



# FutApi

## Proyecto Integrado

Realizado por: Víctor Fernández España y Javier Postigo Arévalo

# ÍNDICE:

1. Introducción
2. Objetivos y requisitos
3. Antecedentes
4. Diseño
  - a. Backend
  - b. Frontend
5. Pruebas
6. Conclusiones



# 1. Introducción

# 1. Introducción

## ¿Cuál es el objetivo principal?

- Elaborar predicciones aplicadas al fútbol, enfocándose en el rendimiento de los jugadores a lo largo de una temporada. Y ofrecer una interfaz de usuario amigable y eficiente por la que cualquier persona pueda probarla.

## Estructura del Backend

- Entrenamiento de modelos de aprendizaje
- API para proporcionar el servicio
- Desarrollado en **Python** usando **Scikit-learn** y **Flask**

## ¿De dónde nace la inspiración para hacer este P.I.?

- De las charlas impartidas por Miguel Ángel Arroyo durante la Sysmana del instituto, donde se utilizaban modelos de aprendizaje para predecir datos sobre partidos y equipos de fútbol.

## Estructura del Frontend

- Interfaz gráfica de usuario en formato web
- Desarrollado con **Angular**, usando **HTML**, **CSS** y **TypeScript**

## 2. Objetivos y Requisitos.

## 2. Objetivos y Requisitos. OBJETIVOS

### Objetivo Principal

Desarrollar predicciones de varios aspectos del juego y datos relevantes, además de desarrollar una web interactiva.

### Propósito del Proyecto

Adquirir conocimientos y experiencia práctica en el uso de modelos de aprendizaje mediante Python y sus bibliotecas especializadas en machine learning.

### Objetivos Específicos

- Predecir goles, asistencias, regates completados, acciones de creación de gol, posición de un jugador, y jugadores similares a uno.
- Desplegar nuestra API utilizando Flask.
- Desarrollar y desplegar nuestra web interactiva usando Angular.
- Mostrar procesos de creación de modelos de inteligencia artificial.

### Aplicación práctica de técnicas de Machine Learning

- Implementar y entender modelos predictivos aplicados a datos reales.
- Evaluar la precisión y efectividad de diferentes algoritmos de predicción.

## 2. Objetivos y Requisitos. REQUISITOS

### Conocimientos necesarios sobre:

- Python
- Flask
- Angular
- Machine Learning
- Despliegue de una aplicación
- Manejo de datos

### Capacidades Técnicas

- Gestión de datos de jugadores.
- Creación de una interfaz de usuario intuitiva.
- Diseños web mobile first, adaptados a ordenador y tableta.

# 3. Antecedentes



# 3. Antecedentes.

## Origen del Proyecto

El proyecto "FutApi" se originó a partir de las charlas impartidas por Miguel Ángel Arroyo, tutor de FCT de Javier Postigo Arévalo, durante la Sysmana organizada por el IES Gran Capitán. Estas charlas presentaban el uso de modelos de aprendizaje para realizar predicciones y proporcionar datos sobre partidos y equipos de fútbol.

## Propósito del Proyecto

- Adquirir conocimientos y experiencia práctica en el uso de modelos de aprendizaje mediante Python y sus bibliotecas especializadas en machine learning.

## Inspiración

- **Charlas de Sysmana:** Inspiración principal provino de las presentaciones de Miguel Ángel Arroyo, donde se demostraban aplicaciones prácticas de machine learning en el análisis deportivo.
- **Interés en el Fútbol:** La pasión por el fútbol y el análisis de rendimiento deportivo impulsaron la idea de aplicar técnicas de machine learning para predecir y evaluar el desempeño de los jugadores.

# 3. Antecedentes.

## Contexto Educativo

El proyecto se enmarca dentro del ciclo superior de Desarrollo de Aplicaciones (DAW) del IES Gran Capitán, como parte del Proyecto Integrado. Este tipo de proyectos integrados permiten a los alumnos aplicar sus conocimientos teóricos en un contexto práctico y real, desarrollando habilidades en:

Programación y Desarrollo Web: Usando tecnologías como Python, Flask, Angular, HTML, CSS y TypeScript.

Machine Learning: Implementación de modelos predictivos utilizando bibliotecas especializadas como Scikit-learn.

Trabajo en Equipo y Gestión de Proyectos: Colaboración entre compañeros y manejo de todas las fases del desarrollo de una aplicación completa.

## Estado Actual y Futuro del Proyecto

El proyecto FutApi es una herramienta en desarrollo que busca ser una referencia en la predicción y evaluación del rendimiento de jugadores de fútbol. A medida que avance, se espera integrar más datos y mejorar la precisión de los modelos predictivos, así como optimizar la interfaz de usuario para una mejor experiencia.

# 4. Diseño

# 4.1. Backend

## 4.1. Backend

### Flask, la opción elegida.

Flask es una opción popular para desarrollar APIs en Python, especialmente cuando se trata de proyectos que implican el uso de modelos guardados en formato .pkl (pickle). Existen otros frameworks como Django que también es una opción robusta y ampliamente utilizada, pero Flask ofrece ciertas ventajas que pueden hacerlo más adecuado para este tipo de proyectos.



#### Manejo de Modelos y Serialización:

- **Flask:** Fácil integración de modelos .pkl.
- **Django:** Puede manejar modelos .pkl, pero su configuración es más complicada.

#### Desempeño:

- **Flask:** Mejor desempeño para servicios específicos y ligeros.
- **Django:** Eficiente, pero su estructura compleja puede afectar el rendimiento en APIs simples.

#### Simplicidad y Flexibilidad:

- **Flask:** Microframework minimalista, ideal para proyectos pequeños.
- **Django:** Framework completo con muchas funcionalidades integradas, lo que puede ser excesivo para nuestra API.

## 4.1. Backend - Databases

- Se ha utilizado la database proveniente de Kaggle de [2022-2023 Football Player Stats de Vivo Vinco.](#)
- Esta database consta de un archivo csv donde se alojan nuestros datos.
- Para el entrenamiento y testeo de nuestros modelos de aprendizaje, se ha separado el database original en dos conjuntos, uno de entrenamiento y otro de prueba para el posterior procesamiento de los datos. Usando 80% para entrenamiento y 20% para testing.

# 4.1. Backend - Modelos aprendizaje

## ● Funciones y librerías.

- Para el desarrollo de la API, se ha utilizado las librerías: *Pandas*, *Numpy*, *Scikit-learn*, *seaborn*, *matplotlib*, *Joblib* y *Pickle*. Estas se han usado para el manejo de datos y la implementación de modelos de aprendizaje automático.

## ● Selección de características.

- La librería **seaborn**, se ha podido generar un mapa de calor con las puntuaciones de correlación entre características.
- La *correlación* entre características indica cómo dos **variables** se relacionan entre sí; valores cercanos a 1 significan una relación positiva, mientras que cercanos a -1, una relación negativa.



## 4.1. Backend - Ran. Forest Regressor

- **Qué es y cómo funciona**
  - Es un algoritmo de aprendizaje supervisado usado para regresión. Pertenece a la familia de métodos de ensemble, que combinan múltiples modelos para mejorar la precisión predictiva.
  - Construye múltiples árboles de decisión de forma independiente, utiliza la combinación de las predicciones de estos árboles para hacer una predicción final y reduce el sobreajuste mediante la aleatorización en el proceso de construcción de árboles.
- **¿Qué modelos se han generado con Random Forest Regressor y por qué se ha utilizado?**

Se ha utilizado Random Forest Regressor en estas características debido a que el método nos proporcionaba resultados significativamente superiores en comparación con otros métodos disponibles.

- **Goals** (Goles): porcentaje de acierto del 94% y un error cuadrático de 0.22.
  - **Assists** (Asistencias): porcentaje de acierto del 92% y un error cuadrático de 0,11.
- **Justificación parámetros de los métodos seleccionados.**
  - Se han utilizado los parámetros (n\_estimators=100, criterion='gini', max\_depth=none, min\_samples\_split=2, min\_samples\_leaf=1, max\_features='auto', bootstrap=true) ya que han sido los que mejor resultado nos han dado.



## 4.1. Backend - Linear Regression

- **Qué es y cómo funciona**

- Es un método estadístico utilizado para modelar la relación entre una variable dependiente (objetivo) y una o más variables independientes (predictores).
- La regresión lineal ajusta una línea recta (o un hiperplano en dimensiones superiores) que mejor se ajusta a los datos, y luego utiliza esta línea para hacer predicciones sobre nuevos datos basados en las relaciones lineales entre las variables.

- **¿Qué modelos se han generado con Linear Regression y por qué se ha utilizado?**

Se ha utilizado la regresión lineal en estas características debido a que el método nos proporcionaba resultados significativamente superiores en comparación con otros métodos disponibles.

- **ACG** (Acciones de creación de gol): porcentaje de acierto del 99% y un error cuadrático de  $2,22 \cdot 10^{-5}$ .
- **TKI** (Entradas realizadas): porcentaje de acierto del 99% y un error cuadrático de  $2,22 \cdot 10^{-8}$ .
- **ToSuc** (Regates exitosos): porcentaje de acierto del 85% y un error cuadrático de 0,11.

- **Justificación parámetros de los métodos seleccionados.**

- Se han utilizado los parámetros: (fit\_intercept=True, normalize='deprecated', copy\_X=True, n\_jobs=None, positive=False) ya que han sido los que mejor resultado nos han dado.

## 4.1. Backend - Rand. Forest Classifier

- **Qué es y cómo funciona**

- Es un algoritmo de aprendizaje supervisado utilizado para clasificación.
- Construye múltiples árboles de decisión de forma independiente, utiliza la votación por mayoría para predecir la clase final y reduce el sobreajuste mediante la aleatorización en el proceso de construcción de árboles.

- **¿Qué modelos se han generado con Rand. Forest Classifier y por qué se ha utilizado?**

Position (Posición del jugador): porcentaje de acierto del 91%

Se ha utilizado Random Forest Classifier en estas características debido a que el método nos proporcionaba resultados significativamente superiores en comparación con otros métodos disponibles.

- **Justificación parámetros de los métodos seleccionados.**

- Se han utilizado los parámetros (n\_estimators=100, criterion='gini', max\_depth=none, min\_samples\_split=2, min\_samples\_leaf=1, max\_features='auto', bootstrap=true) ya que han sido los que mejor resultado nos han dado.

## 4.1. Backend - MLP Classifier

- **Qué es y cómo funciona**
  - Es un clasificador basado en redes neuronales artificiales, específicamente en perceptrones multicapa utilizado en aprendizaje supervisado para problemas de clasificación.
  - Funciona como un clasificador basado en redes neuronales artificiales que aprende a mapear características de entrada a etiquetas de salida mediante el ajuste de pesos de conexiones entre neuronas.

- **¿Qué modelos se han generado con MLP Classifier y por qué se ha utilizado?**

Position (Posición del jugador): porcentaje de acierto del 85%.

Se ha utilizado MLP Classifier en estas características debido a que el método nos proporcionaba resultados significativamente superiores en comparación con otros métodos disponibles.

- **Justificación parámetros de los métodos seleccionados.**

- Se han utilizado los parámetros (hidden\_layer\_sizes=(100, 50), activation='relu', solver='adam', max\_iter=500, random\_state=42) ya que han sido los que mejor resultado nos han dado.

## 4.1. Backend - Kmeans

- **Qué es y cómo funciona**

- Es un algoritmo de agrupamiento (clustering) utilizado en aprendizaje no supervisado para dividir un conjunto de datos en grupos o clústeres.
- KMeans busca minimizar la distancia intra-cluster y maximizar la distancia inter-cluster, lo que significa que los puntos dentro de un mismo clúster son similares entre sí y diferentes de los puntos en otros clústeres.

- **¿Qué modelos se han generado con Kmeans y por qué se ha utilizado?**

Jugadores similares: porcentaje de acierto del 86%.

Se ha utilizado Kmeans en estas características debido a que el método nos proporcionaba resultados significativamente superiores en comparación con otros métodos disponibles y además por ser datos linealmente separables.

- **Justificación parámetros de los métodos seleccionados.**

- Se han utilizado los parámetros (n\_clusters=5, init=k\_means++, n\_init=10, max\_iter=300, tol=1e-4, random\_state=none) ya que han sido los que mejor resultado nos han dado.

## 4.1. Backend - Despliegue



El I.E.S. Gran Capitán ha proporcionado una M.V. Ubuntu Server 22.04 LTS con 2 núcleos, 4 GB de RAM y 20 GB de espacio. Esta plataforma proporciona un entorno estable y seguro para desplegar la API Futapi.

Para el despliegue de la API, hemos utilizado una máquina virtual proporcionada por el instituto. Sin embargo, debido a problemas con brechas de seguridad, no ha sido posible abrir el puerto 5000 para acceso externo. Para superar esta limitación y permitir que la API esté accesible desde el exterior, hemos utilizado Ngrok.

**Ngrok** es una herramienta que ofrece un túnel seguro para exponer servicios web locales a través de Internet de manera temporal. Este servicio facilita pruebas y demostraciones al generar una URL pública que redirige las solicitudes a nuestra aplicación local.

- Seguridad: Al no abrir puertos directamente en la máquina virtual, se minimizan las posibles brechas de seguridad.
- Facilidad de Uso: Ngrok simplifica la exposición de servicios locales, eliminando la necesidad de configuraciones complejas de red.

## 4.2. Frontend

## 4.2. Frontend

### Angular, la opción elegida

Angular es un framework de JavaScript para desarrollar aplicaciones web dinámicas y robustas. Facilita la creación, mantenimiento y escalabilidad del código, además, es el framework aprendido y con el que ha trabajado Víctor Fernández España durante su FCT en atmira.

### ¿Por qué se ha usado Vercel y no el servidor del instituto?

Se ha decidido utilizar *Vercel* para el despliegue front. *Vercel* es una plataforma para desplegar aplicaciones web de manera rápida y eficiente.

### ¿Por qué se ha usado una app de terceros para el despliegue?

- Porque ofrece un despliegue continuo ya que se sincroniza con el repositorio de Github.

## 4.2. Frontend

### ¿Cómo es este Framework?

Angular está organizado en varias partes fundamentales que permiten una estructura clara y modular. Los componentes son las piezas básicas de una aplicación Angular y controlan una vista específica de la interfaz de usuario. Cada componente está compuesto por un archivo TypeScript para la lógica, un archivo HTML para la plantilla y un archivo CSS para los estilos.

### ¿Qué hemos utilizado realmente ?

- **Componentes:** Controlan partes específicas de la interfaz de usuario.
- **Servicios:** Conectan con la API, realizan la petición.
- **Módulos:** Organizan y agrupan componentes, servicios y otras funcionalidades.
- **Routing Module:** Gestiona la navegación y las rutas de la aplicación.
- **Interfaces:** Definen la estructura de datos y tipos, facilitando la tipificación fuerte y la coherencia en el manejo de datos.



# 5. Pruebas

## 5. Pruebas

https://ibex-massive-reindeer.ngrok-free.app/getPlayers

GET https://ibex-massive-reindeer.ngrok-free.app/getPlayers

Params Authorization Headers (8) Body ● Scripts Settings

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "Rk": 1341,
4     "Player": "Lukas Kübler",
5     "Nation": "GER",
6     "Pos": 1,
7     "Squad": "Freiburg",
8     "Comp": "Bundesliga",
9     "Age": 30,
10    "Born": 1992,
11    "MP": 9,
12    "Starts": 9,
13    "Min": 778,
14    "90s": 8.6,
15    "Goals": 2,
16    "Shots": 0.58,
17    "SoT": 0.35,
18    "SoT%": 60.0,
19    "G/Sh": 0.4,
20    "G/SoT": 0.67,
21    "ShoDist": 17.9
```

https://ibex-massive-reindeer.ngrok-free.app/getPlayerByRk/808

GET https://ibex-massive-reindeer.ngrok-free.app/getPlayerByRk/808

Params Authorization Headers (8) Body ● Scripts Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize HTML

```
1 [{"Rk":808,"Player":"Abdessamad Ezzalzouli","Nation":"MAR","Pos":3,"Squad":"Osasuna","Comp":"La
62,"SoT%":28.6,"G/Sh":0.0,"G/SoT":0.0,"ShoDist":17.9,"ShoFK":0.15,"ShoPK":0.0,"PKAtt":0.0
"PasShoCmp":14.5,"PasShoAtt":18.3,"PasShoCmp%":79.0,"PasMedCmp":6.31,"PasMedAtt":9.23,"PasM
"Pas3rd":0.62,"PPA":2.0,"CrsPA":0.62,"PasProg":2.62,"PasAtt":33.7,"PasLive":32.0,"PasDead":
"CkStr":0.0,"PasCmp":22.9,"PasOff":0.15,"PasBlocks":2.62,"SCA":3.08,"ScaPassLive":1.85,"Sca
46,"GcaPassDead":0.0,"GcaDrib":0.0,"GcaSh":0.0,"GcaFld":0.0,"GcaDef":0.0,"Tkl":2.31,"TklWon
"TklDri%":66.7,"TklDriPast":0.46,"Blocks":1.85,"BlkSh":0.0,"BlkPass":1.85,"Int":0.31,"Tkl+I
"TouAtt3rd":30.2,"TouAttPen":3.54,"TouLive":51.4,"ToAtt":5.69,"ToSuc":2.15,"ToSuc%":37.8,"T
"Car3rd":3.23,"CPA":2.62,"CarMis":3.69,"CarDis":3.85,"Rec":36.3,"RecProg":9.08,"CrdY":0.46,
"PKcon":0.0,"OG":0.0,"Recov":3.85,"AerWon":0.15,"AerLost":0.31,"AerWon%":33.3,"Cluster":4}]
```

## 5. Pruebas

Puedes seleccionar a un jugador como base de las estadísticas:

Aaron Cresswell (West Ham)  
Aaron Hickey (Sheff Wed)  
Aaron Ramsdale (Arsenal)  
Aaron Wan-Bissaka (Manchester Utd)  
**Ademola Lookman (Sheff Wed)**  
Adrian Murrells (Sheff Wed)  
Adrian Prati (Mallorca)  
Abdoulaye Doucoure (Sheff Wed)  
Abdul Mumin (Bayo Velazquez)  
Adam Buxa (Lens)  
Adam Lallana (Brighton)  
Adam Marshall (Lazio)  
Adam Ounas (Lille)  
Adam Webster (Brighton)  
Adama Soumaro (Bologna)  
Adil Aouchiche (Lorient)  
Adri Embarba (Espanyol)  
Adrian Thompson (Strasbourg)  
Agustín Marchesini (Celta Vigo)  
Ainsley Maitland-Niles (Southampton)  
Aitor Fernández (Osasuna)

Precisión de los tiros a puerta  
Goles/Tiro a puerta  
Lanzamientos de falta  
Penalís lanzados  
Pases totales completados  
Pases totales intentados  
Precisión de los pases  
Distancia recorrida por los pases  
Pases que acaban en tiro completados  
Pases que acaban en tiro intentados  
Precisión de pases que llevan a tiro  
Pases completados en 1/2 campo

Puedes seleccionar a un jugador como base de las estadísticas:

Abdessamad Ezzaoui (Ossana)

Seleccionar Predecir

## Rellena cada campo

Posición	Delantero	Edad	21
Nacimiento	2005	Partidos jugados	14
Partidos jugados de inicio	9	Minutos jugados	388
Minutos jugados/partido	6,5	Tiros	2,11
Tiros a puerta	9,6	Precisión de los tiros a puerta	28%
Goles/Tiro	9	Goles/Tiro a puerta	6
Distancia de tiro (yardas)	171	Lanzamientos de falta	6,51
Penalis marcados	9	Penalis lanzados	9
Pases totales completados	221	Pases totales intentados	332
Precisión de los pases	68	Distancia recorrida por los pases	217
Distancia de los pases progresivos	890	Pases que acaban en tiro completados	142
Pases que acaban en tiro intentados	182	Precisión de pases que llevan a tiro	79
Pases completados en 1/2 campo	6,5	Pases intentados en 1/2 campo	9,2
Precisión de pase en 1/2 campo	68	Pases largos completados	1,39
Pases largos intentados	2,3	Precisión de pase largo	60
Asistencias	0,9	Pases que acaban en asistencia	6,7

**Volver**

**Goles esperados: 2.529 goles**



# 6. Conclusiones

## 6. Conclusiones

### Backend

- **Framework:** Flask (Python)
- **Función:** Crear una API para gestionar solicitudes de predicción
- **Modelos** de Machine Learning:
  - a. Regresión lineal
  - b. Clasificación
  - c. Clustering
- **Bibliotecas:** Scikit-learn, pandas, numpy
- **Datos:** Datasets de Kaggle
- **Salidas:** Resultados en formato JSON

### Frontend

- **Framework:** Angular
- **Función:** Interfaz de usuario interactiva y responsive
- **Comunicación:** Consume API de Flask
- **Datos:** Recibe datos en formato JSON
- **Experiencia de Usuario:** Optimizada para móviles y ordenadores
- **Visualización:** Presenta predicciones de manera clara e intuitiva

## 6. Conclusiones

Hemos dado un enfoque integral que demuestra la aplicabilidad de técnicas de machine learning en el análisis deportivo.

La combinación de herramientas de análisis de datos, desarrollo web y machine learning crea una solución integral que tiene el potencial de ser utilizada en diversos contextos dentro del mundo del deporte. La experiencia obtenida a través de este proceso es inestimable para quienes buscan especializarse en el análisis de datos deportivos y el desarrollo de tecnologías avanzadas para mejorar el rendimiento en el fútbol.

**MUCHAS  
GRACIAS POR SU  
ATENCIÓN**



**FUTAPI**

Realizado por: Víctor Fernández España y Javier Postigo Arévalo