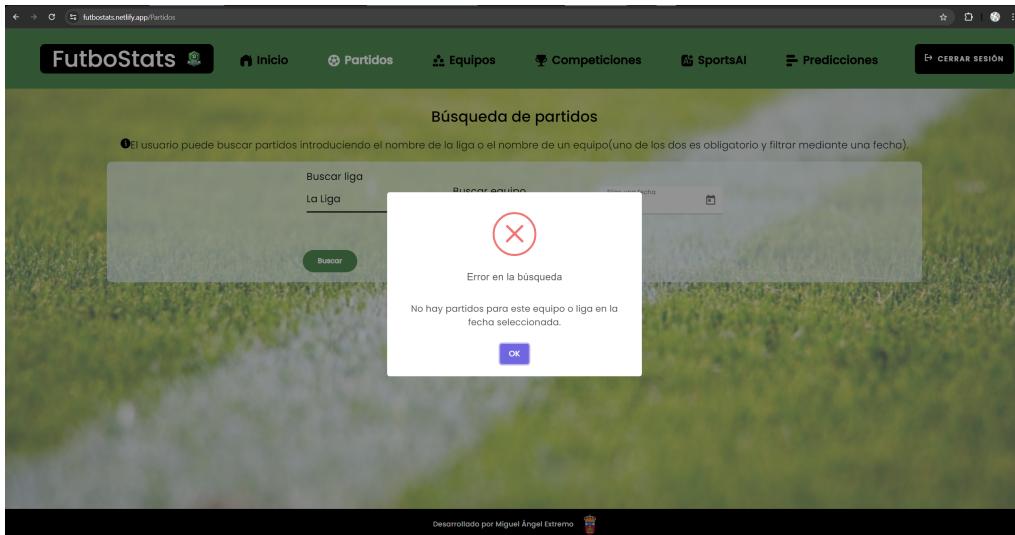


Figura E.4: Búsqueda de partidos en FutboStats sin resultados.

Buscar equipos

El apartado 'Equipos' de FutboStats proporciona al usuario una forma sencilla de buscar información sobre un equipo de fútbol y consultar algunas

estadísticas.

Para buscar equipos el usuario debe:

1. Pulsar en el menú superior la opción 'Equipos'.
2. Introducir el nombre de un equipo.
3. Elegir un año del que quiera ver las estadísticas. Por defecto, se elige el 2023.
4. Darle a enter o al botón de buscar para realizar la búsqueda.
5. El usuario visualizará información del equipo como su escudo, año de fundación, estadio, liga, los cuatro partidos anteriores que ha disputado, los cuatro partidos siguientes que ha disputado, la plantilla del equipo, estadísticas sobre las tarjetas rojas y amarillas, goles a favor y en contra, alineaciones más utilizadas, estadísticas de penaltis.

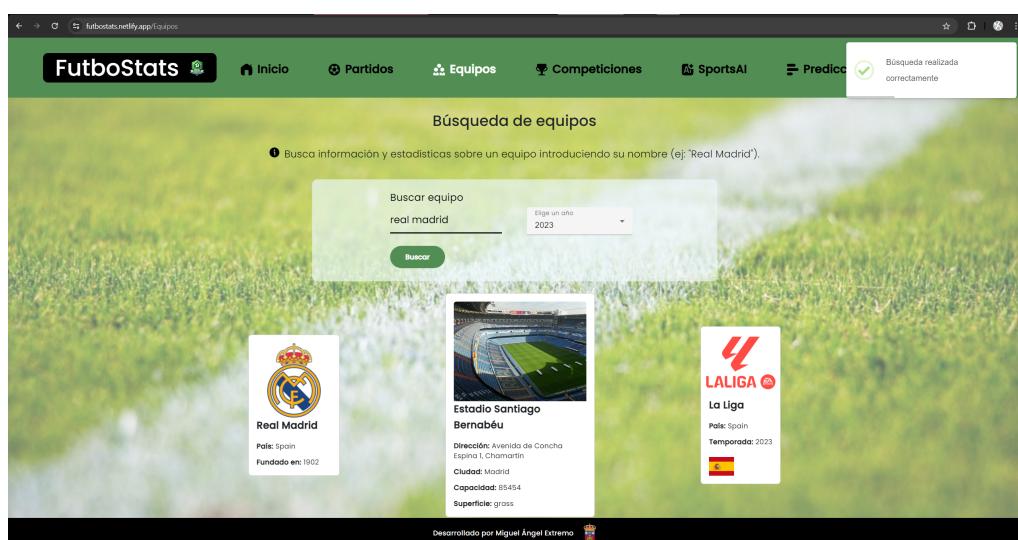


Figura E.5: Búsqueda de equipos en FutboStats.

Buscar clasificaciones

El apartado 'Competiciones' de FutboStats permite al usuario consultar la clasificación de una liga de forma sencilla.

Para buscar clasificaciones el usuario debe:

1. Pulsar en el menú superior la opción 'Competiciones'.
2. Introducir el nombre de la competición. El sistema genera una lista de sugerencias al usuario a medida que este escribe.
3. Pulsar en una de las sugerencias y dar al botón de buscar.
4. Elegir el año del que se quiere ver la clasificación. Por defecto se elige el 2023.
5. El usuario visualizará una tabla que muestra la clasificación de la liga seleccionada.
6. Si se cambia el año, la tabla cambiará para mostrar la información del nuevo año seleccionado.

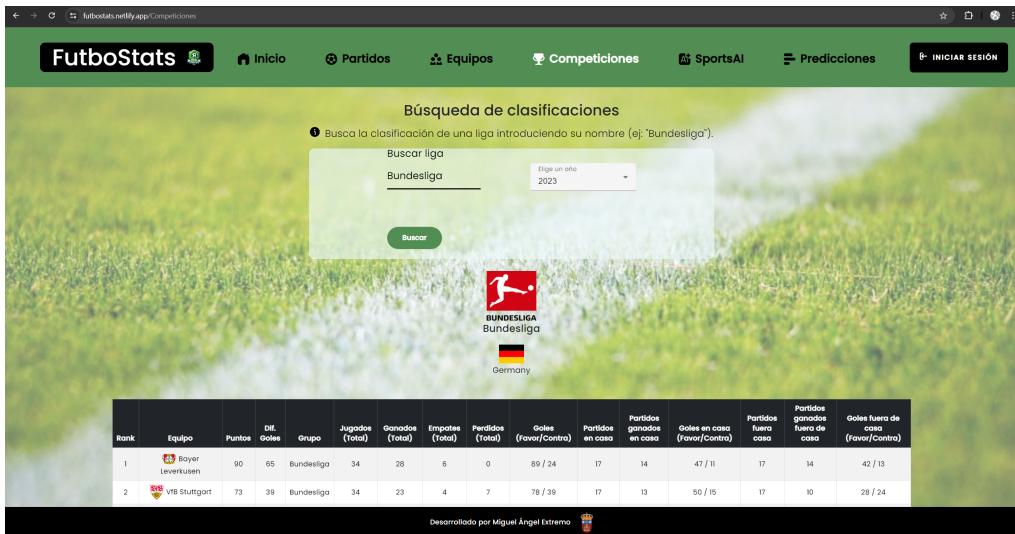


Figura E.6: Búsqueda de competiciones en FutboStats.

Iniciar sesión en FutboStats

Si el usuario quiere utilizar las opciones 'SportsAI' y 'Predicciones' deberá iniciar sesión en la aplicación. La forma de iniciar sesión es muy sencilla, para ello el usuario deberá:

1. Pulsar en el botón de iniciar sesión o en las opciones 'SportsAI' o 'Predicciones', las cuales al no estar autenticado redirigen al usuario a la pantalla de inicio de sesión.

2. En la pantalla de iniciar sesión, pulsar de nuevo en el botón 'Iniciar sesión' dentro del recuadro debajo del logotipo de Google.
3. Se redirige al usuario para que se registre con su cuenta de Google.
4. Una vez registrado, se redirige al usuario de nuevo a la página de 'Inicio' de FutboStats.
5. El usuario puede acceder a las funcionalidades protegidas de la aplicación.

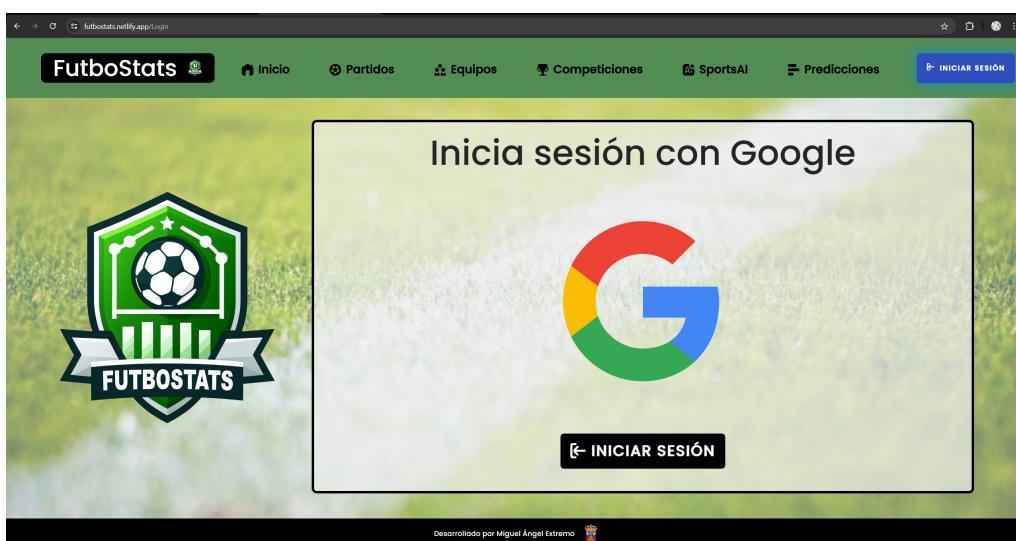


Figura E.7: Paso 1 de iniciar sesión en FutboStats: entrar en iniciar sesión.

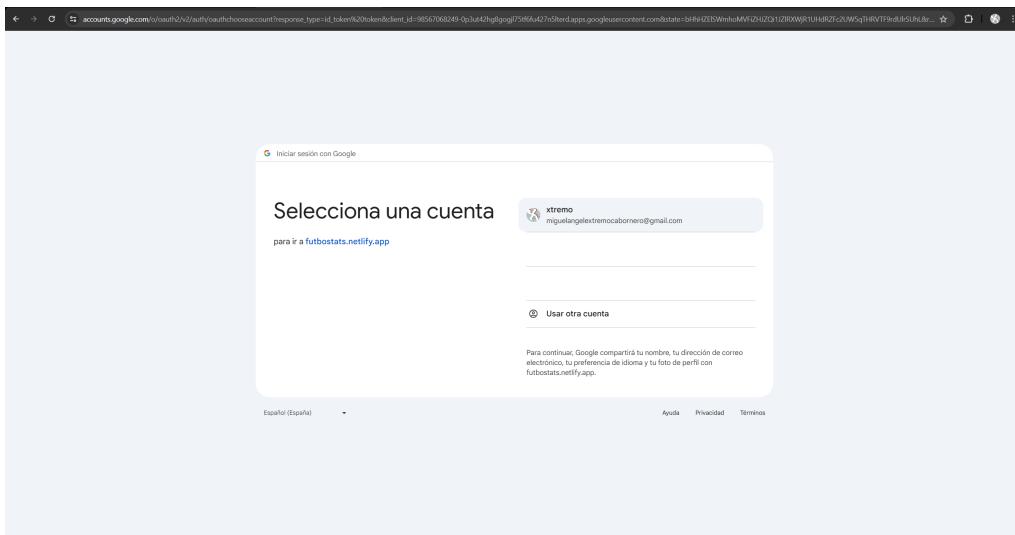


Figura E.8: Paso 2 de iniciar sesión en FutboStats: elegir la cuenta de Google.

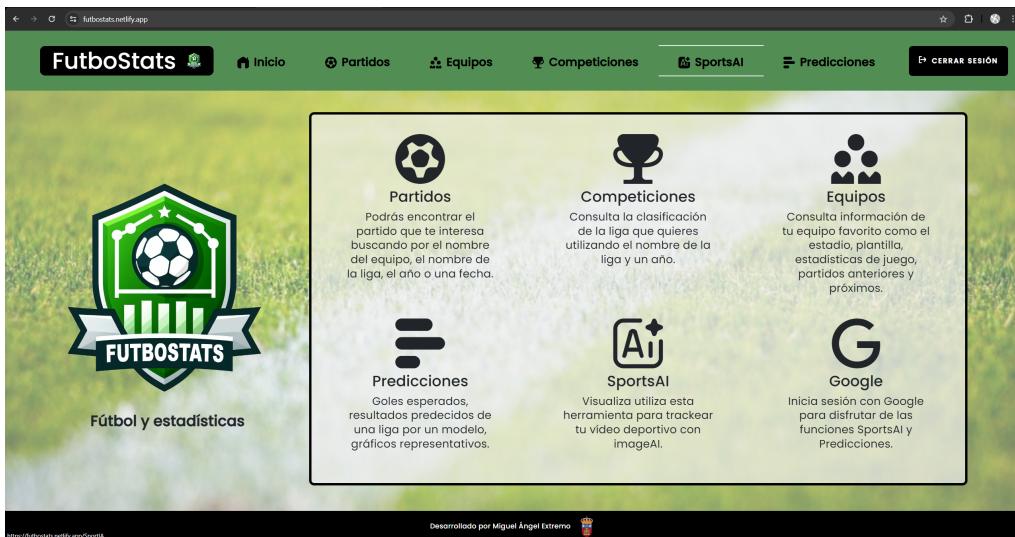


Figura E.9: Paso 3 de iniciar sesión en FutboStats: autenticado con éxito.

Cerrar sesión en FutboStats

Si el usuario desea cerrar la sesión en FutboStats seguirá los siguientes pasos.

1. Pulsar en el botón de 'Cerrar sesión'.
2. Se mostrará una modal con dos opciones: cerrar sesión o cancelar.
3. Si el usuario acepta cerrar sesión se cierra la sesión y no puede acceder a las funcionalidades protegidas de la aplicación.
4. Si el usuario cancela cerrar sesión puede seguir accediendo a las funcionalidades protegidas de la aplicación.

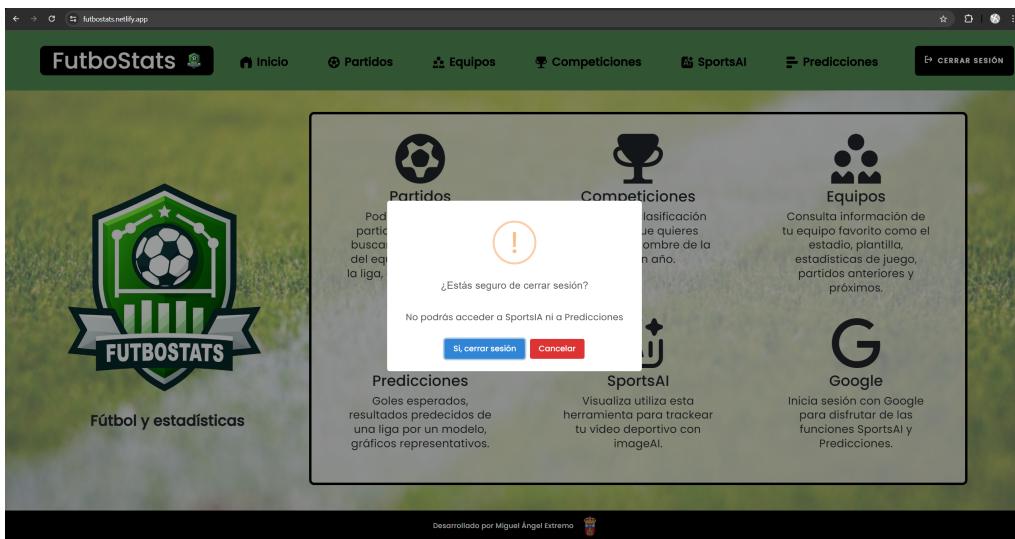


Figura E.10: Cerrar sesión en FutboStats.

Utilizar SportAI en FutboStats

En la opción 'SportsAI' de FutboStats el usuario podrá subir un vídeo suyo en la aplicación y obtener un vídeo con el procesado de imageAI. Los pasos que debe seguir para utilizar esta funcionalidad son:

1. Iniciar sesión en la aplicación.
2. Pulsar en la opción 'SportsAI' del menú superior.
3. Hacer click para elegir o arrastrar un vídeo en formato '.mp4'.
4. Esperar a que la aplicación genere el vídeo procesado por imageAI.

5. El usuario podrá visualizar el vídeo nuevo en la aplicación.

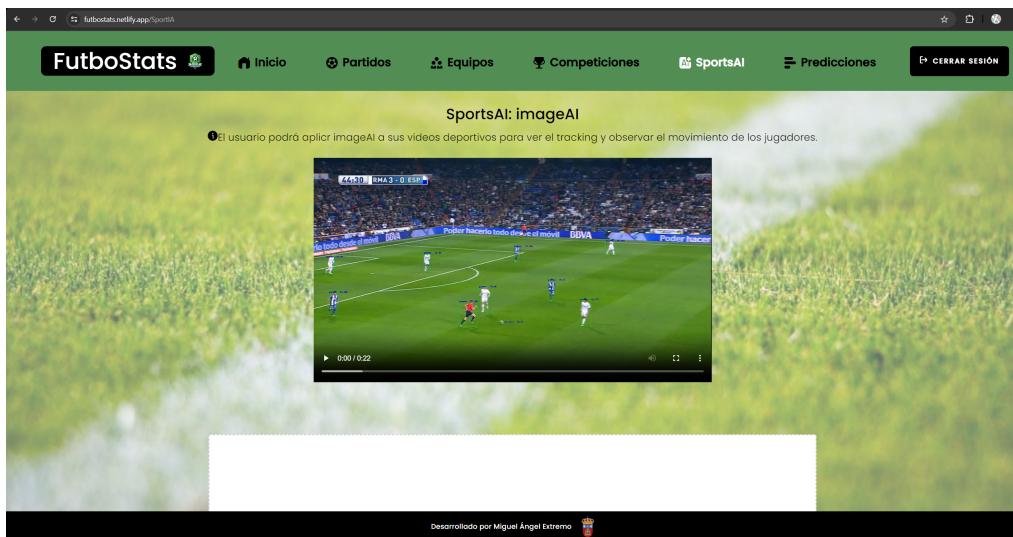


Figura E.11: Paso 1 para utilizar SportAI: entrar en 'SportsAI'.

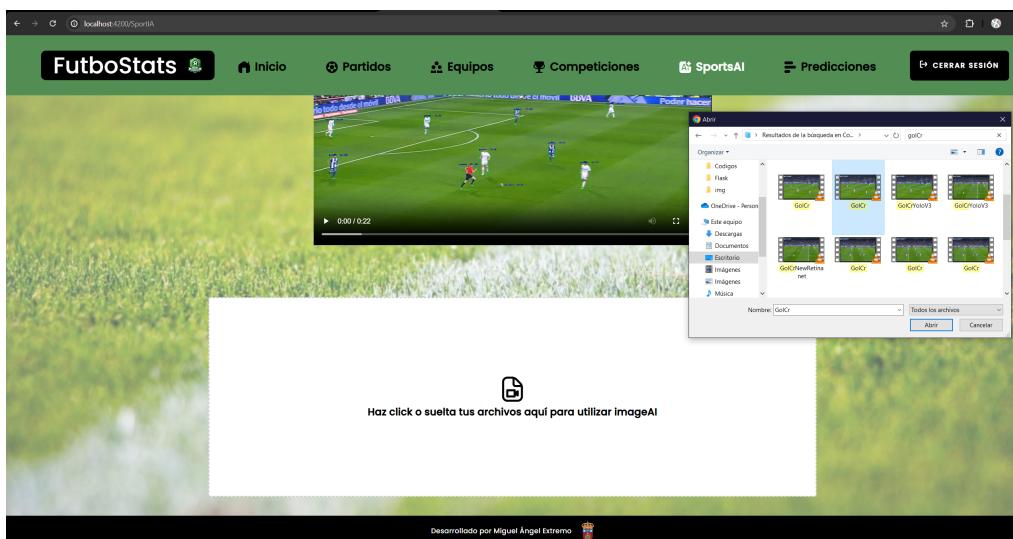


Figura E.12: Paso 2 para utilizar SportAI: pulsar o arrastrar un vídeo '.mp4'.

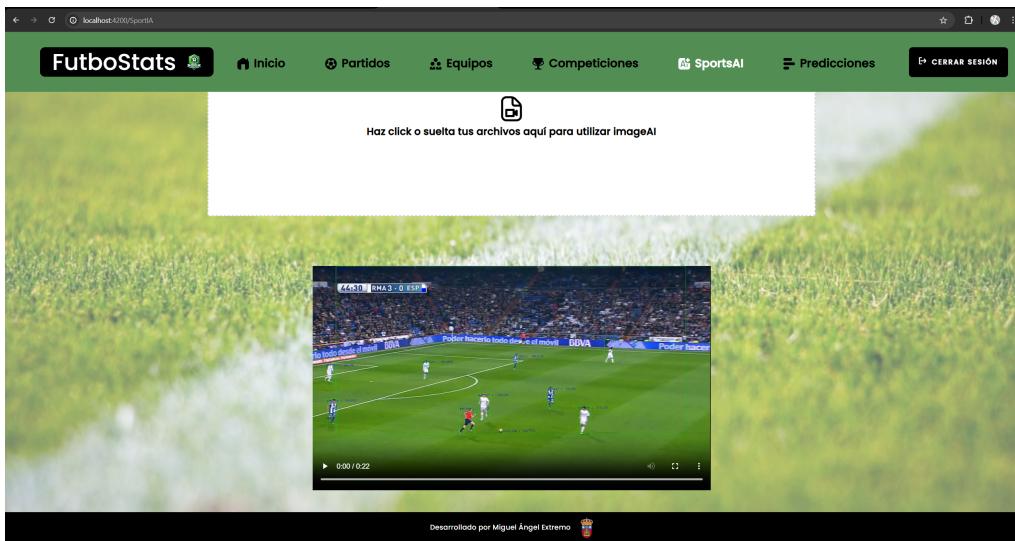


Figura E.13: Paso 3 para utilizar SportAI: visualizar el vídeo generado por imageAI

Utilizar Predicciones en FutboStats

En la opción 'Predicciones' de FutboStats el usuario podrá consultar predicciones, modelo de goles esperados, puntos esperados y gráficas de estadísticas del mundial de una forma sencilla.

Los pasos a seguir son:

1. Iniciar sesión en la aplicación.
2. Pulsar en la opción 'Predicciones' del menú superior.
3. El usuario accede a clasificaciones y puede visualizar una comparación entre la tabla de puntos esperados generada por el modelo y la clasificación real. Además, puede visualizar un gráfico que muestra el desarrollo de los equipos en la liga a lo largo de las jornadas. Las ligas disponibles son La Liga, Bundesliga, Premier League, Serie A y Ligue1. Los datos pertenecen a la temporada 2015/2016.
4. El usuario puede pulsar en el menú secundario y acceder a 'Goles esperados xG'. El usuario puede visualizar los resultados del modelo mediante tablas y gráficas. Si el usuario hace click o pone el cursor encima de las tarjetas podrá leer información acerca de las representaciones gráficas.

5. El usuario puede pulsar en el menú secundario y accedera 'Mundial'. El usuario puede visualizar gráficos que representan estadísticas de la final del mundial de 2022 entre Francia y Argentina. El usuario puede elegir el gráfico que quiere visualizar eligiendo el jugador que prefiera. Si el usuario hace click o pone el cursor encima de la tarjeta podrá leer información acerca de las representaciones gráficas.

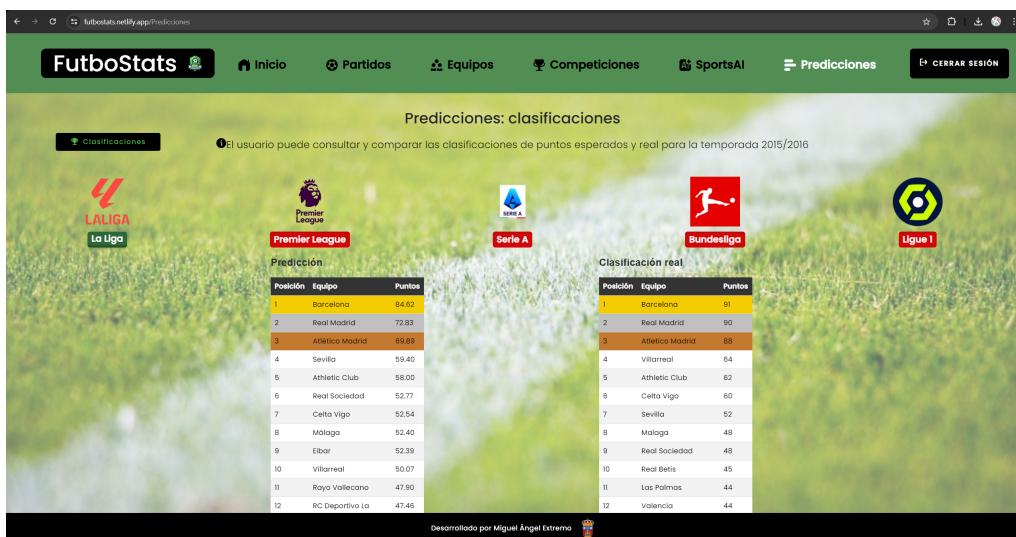


Figura E.14: Visualización de predicción de puntos esperados y real.

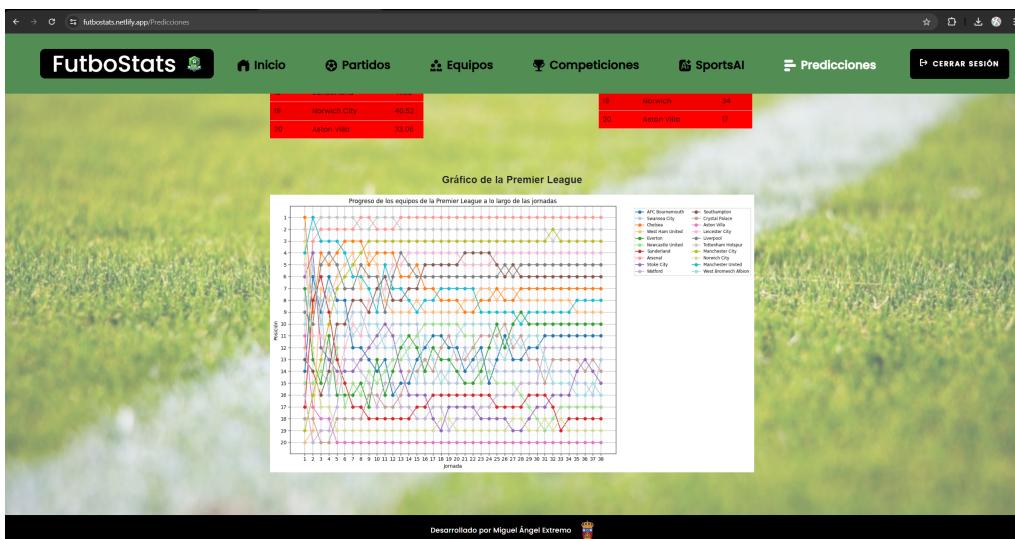


Figura E.15: Representación gráfica de los puntos esperados a lo largo de las jornadas.



Figura E.16: Visualización tablas del modelo de goles esperados xG

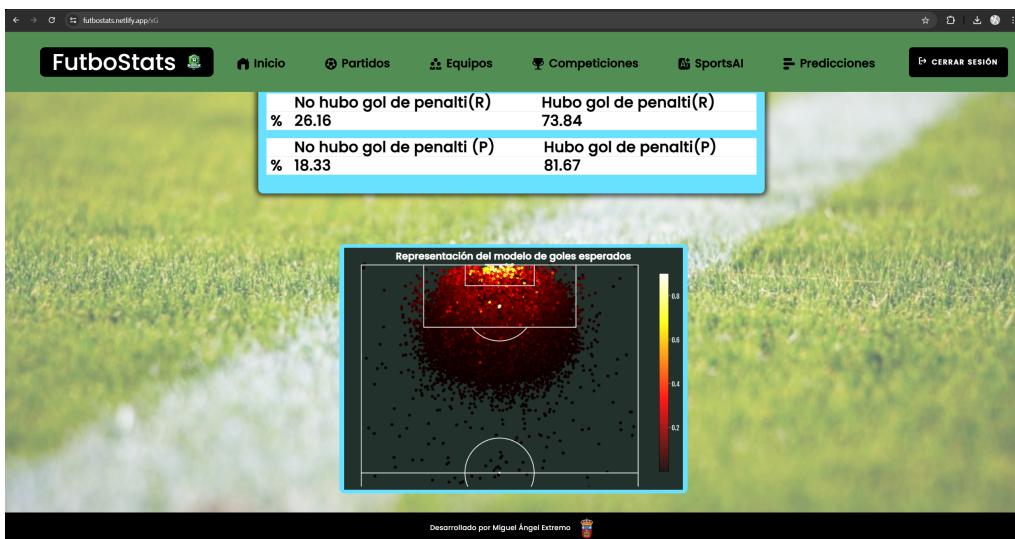


Figura E.17: Representación gráfica del modelo de goles esperados xG.



Figura E.18: Visualización de las estadísticas del mundial 2022

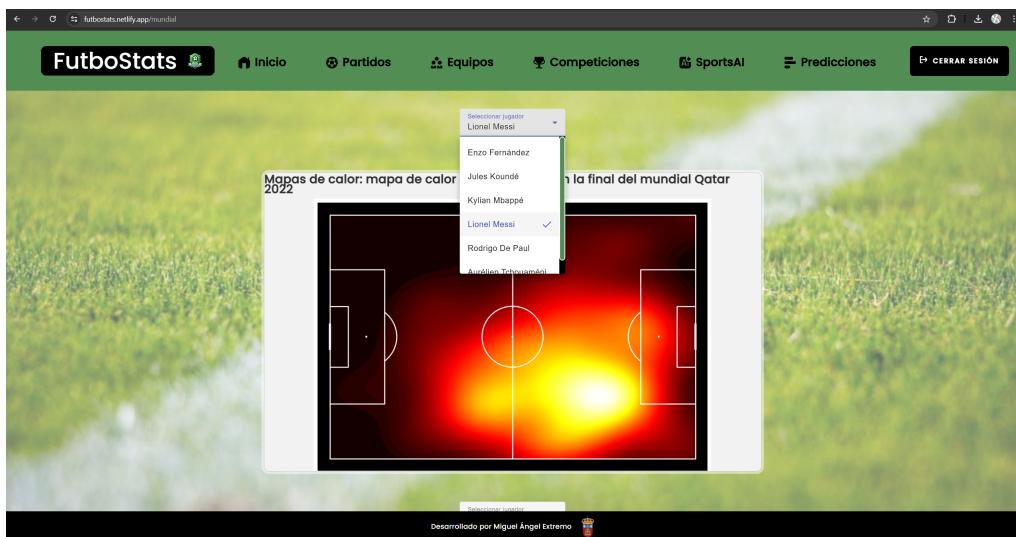


Figura E.19: Seleccionar el jugador para visualizar su gráfico.

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

FutboStats es una aplicación web para la consulta de datos de fútbol, estadísticas y procesado de vídeos. Ha permitido abordar varios aspectos clave de la sostenibilidad. A continuación, se detallan las competencias de sostenibilidad adquiridas y aplicadas en este proyecto, en relación con los objetivos de desarrollo sostenible (ODS).

Salud y bienestar (ODS 3)

FutboStats promueve la actividad física y el bienestar al proporcionar análisis detallados y visualización de datos deportivos. Esta herramienta puede ser utilizada por entrenadores, jugadores y aficionados para mejorar el rendimiento deportivo y fomentar estilos de vida saludables.

Educación de calidad (ODS 4)

Durante el desarrollo de FutboStats, se han adquirido y aplicado conocimientos avanzados en análisis de datos, programación y desarrollo web. Estos conocimientos no solo mejoran las competencias técnicas del alumnado, sino que también pueden ser compartidos y enseñados a otros estudiantes, contribuyendo así a una educación de calidad y al desarrollo profesional. El proceso de desarrollo del proyecto ha involucrado el aprendizaje de lenguajes de programación como Python, TypeScript, HTML, CSS, así como el uso de

bibliotecas y herramientas específicas como StatsBomb e imageAI. Además, la experiencia en el manejo de datos deportivos y la creación de visualizaciones complejas proporciona un conjunto de habilidades aplicables en diversas áreas tecnológicas y científicas. Este conocimiento puede ser transmitido a otros estudiantes ampliando el impacto educativo del proyecto.

Energía Asequible y No Contaminante (ODS 7)

El desarrollo de aplicaciones eficientes en términos de energía y recursos computacionales contribuye a la sostenibilidad ambiental. FutboStats ha sido desarrollado teniendo en cuenta la optimización del uso de recursos, lo cual es un paso hacia la creación de soluciones tecnológicas más sostenibles. La eficiencia energética en el desarrollo de software implica minimizar el uso de recursos computacionales y reducir el consumo de energía en servidores y dispositivos de los usuarios. Esto se logra mediante la optimización del código y la implementación de prácticas de programación sostenible. Al reducir la huella de carbono de las aplicaciones tecnológicas, FutboStats contribuye a la protección del medio ambiente y promueve el uso responsable de la energía.

Conclusiones

En resumen, el desarrollo de FutboStats no solo ha permitido la adquisición de competencias técnicas avanzadas, sino que también ha promovido la reflexión y aplicación de principios de sostenibilidad en diversas áreas. Este proyecto demuestra cómo la tecnología y el análisis de datos pueden contribuir a un desarrollo más sostenible y equitativo, alineado con los objetivos de desarrollo sostenible.

FutboStats, al abordar múltiples ODS, resalta la importancia de integrar la sostenibilidad en el desarrollo tecnológico. La promoción de la salud y el bienestar, la mejora de la educación y la optimización del uso de energía son solo algunas de las formas en que este proyecto contribuye a un futuro más sostenible.

Continuar con el desarrollo de FutboStats permite seguir explorando nuevas formas de integrar la sostenibilidad en el deporte y la tecnología, asegurando que estas prácticas se conviertan en una parte importante de la industria.

Bibliografía

- [1] Angular. Angular. <https://angular.dev/>, 2024. [Internet; accedido 02-junio-2024].
- [2] Creative Commons. Attribution 4.0 international (cc by 4.0). <https://creativecommons.org/licenses/by/4.0/>, 2016. [Internet; accedido 28-mayo-2024].
- [3] Fernán García de Zúñiga. Todo sobre la arquitectura cliente-servidor. <https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor>, 2024. [Internet; accedido 30-mayo-2024].
- [4] Git. Git. <https://git-scm.com/download/win>, 2024. [Internet; accedido 02-junio-2024].
- [5] GitHub. Cloning a repository. <https://docs.github.com/es/repositories/creating-and-managing-repositories/cloning-a-repository>, 2017. [Internet; accedido 02-junio-2024].
- [6] Fernando Herrera. Angular: De cero a experto. <https://www.udemy.com/course/angular-fernando-herrera/>, Última actualización: 3/2024. [Internet; accedido 20-mayo-2024].
- [7] IEEE. Ieee sa - 830-1998 - ieee recommended practice for software requirements specifications. <https://standards.ieee.org/ieee/830/1222/>. [Internet; accedido 29-mayo-2024].
- [8] Microsoft. Visual studio code. <https://code.visualstudio.com/>, 2024. [Internet; accedido 02-junio-2024].
- [9] Node. Node.js. <https://nodejs.org/en/>, 2024. [Internet; accedido 02-junio-2024].

- [10] Moses Olafenwa. Official english documentation for imageai. <https://imageai.readthedocs.io/en/latest/index.html>, 2022. [Internet; accedido 18-mayo-2024].
- [11] open source initiative. The mit license. <https://opensource.org/license/mit>, 2024. [Internet; accedido 28-mayo-2024].
- [12] Python. Python 3.8.0. <https://www.python.org/downloads/release/python-380/>, 2024. [Internet; accedido 02-junio-2024].
- [13] Seguridad Social. Seguridad social. <https://www.seg-social.es/wps/portal/wss/internet/Inicio>, 2024. [Internet; accedido 28-mayo-2024].
- [14] StatsBomb. Statsbomb. <https://statsbomb.com/es/>, 2024. Último acceso: 23 de mayo de 2024.
- [15] CHSO Sunal. Estructura cliente-servidor. <https://chhosunal20181913034worpress.wordpress.com/2018/05/19/estructura-cliente-servidor/>, 2018. [Internet; accedido 30-mayo-2024].
- [16] Wikipedia. Licencia — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Licencia>, 2024. [Internet; accedido 28-mayo-2024].