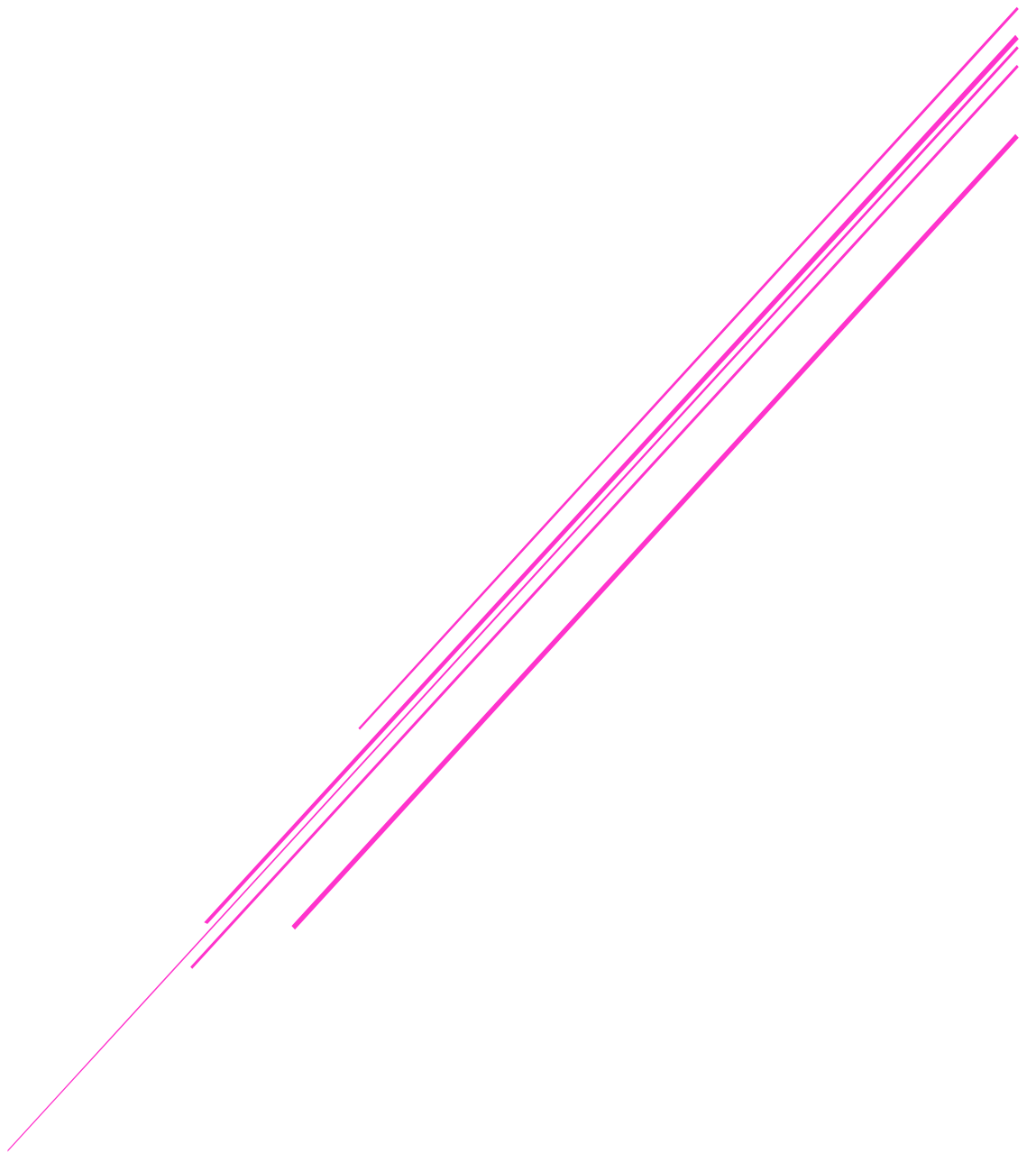


# SPRINT 4 - WISEBET

Análisis de viabilidad & Manual de Usuario



WISEBET  
3º ESI UCLM

## Contenido

1. Análisis de viabilidad.....	2
Desarrollo.....	2
Marketing.....	2
Análisis del Mercado:.....	2
Público Objetivo: .....	3
Estrategias de Marketing: .....	3
Métricas de Éxito: .....	3
Mantenimiento .....	3
Mantenimiento del Sitio Web: .....	3
Actualización de Contenido:.....	4
Atención al Cliente:.....	4
Mantenimiento de la Base de Datos: .....	4
Seguimiento de Métricas de Desempeño: .....	4
Inversión .....	4
Inversión en Marketing:.....	4
Inversión en Mantenimiento:.....	5
2. Manual de usuario .....	5
Soporte.....	11
¿Cómo obtener tus propias API Keys?.....	11
Obtener API Keys para configCuo.json .....	11
Obtener API Keys para configEst.json.....	13
Credenciales para inicio de sesión .....	13

AUTORES	% PARTICIPACIÓN LABORATORIO
Javier Cuartero Corredor	60%
Juan de Dios Carrera Zazo	25 %
Víctor Felipe Jiménez Díaz	15%

## 1. Análisis de viabilidad

### Desarrollo

En este proyecto hemos sido 3 desarrolladores repartidos de esta manera:

Backend: Javier Cuartero Corredor

Testing: Juan de Dios Carrera Zazo

Frontend: Víctor Felipe Jiménez Díaz

De todas formas, al ser un equipo pequeño estos roles han sido orientativos y al final todos los miembros del grupo hemos trabajado en los tres frentes.

Para resolver este proyecto hemos empleado alrededor de 150 horas de trabajo. Durante estas horas hemos desarrollado la aplicación con ayuda de varios lenguajes: HTML, Python, javascript y css.

También hemos utilizado otras tecnologías como Balsamiq Mockups, Flask, Github, Visual Studio Code y Docker.

Como desarrollo para el futuro tenemos pensado desarrollar una aplicación para móvil porque creemos que a día de hoy se utiliza mucho y podría ser otra opción por explotar.

Las APIs que hemos utilizado son: RapidAPI y The Odds-API

### Marketing

Análisis del Mercado:

- **Demanda:** Existe una demanda creciente de servicios que combinen estadísticas deportivas con apuestas, ya que los apostadores buscan maximizar sus posibilidades de ganar dinero.
- **Competencia:** Hay competidores en el mercado que ofrecen servicios similares, pero nuestro enfoque en la rentabilidad de las apuestas puede diferenciarnos de otros.
- **Tendencias:** Las apuestas deportivas en línea están en aumento, especialmente entre los jóvenes adultos, lo que sugiere un mercado en expansión para nuestra plataforma.

#### Público Objetivo:

- **Apostadores Deportivos:** Personas interesadas en las apuestas deportivas, especialmente aquellas que valoran la seguridad y la inteligencia en sus decisiones de apuestas.
- **Entusiastas del Fútbol:** Aficionados al fútbol que buscan información y consejos para mejorar sus apuestas en eventos deportivos.

#### Estrategias de Marketing:

- **Marketing de Contenidos:** Crearemos contenido útil y relevante relacionado con el fútbol, las apuestas y las estadísticas deportivas para atraer a nuestro público objetivo. Esto incluye contenido “vertical” como tiktoks y reels de instagram, y otro tipo de contenido como vídeos informativos y podcasts.
- **Redes Sociales:** Utiliza plataformas como Instagram y Twitter para interactuar con nuestra audiencia y promocionar nuestra plataforma.
- **Colaboraciones:** Buscaremos asociarnos con influencers y expertos en fútbol y apuestas para llegar a una audiencia más amplia y aumentar la credibilidad de nuestra plataforma.
- **Publicidad Online:** Vamos a invertir en anuncios en línea dirigidos a nuestro público objetivo en plataformas como Google Ads y redes sociales para aumentar la visibilidad de WiseBet.

#### Métricas de Éxito:

- **Número de Usuarios Registrados:** Seguimiento del crecimiento de la base de usuarios a lo largo del tiempo.
- **Tasa de Conversión:** Monitorización de la proporción de visitantes del sitio web que se convierten en usuarios registrados.
- **Ingresos Generados:** Seguimiento de los ingresos generados a través de las comisiones de las apuestas realizadas en la plataforma.
- **Feedback del Usuario:** Recopilación y análisis del feedback de los usuarios para realizar mejoras continuas en la plataforma.

## Mantenimiento

#### Mantenimiento del Sitio Web:

- **Actualizaciones Técnicas:** Regularmente actualizaremos el software y los plugins utilizados en el sitio web para garantizar la seguridad y el rendimiento óptimo.
- **Resolución de Problemas:** Implementaremos un sistema para identificar y solucionar rápidamente cualquier problema técnico que pueda surgir en el sitio web.

- **Optimización del Rendimiento:** Realizaremos pruebas periódicas de velocidad y rendimiento para garantizar una experiencia de usuario óptima.
- **Seguridad:** Mantendremos medidas de seguridad actualizadas, como firewalls y sistemas de detección de intrusiones, para proteger los datos de los usuarios y la integridad del sitio web.

#### Actualización de Contenido:

- **Datos Deportivos:** Mantendremos actualizadas las estadísticas y los datos deportivos en la plataforma para proporcionar información precisa a los usuarios.
- **Contenido Informativo:** Actualizaremos regularmente el contenido educativo y de consejos para mantenerlo relevante y útil para los usuarios.

#### Atención al Cliente:

- **Soporte Técnico:** Proporcionaremos un sistema de soporte técnico eficiente para ayudar a los usuarios con cualquier problema o pregunta que puedan tener.
- **Feedback de los Usuarios:** Estableceremos canales para que los usuarios puedan enviar comentarios y sugerencias, y nos aseguraremos de responder y tomar medidas según sea necesario.

#### Mantenimiento de la Base de Datos:

- **Respaldo de Datos:** Implementaremos un plan de respaldo regular para garantizar la seguridad y la integridad de los datos almacenados en la plataforma.
- **Limpieza de Datos:** Realizaremos limpiezas periódicas de la base de datos para eliminar datos obsoletos o innecesarios y mejorar el rendimiento del sistema.

#### Seguimiento de Métricas de Desempeño:

- **Análisis de Usuarios:** Realizaremos un seguimiento regular del comportamiento de los usuarios en el sitio web, incluyendo métricas como la tasa de retención de usuarios y el tiempo de permanencia en el sitio.
- **Feedback de los Usuarios:** Utilizaremos encuestas y análisis de feedback de los usuarios para identificar áreas de mejora y realizar ajustes en la plataforma según sea necesario.

### Inversión

Teniendo en cuenta lo visto anteriormente tanto en marketing como en mantenimiento y teniendo en cuenta que en desarrollo no invertimos porque somos nosotros los desarrolladores hemos estimado estos precios aproximados.

#### Inversión en Marketing:

- **Marketing de Contenidos:** Puedes esperar pagar entre \$500 y \$2000 por pieza de contenido, dependiendo de la complejidad y la calidad requerida. Esto puede incluir artículos de blog, videos o infografías.
- **Redes Sociales:** Los anuncios en redes sociales pueden variar desde unos pocos dólares al día hasta varios cientos de dólares al mes, dependiendo del alcance y la segmentación demográfica deseada. Un presupuesto inicial podría oscilar entre \$500 y \$1000 al mes.

- **Colaboraciones:** El costo de asociarse con influencers y expertos puede variar considerablemente, pero podrías presupuestar entre \$500 y \$3000 por colaboración, dependiendo del tamaño de su audiencia y su influencia.
- **Publicidad Online:** El costo de la publicidad en línea depende del modelo de pago utilizado (PPC, CPM, etc.) y el alcance deseado. Un presupuesto mensual típico podría oscilar entre \$500 y \$2000, dependiendo de la plataforma y la estrategia publicitaria.

#### Inversión en Mantenimiento:

- **Actualizaciones Técnicas:** Contratar a un desarrollador web o una agencia para mantenimiento técnico podría costar entre \$500 y \$2000 al mes, dependiendo del nivel de servicio requerido y la frecuencia de las actualizaciones.
- **Contenido y Soporte:** Contratar personal para crear y mantener contenido, así como para proporcionar soporte técnico y atención al cliente, podría costar entre \$1000 y \$3000 al mes, dependiendo del tamaño del equipo y la carga de trabajo.
- **Seguridad y Respaldo de Datos:** El costo de software de seguridad y servicios de respaldo de datos puede variar. Podrías presupuestar entre \$100 y \$500 al mes para soluciones básicas de seguridad y respaldo.

El coste de las APIs son:

- RapidAPI: 39\$ la versión premium. Para este desarrollo alpha hemos utilizado la básica que es gratis
- The Odds-API: 199\$ la versión más premium. Igual que antes, hemos utilizado la versión gratuita

Para el servidor hemos evaluado que la mejor opción es un servidor dedicado. Los precios pueden variar desde \$80 a \$500 o más por mes, dependiendo de la capacidad y la calidad del hardware.

## 2. Manual de usuario

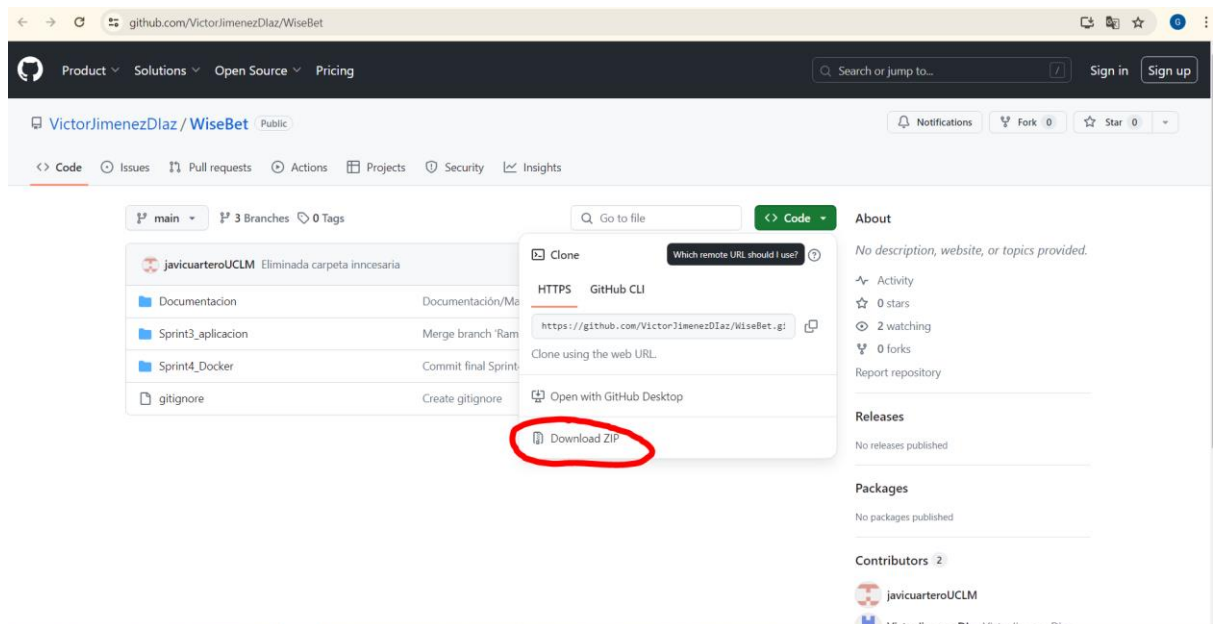
hecho por Javier Cuartero

**\*Antes de comenzar, requisito indispensable:**

-Tener instalado **Python**

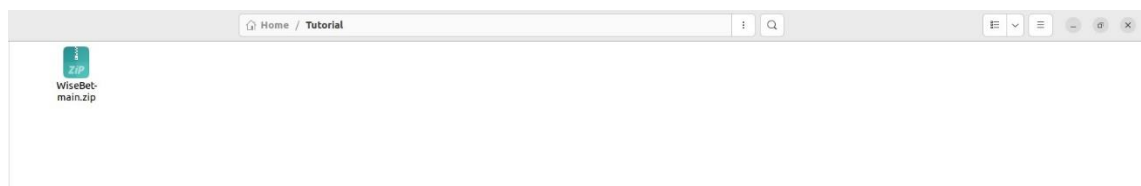
-Tener instalado **Docker**

- 1- Accede <https://github.com/VictorJimenezDlaz/WiseBet> y descárgate el repositorio main.

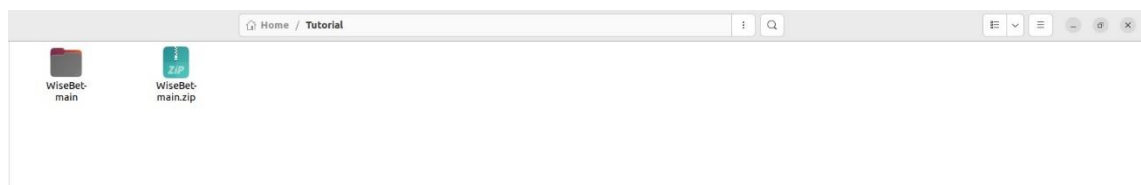


o descarga la carpeta “*Sprint4\_Docker*” que hay en la entrega del Sprint4 en Campus Virtual.

- 2- Una vez descargado guárdalo en la carpeta que tú quieras, en mi caso lo guardo en la carpeta ‘Tutorial’.



- 3- Descomprime el archivo

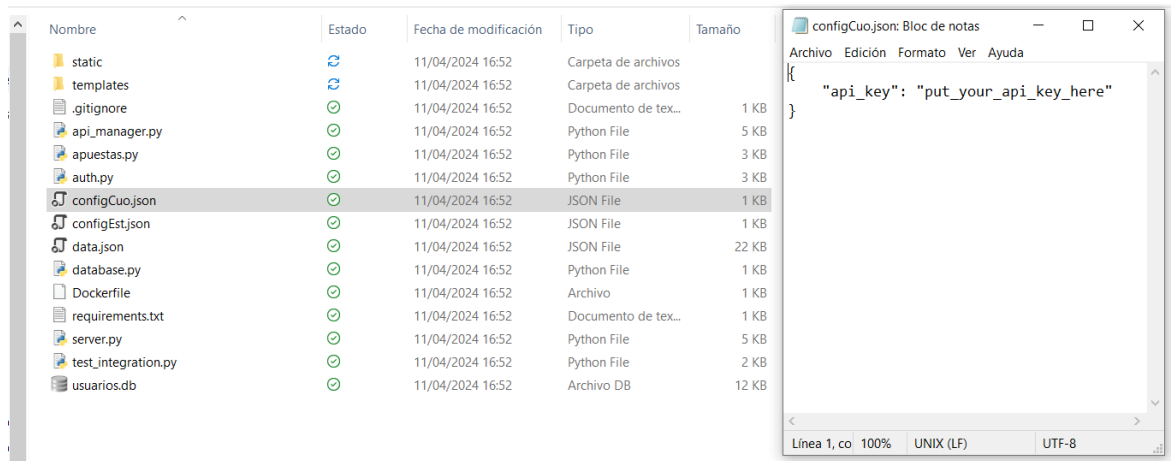


- 4- Entra a la carpeta “*Sprint4\_Docker*”

Documentacion	🔄	11/04/2024 16:52	Carpeta de archivos
Sprint3_aplicacion	🔄	11/04/2024 16:52	Carpeta de archivos
Sprint4_Docker	🔄	11/04/2024 16:52	Carpeta de archivos
gitignore	🔄	11/04/2024 16:52	Archivo

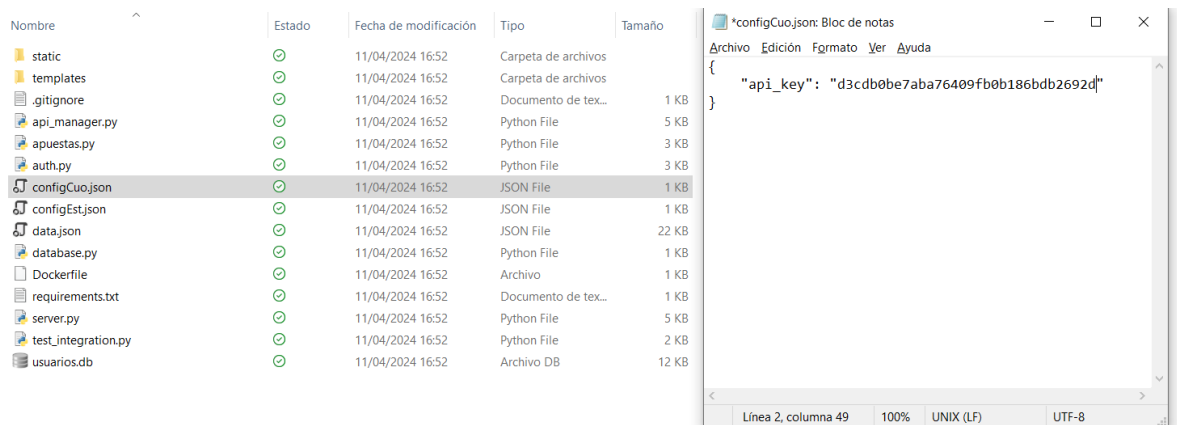
1 KB

- 5- Una vez en la carpeta “*Sprint4\_Docker*”, ve al archivo ‘*configCuo.json*’ y ábrelo con tu editor de texto preferido.



- 6- Una vez abierto, modifica lo que está entre comillas (" ") por lo siguiente:  
**d3cdb0be7aba76409fb0b186bdb2692d**

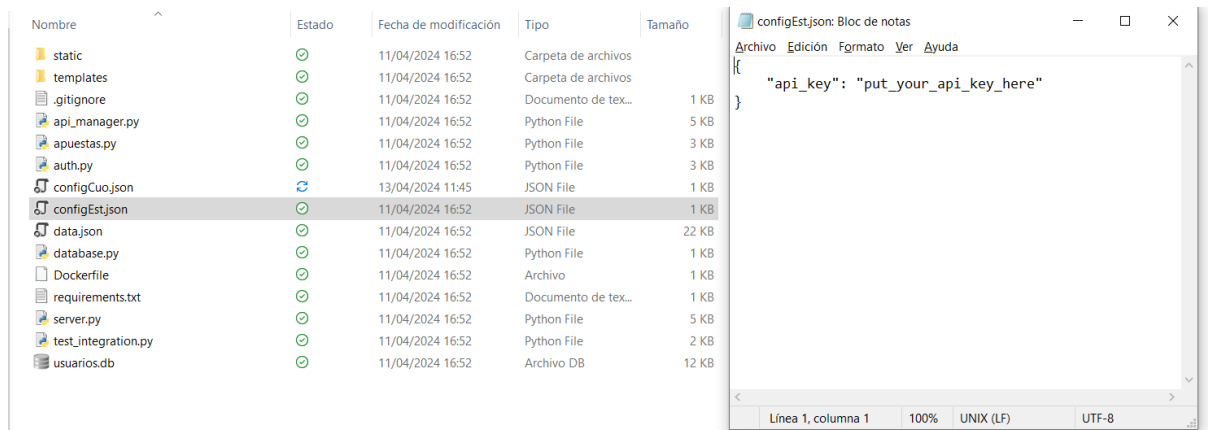
u obtén tu propia API KEY y pégala. Al [final del documento](#) te explico cómo hacerlo.



Y guardamos el archivo.

- 7- Hacemos lo mismo con el archivo `'configEst.json'`

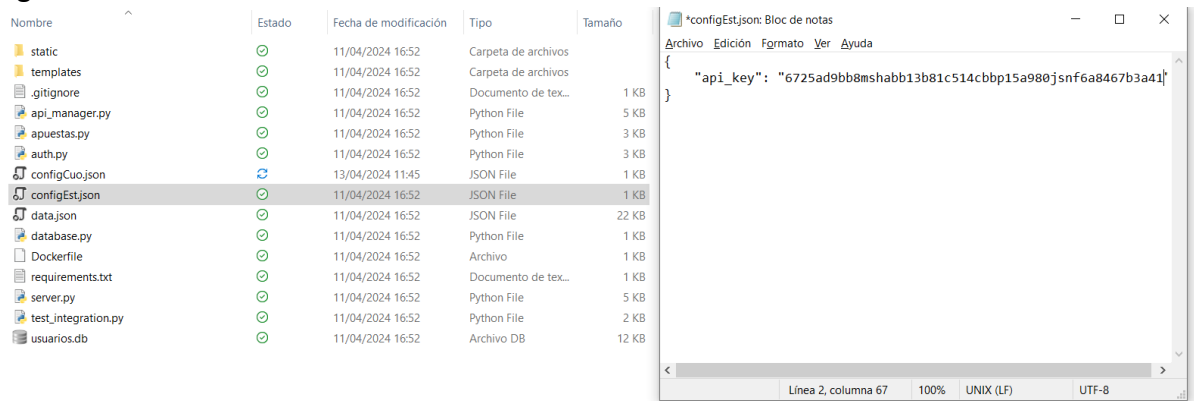




Sustituimos por: **6725ad9bb8mshabb13b81c514cbbp15a980jsnf6a8467b3a41**

u obtén tu propia API KEY y pégala. Al [final del documento](#) te explico cómo hacerlo.

Y guardamos.



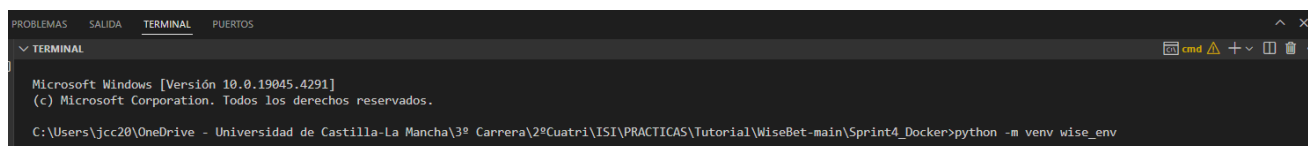
\*Lo de poner las apikeys manualmente lo he hecho así para no tener las apikeys directamente en el repositorio github, que es público.

- Una vez modificados y guardados los archivos, abrimos una terminal en la misma ubicación y creamos un entorno virtual.

Para crear el entorno virtual usaremos el comando:

(EN LINUX) \$ python3 -m venv wise\_env

(EN WINDOWS) \$ python -m venv wise\_env



- Ya creado el entorno virtual, lo cargamos con

(EN LINUX)\$ source wise\_env/bin/activate

(EN WINDOWS) \$ wise\_env\Scripts\activate

```
C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>wise_env\Scripts\activate
(wise_env) C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>
```

10- Construimos la imagen de Docker con el siguiente comando:

\$ docker build -t imagen-wisebet .

```
(wise_env) C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>docker build -t imagen-wisebet .
[*] Building 12.6s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 190B 0.0s
-> [internal] load metadata for docker.io/library/python:3.10 0.5s
-> [internal] load .dockerignore 0.0s
-> => transferring context: 2B 0.0s
-> [1/5] FROM docker.io/library/python:3.10@sha256:f68383667ffe53e85cc0fe4f5a604d303dfa364f238ac37a4675980a2b93b1c5 0.0s
-> [internal] load build context 0.9s
-> => transferring context: 23.11MB 0.8s
-> CACHED [2/5] RUN mkdir /server 0.0s
-> CACHED [3/5] WORKDIR /server 0.0s
-> [4/5] ADD . /server/ 0.3s
-> [5/5] RUN pip install -r requirements.txt 10.3s
-> exporting to image 0.4s
-> => exporting layers 0.4s
-> => writing image sha256:cbcf1c40c90cb6f1091dcb0c24d4d3e9d3d5c0ab617a5b5411c65503b2fd7 0.0s
-> => naming to docker.io/library/imagen-wisebet 0.0s

What's Next?
View a summary of image vulnerabilities and recommendations + docker scout quickview

(wise_env) C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>
```

11- Comprobamos que hemos creado la imagen correctamente con el comando:

\$ docker image ls

```
(wise_env) C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>docker image ls
REPOSITORY      TAG       IMAGE ID       CREATED        SIZE
imagen-wisebet  latest   cbcfc1c40c90   About a minute ago   1.08GB
```

12- Arrancamos la imagen con el comando:

\$ docker run -p 2001:5000 -i imagen-wisebet

```
(wise_env) C:\Users\jcc20\OneDrive - Universidad de Castilla-La Mancha\3º Carrera\2ºCuatri\ISI\PRACTICAS\Tutorial\WiseBet-main\Sprint4_Docker>docker run -p 2001:5000 -i imagen-wisebet
* Serving Flask app 'server'
* 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 all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 154-277-457
```

Ya tenemos el servidor corriendo

13- Desde nuestro navegador accedemos a la dirección <http://127.0.0.1:2001/>

WiseBet  
Apuestas inteligentes, Ganancias seguras

Precios APIS Quiénes somos Iniciar Sesión

Regístrate o Inicia Sesión para acceder a todas las estadísticas y recomendaciones de apuestas

	Equipo	Puntos	DTFC	Resultados	PJ
1		75	46	W W W D W	30
2		67	27	W W W D W	30
3		65	26	W L W L W	30
4		58	21	W L L W D	30
5		56	22	L W W D L	30
6		49	12	W W W L L	30
7		44	1	W D L W D	30
8		42	0	L L L L W	30
9		39	-7	W L L W D	30
10		38	-5	L W W W W	30
11		38	-6	L W L D L	30
12		37	-4	L L L D D	30
13		32	-10	L L W L D	30
14		31	-7	W L D W L	30
15		31	-10	D W L W D	30
16		30	-13	D W L D L	30

Fecha	Equipo Local	Equipo Visitante	Estadio
2024-04-12 21:00:00	Real Betis	Celta Vigo	Estadio Benito Villamarín
2024-04-13 14:00:00	Atletico Madrid	Girona	Estadio Cívitas Metropolitano
2024-04-13 16:15:00	Rayo Vallecano	Getafe	Estadio de Vallecas
2024-04-13 18:30:00	Mallorca	Real Madrid	Estadi Mallorca Son Moix
2024-04-13 21:00:00	Cádiz	Barcelona	Estadio Nuevo Mirandilla
2024-04-14 14:00:00	Las Palmas	Sevilla	Estadio de Gran Canaria
2024-04-14 16:15:00	Granada CF	Alaves	Estadio Nuevo Los Cármenes
2024-04-14 18:30:00	Athletic Club	Villarreal	San Mamés Barria
2024-04-14 21:00:00	Real Sociedad	Almería	Reale Arena
2024-04-15 21:00:00	Osasuna	Valencia	Estadio El Sadar

14- Actualizamos los datos de las APIs accediendo a la siguiente dirección:  
<http://127.0.0.1:2001/update-data>

127.0.0.1:2001/update-data

Dar formato al texto

```
{  
  "message": "Datos actualizados y guardados en data.json"  
}
```

Esto es para actualizar los datos del api en el json de los datos, esto lo he hecho porque teníamos un número muy limitado de peticiones a una de las apis, y lo he hecho de tal forma que cada vez que cargue esta url se actualizan los datos y se escriben en un json, luego el código html lee los datos desde este json y no desde la api directamente. Lo he hecho así porque cada vez que recargaba la página principal para probar algo gastaba peticiones y de esta forma he conseguido reducir el número de peticiones.

15- Volvemos a <http://127.0.0.1:2001/> e iniciamos sesión.

Para iniciar sesión ver anexo [credenciales](#) al final del documento.

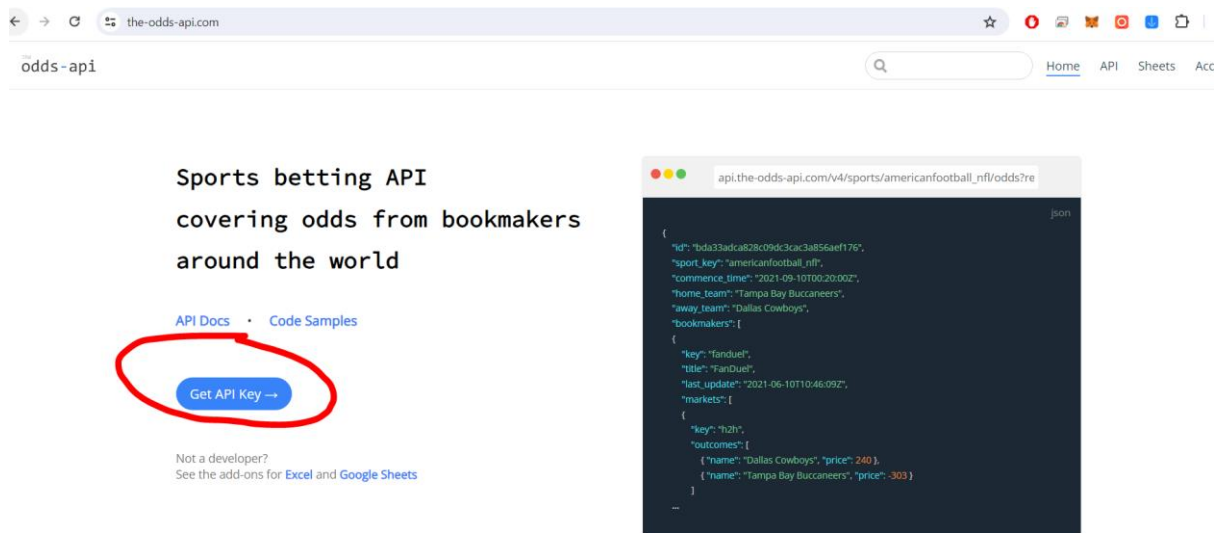
#### Soporte

Para cualquier problema o error, póngase en contacto con [Javier.Cuartero@alu.uclm.es](mailto:Javier.Cuartero@alu.uclm.es)

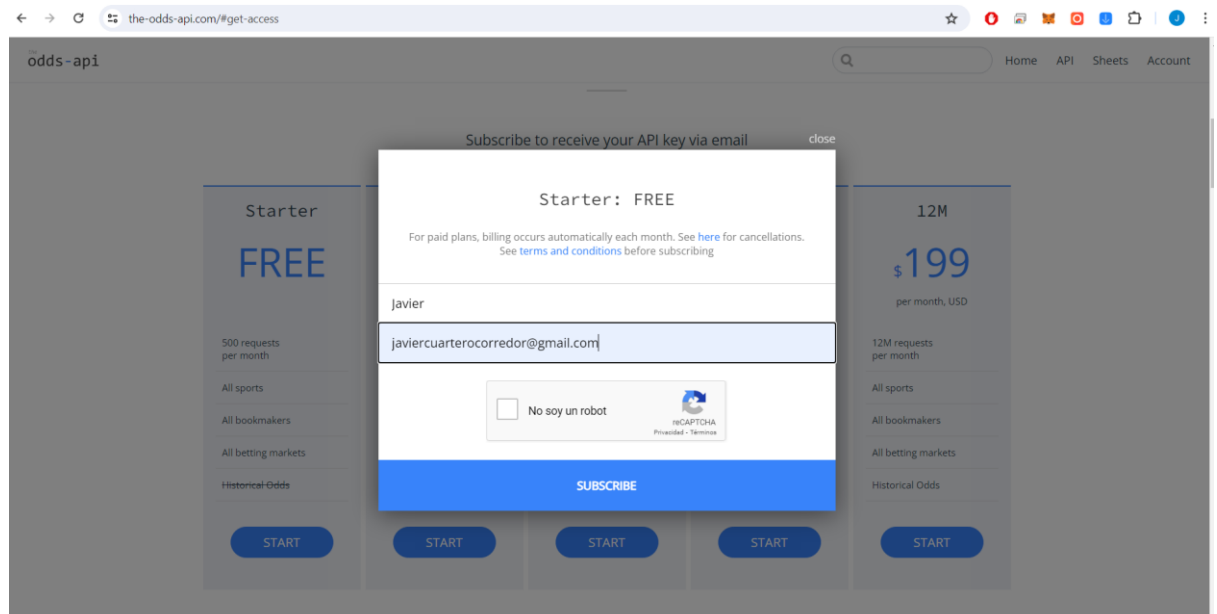
## ¿Cómo obtener tus propias API Keys?

Obtener API Keys para configCuo.json

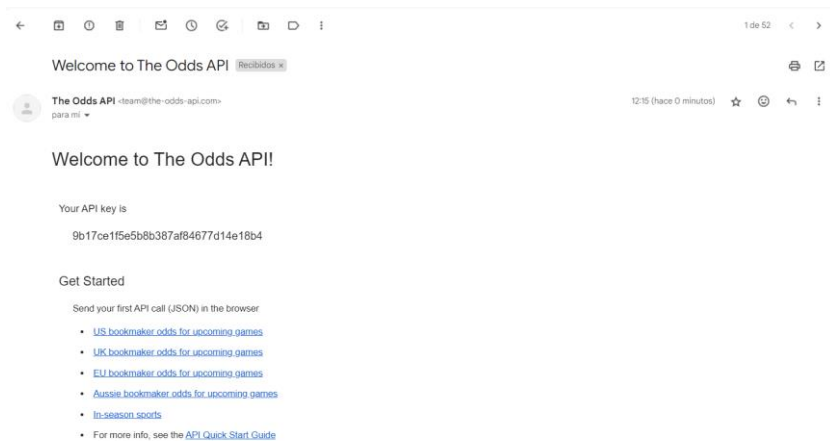
- 1- Accedemos a <https://the-odds-api.com/>
- 2- Clickamos en “Get API Key”



- 3- Seleccionamos el plan “Starter” (plan gratuito) y completamos con la información que nos pide.



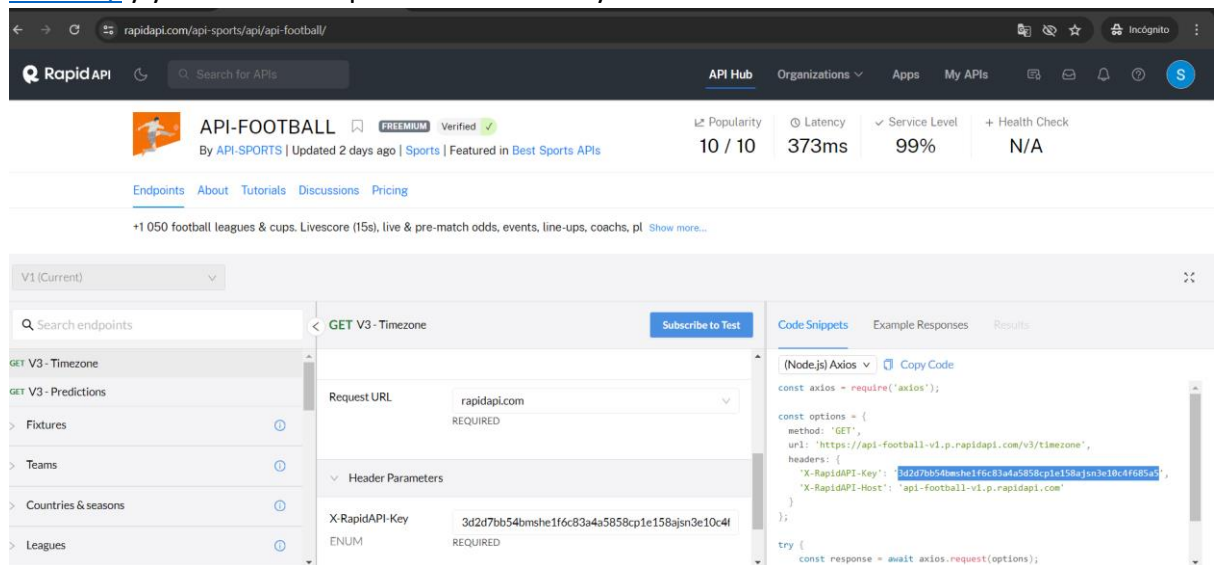
- 4- Nos habrán enviado la API Key al correo electrónico



5- Volvemos al [paso 6](#) y pegamos la API Keys que nos han enviado

## Obtener API Keys para configEst.json

- 1- Accedemos a <https://rapidapi.com/api-sports/api/api-football/>
- 2- Nos creamos una cuenta pulsando el botón azul “SIGN UP” arriba a la derecha.
- 3- Una vez creada la cuenta volvemos a <https://rapidapi.com/api-sports/api/api-football/> y ya deberíamos poder ver la API Key



4- Volvemos al [paso 7](#) y pegamos la API Key obtenida

**\*IMPORTANTE\***-> Si la API Key obtenida no funciona porque requiere introducir una tarjeta de crédito/débito para activarla, recomendamos usar nuestra API Key.  
**Nuestra API Key: 6725ad9bb8mshabb13b81c514cbbp15a980jsnf6a8467b3a41**

## Credenciales para inicio de sesión

-usuario: [felix@gmail.com](mailto:felix@gmail.com)

-contraseña: felix

Otras credenciales para probar los distintos planes:

-usuario: [premium@gmail.com](mailto:premium@gmail.com)

-contraseña: premium

-usuario: [pro@gmail.com](mailto:pro@gmail.com)

-contraseña: pro

También tiene la opción de crear una cuenta nueva.

### **DIFERENCIA DE LOS DISTINTOS TIPOS DE USUARIO**

El usuario Félix tiene activado el **plan gratis**, esto significa que se le muestra sólo el 30% de todas las apuestas recomendables.

El usuario Premium tiene activado el **plan premium**, esto significa que se le muestra sólo el 50% de todas las apuestas recomendables.

El usuario Pro tiene activado el **plan pro**, esto significa que se le muestran todas las apuestas recomendables.