



ExpectedFoot

Powered by AI

Realizado por:

Alberto Moreno González
Francisco López González
Ciro Leon Espinosa Aviles

INDICE

01 Obtención de Datos

02 Exploración y
Visualización de los
Datos

03 Limpieza de Datos

04 Preparación de Datos

05 Entrenamiento del modelo

06 Procesamiento de Lenguaje Natural

07 Aplicación Web

08 Conclusión



ExpectedFoot

Powered by AI

ExpectedFoot

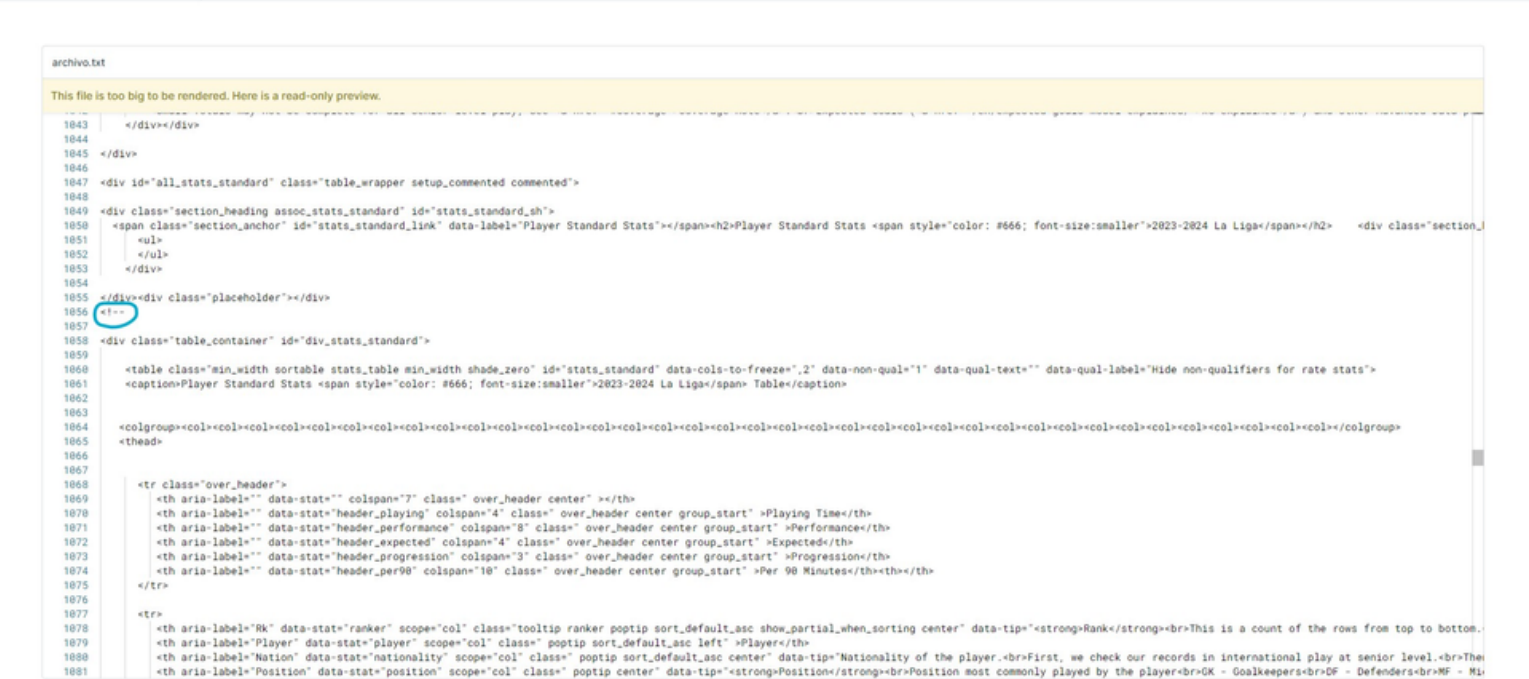
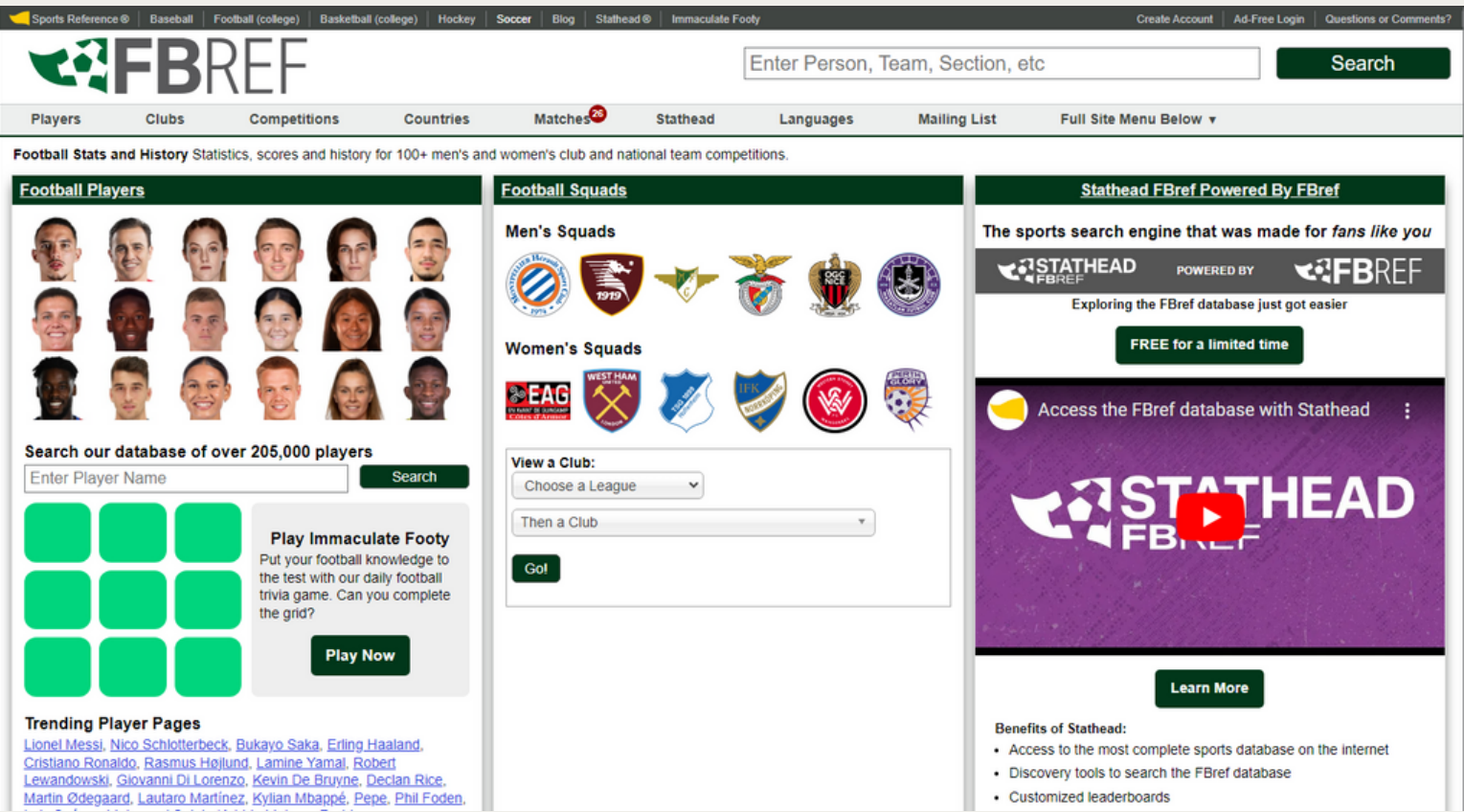
Se trata de un proyecto mediante el cual con datos recogidos de jugadores de las 5 grandes ligas (Liga Española, Liga Inglesa, Liga Alemana, Liga Italiana y Liga Francesa). Se tratará de predecir en base a unas estadísticas introducidas cuántos Expected Goals ('xG') que será capaz de anotar un futbolista.

Los Expected Goals ('xG') son una medida de la llamada "estadística avanzada" que se utiliza en fútbol para cuantificar la probabilidad de que un disparo a portería se convierta en gol. Un jugador con un número elevado de xG será un jugador muy a tener en cuenta.

Obtención de Datos

Para la obtención de los datos se ha decidido realizar Web Scrapping sobre la web **FBREF**, la cual nos proporciona los datos perfectos para realizar la predicción.

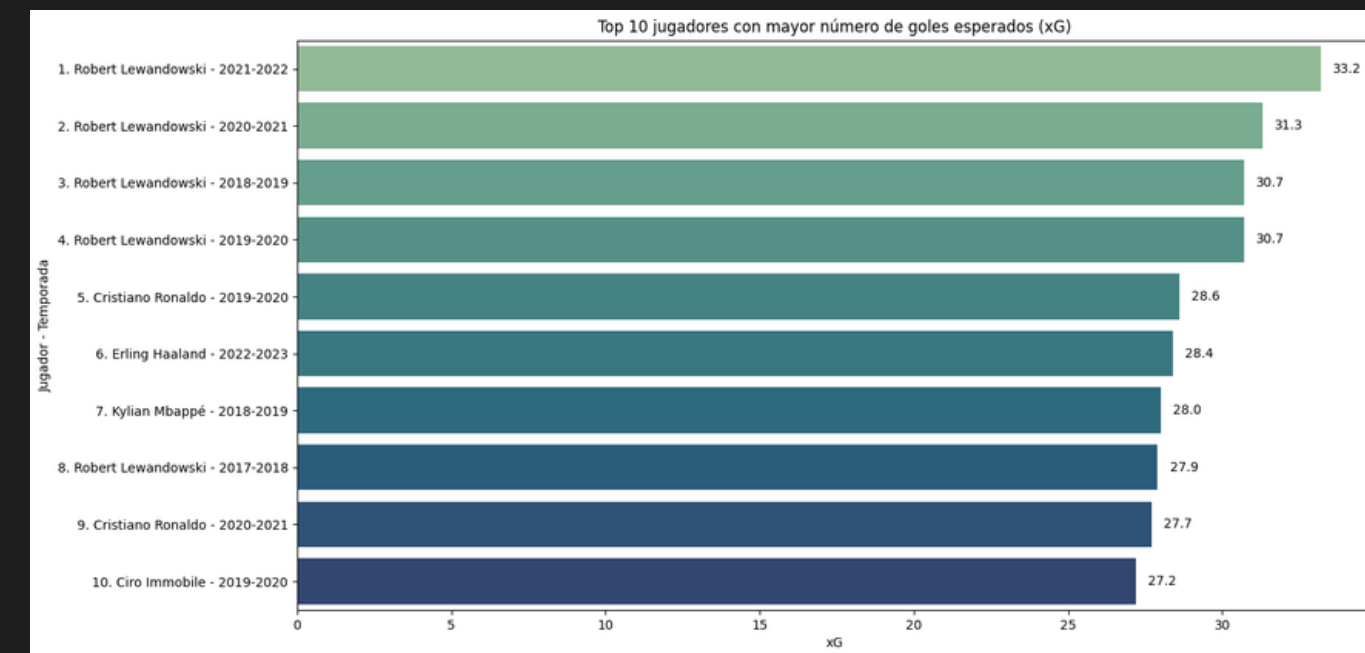
Se ha realizado mediante tecnicas avanzadas de scrapping así como medidas de seguridad para no ser detectados por los servidores de la web.



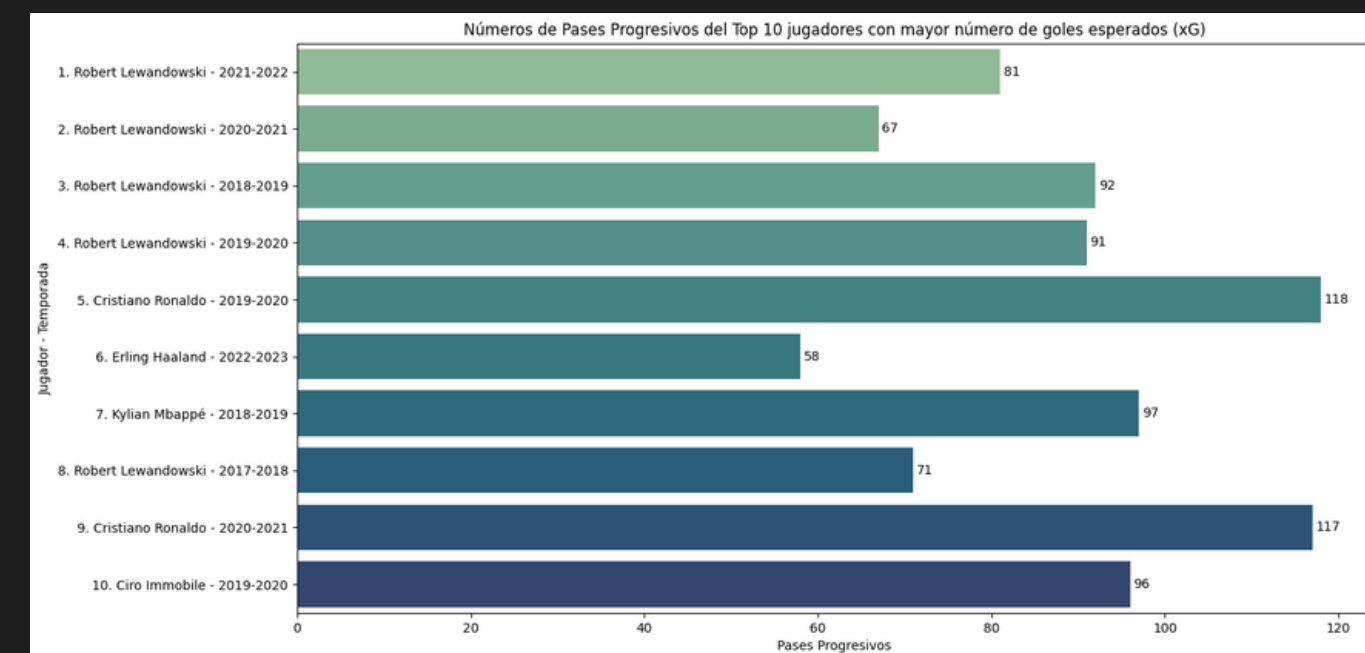
Exploración y Visualización de los Datos

Una vez tengamos los datos recopilados, pasaremos a la visualización de datos, en este caso con dos ejemplos

Los 10 jugadores con mayor número de goles esperados



Los 10 jugadores con mayor avance progresivo



Limpieza de Datos

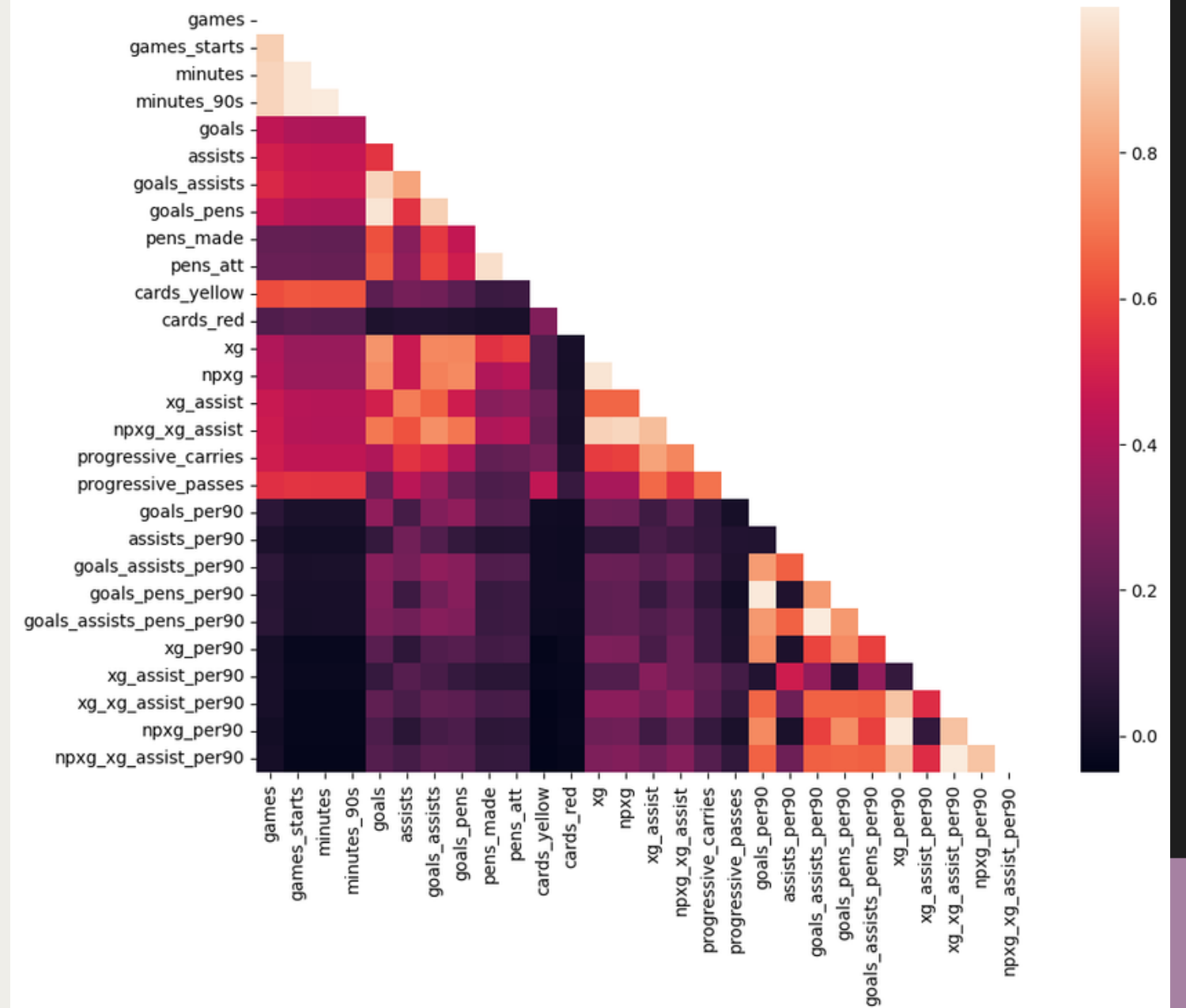
Para la limpieza de datos hemos eliminado las columnas innecesarias

En esta limpieza hemos seleccionado los datos no necesarios, como columnas "Expected" ya que ese no va a ser dato de entrenamiento. También eliminamos los datos de 2015 hasta 2017 ya que investigamos que faltaban datos de "XG".

player	season	team	games	games_starts	minutes	minutes_90s	goals	assists	goals
David Abraham	2017-2018	Eint Frankfurt	27	27	2302	25.6	0	2	
Amir Abrashi	2017-2018	Freiburg	12	11	850	9.4	0	0	
René Adler	2017-2018	Mainz 05	14	14	1260	14.0	0	0	
Ailton	2017-2018	Stuttgart	5	1	108	1.2	0	0	
Manuel Akanji	2017-2018	Dortmund	11	10	904	10.0	0	0	
Chadrac Akolo	2017-2018	Stuttgart	22	13	1102	12.2	5	0	
Kevin Akpoguma	2017-2018	Hoffenheim	22	17	1690	18.8	0	1	
David Alaba	2017-2018	Bayern Munich	23	17	1551	17.2	2	2	
Lucas Alario	2017-2018	Leverkusen	23	17	1550	17.2	9	4	
Miiko Albornoz	2017-2018	Hannover 96	10	7	703	7.8	0	0	
Thiago Alcántara	2017-2018	Bayern Munich	19	12	1247	13.9	2	2	
Stephan Ambrosius	2017-2018	Hamburger SV	1	1	45	0.5	0	0	
Nadiem Amiri	2017-2018	Hoffenheim	28	19	1744	19.4	2	3	
Waldemar Anton	2017-2018	Hannover 96	27	26	2278	25.3	1	0	
Dennis Aogo	2017-2018	Stuttgart	29	22	1954	21.7	0	6	
Charles Aránguiz	2017-2018	Leverkusen	27	27	2302	25.6	1	3	
Maximilian Arnold	2017-2018	Wolfsburg	29	28	2454	27.3	2	0	
Jann-Fiete Arp	2017-2018	Hamburger SV	18	8	850	9.4	2	0	
Takuma Asano	2017-2018	Stuttgart	15	7	721	8.0	1	0	
Santiago Ascacíbar	2017-2018	Stuttgart	29	27	2381	26.5	0	1	

Preparación de Datos

Se comprueba que la correlación de los datos se encuentre correctamente y se encuentran los datos en float o int para su posterior entrenamiento.



Entrenamiento del modelo

Tras realizar varios modelos de entrenamiento el elegido ha sido Decision Tree (árbol de decisiones), por su gran porcentaje de aciertos a la hora de entrenar los datos.

Se puede comprobar como el modelo tiene un 100% de acierto, he de aclarar que los modelos comparados obtenían entre un 70% y 90% de acierto, por eso elegimos este.

```
from sklearn.tree import DecisionTreeRegressor

dt_model = DecisionTreeRegressor()

# Entrenamiento del modelo
dt_model.fit(X_train, y_train)

# Predicción
y_pred = dt_model.predict(X_test)

y_test_list = y_test.tolist()

print("XG_Real    XG_Estimado    Error absoluto")
for i in range(20):
    r = y_test_list[i]
    e = y_pred[i]
    e_abs = abs(r - e)
```

```
zero_num = [x for x in results['Error absoluto'] if x == 0]

# Calcula el porcentaje de ceros
zero_per = (len(zero_num) / results['Error absoluto'].count()) * 100

print("Porcentaje de acierto:", zero_per, "%")
```

```
Porcentaje de acierto: 100.0 %
```


Procesamiento del lenguaje natural

Para el procesamiento de lenguaje natural, en la web hemos incorporado un ChatBot diseñado desde 0.

También contamos con varios idiomas en la web para que ningún usuario que navegue por la misma se encuentre con problemas gramaticales.



Aplicación Web

ExpectedFoot



Conclusión

Este trabajo en el que hemos aplicado todos los conocimientos aprendidos durante todo el curso nos ha servido para ver cuanto sabemos y de lo que somos capaces. Estamos muy orgullosos del resultado final que pensamos que es proporcional al esfuerzo invertido. Muchas gracias por leer hasta el final y esperamos que os haya gustado tanto el proyecto como a nosotros realizarlo.

**MUCHAS
GRACIAS**