

# MEMORIA DE FOOTBALL DATA



Javier Darío Castrillo Rodríguez

## Índice:

1-Descripción del problema .....	2
2-Herramientas y descripción del programa .....	3
3-Descripción de la aplicación .....	5
4-Explicación algoritmo .....	21
5-Análisis de resultados .....	22
6-DAFO .....	33
7-Líneas de futuro .....	35
8-Lecciones aprendidas .....	37
9-Bibliografía .....	38

## 1-Descripción del problema:

Una de las características principales del fútbol es que, al contrario que otros deportes (como por ejemplo la F1 o el baloncesto), sigue existiendo una gran parte de los seguidores e incluso de los equipos que rechazan el uso de análisis de datos para mejorar el rendimiento de sus equipos y prefieren utilizar un sistema más “tradicional”. Debido a esto, me planteé realizar una aplicación en la que realizara algún tipo de análisis de datos relacionado con el fútbol, y al final decidí realizar un sistema de recomendación de tácticas en función de los jugadores de un equipo.

Una vez elegido qué tipo de sistema de recomendación era necesario especificar con más detalle de qué iba a ir el sistema de recomendación exactamente. En mi caso decidí hacer un programa cuyo principal objetivo sea el de ayudar a los entrenadores para elegir la táctica más adecuada en función de los atributos de los jugadores.



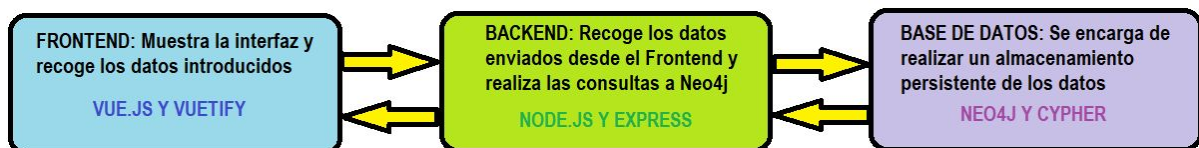
Como me parecía que este programa se podía quedar un poco corto y no iba a hacer un buen uso de la base de datos de neo4j decidí ampliar la idea a que, una vez decidida cuál es la mejor táctica del equipo, se digan qué jugadores encajan en esta táctica, qué jugadores no encajan en esta táctica y qué jugadores de la vida real se podrían incluir para sustituirlos.

Por lo tanto, el problema final es el de obtener la táctica más adecuada a través de los atributos de los jugadores y obtener jugadores que encajen de una forma mejor en esta táctica.

## 2-Herramientas y arquitectura del programa:

En mi caso he decidido realizar una **aplicación web**, ya que, hablando con los compañeros que utilizaron neo4j el año anterior recomendaron realizar este tipo de programa ya que era el que mejor compatibilidad tenía con respecto a drivers, ya que neo4j es una plataforma relativamente nueva que aún tiene algunos problemas de este tipo. Además, debido a que estoy bastante familiarizado con herramientas de desarrollo web, consideré que el proyecto que tenía en mente se podría desarrollar perfectamente en una aplicación de este tipo.

En una aplicación web normalmente existen **3 elementos**: la **interfaz de usuario** (o frontend), que se encarga de recoger los datos que nos ofrecen las personas que ejecutan la aplicación, la **base de datos**, que se encarga de almacenar todos los datos que necesitamos guardar, y el **backend**, que nos servirá para unir ambos elementos de una manera cómoda y evitar al frontend demasiada carga de trabajo.



La herramienta utilizada para la creación de la base de datos es **neo4j**, un sistema de **bases de datos mediante grafos**. La principal ventaja que tienen este tipos de bases de datos respecto a las típicas relacionales es que son muy potentes a la hora de almacenar las **relaciones entre los elementos**, ya que gracias a que cada uno de los elementos se almacenan en forma de nodo es muy fácil relacionarlos de formas distintas sin necesidad de realizar complejas operaciones como las que se realizan en SQL. Otra de las ventajas es que para muchos datos muy relacionados las **velocidades de respuesta** de este tipo de base de datos también son mucho más rápidas que las bases de datos relacionales, por lo que para un sistema de recomendación como el que voy a desarrollar son **ideales**.





La herramienta para el desarrollo del **backend**, que va a hacer de **nexo entre la interfaz de usuario y la base de datos es Node.js**. Las principales ventajas de este sistema son que, por una parte, la **comunicación con la interfaz** de usuario es muy **rápida y sencilla de hacer**, ya que este tipo de backend está optimizado para tal labor, y la segunda ventaja es que la **instalación del driver** para realizar la comunicación con la base de datos de **neo4j tiene una buena compatibilidad**, por lo que no tendremos muchos problemas que sí que se pueden tener en otro tipo de aplicaciones que aún dan problemas con neo4j.



En cuanto al frontend (interfaz de usuario), el plugin de JavaScript que he decidido utilizar es **Vue.js**, ya que es con el que más estoy familiarizado y además es un sistema que cada vez tiene más usuarios y que están en pleno auge, por lo que si en un futuro quiero continuar con la aplicación tengo asegurado que esta herramienta va a estar a pleno funcionamiento. También necesitaré un plugin de Vue.js que permita realizar el diseño gráfico como tal, y en mi caso he decidido optar por **Vuetify** ya que permite un diseño de interfaz limpio, sencillo y que recuerda a las aplicaciones de Google, por lo que los potenciales usuarios de mi aplicación se encontrarán en un entorno cómodo de uso.



### 3-Descripción de la aplicación.

-Antes de comenzar a hablar de la aplicación a nivel interno, para las personas que no estén muy familiarizadas con el fútbol voy a explicar los atributos básicos de los jugadores y los diferentes estilos de juego.

#### **Atributos:**

**Velocidad:** Es la característica de un jugador para poder desplazarse por el campo lo más rápido posible. Esta habilidad es clave en jugadores como por ejemplo extremos y laterales y el estilo de juego en el que es primordial es el de contraataques.

**Resistencia:** Es la capacidad de un jugador de poder mantener la energía a lo largo de un partido, es decir, de estar menos cansado que el resto de jugadores al final de los partidos. Es clave en estilos de juego como el Gegenpressing, en el que el equipo está constantemente presionando al rival y por lo tanto se exige no dejar de correr en todo el partido.

**Fuerza:** Es la fortaleza física de un jugador (se incluye la capacidad de salto en este atributo). Es muy importante sobre todo en jugadores que juegan en la zona central del campo, como por ejemplo los defensas centrales, los delanteros o los mediocentros, y la estrategia en la que más importa es en la de ataque por bandas.

**Remate:** Es la capacidad de un jugador de acertar cuando va a tirar a la portería rival. Esta habilidad es muy importante para jugadores ofensivos, en especial los delanteros, ya que si tiene una característica de remate baja van a fallar muchos remates a portería.

**Pase:** Es el atributo que define la habilidad de un jugador para acertar con los pases que realiza a sus compañeros de equipo. Es especialmente importante en jugadores creativos, como por ejemplo los mediocentros y los mediapuntas, y tiene una importancia muy relevante en las tácticas de posesión, especialmente en el de posesión lenta.

**Regate:** Es la habilidad que tiene un jugador de driblar a un defensor para poder seguir la jugada. Este atributo es imprescindible en jugadores que juegan en posiciones ofensivas, en especial los interiores y los extremos.

**Marcaje:** Es la capacidad que tiene un futbolista para poder seguir al jugador rival que tiene (o no) la pelota y estar atento para poder aprovechar el posible error de uno de estos jugadores. Este atributo es importante en estrategias más defensivas, como el catenaccio, aunque no se puede despreciar en estrategias más ofensivas.

**Entradas:** Este atributo define la calidad de los jugadores a la hora de intentar realizar un robo a un jugador del equipo enemigo. Al igual que en Marcaje, esta habilidad es clave en jugadores defensivos y en estilos de juego como el catenaccio.

**Cabeza:** Este atributo se caracteriza por definir la habilidad de un jugador a la hora de realizar un remate de cabeza. Es de especial importancia en delanteros y en defensas centrales, ya que normalmente suelen ser los jugadores que más batallas de cabeza suele tener junto con los mediocentros. Es importante en el ataque por bandas ya que normalmente utiliza delanteros altos con buen remate de cabeza.

**Decisiones:** Es la capacidad de un jugador de tomar la decisión correcta en un momento determinado. Este atributo afecta a todas las posiciones y tácticas, ya que por muy buena técnica o capacidad física que tenga un jugador si le falta inteligencia nunca va a poder explotar todo su potencial.

**Colocación:** Es la capacidad que tiene un jugador para estar colocado siempre en la posición más adecuada que le corresponde en el campo. Este atributo también afecta a todas las posiciones y las tácticas por la misma razón que la toma de decisiones.

**Paradas:** Es la habilidad que tiene un portero para poder hacer que el balón no entre en su portería. En este caso el único jugador al que le afecta este atributo es a los porteros, ya que el resto de jugadores nunca van a tener que ejercer este rol salvo casos muy puntuales. Es especialmente importante tener a un buen portero en estilos de juego defensivos, ya que normalmente al portero le van a llegar más tiros.



## **Estilos de juego:**

**Posesión rápida:** En el estilo de posesión rápida, si bien se intentará mantener el control del balón la mayoría de las veces, en los momentos en los que se pueda realizar un ataque rápido se realizará. La zona de juego normalmente será por el centro, si bien se utilizarán las bandas para dar profundidad al equipo. En ataque la mentalidad será ofensiva, subiendo a atacar la mayoría de los jugadores del equipo.

**Posesión lenta:** En el estilo de posesión lenta, la principal característica es que en todo momento intentaremos mantener el balón en nuestro poder, normalmente mediante pases de poco riesgo, no realizando pases largos salvo en ocasiones claras de gol, y jugando principalmente por el centro. En ataque, la mentalidad del equipo será ofensiva, ya que se intentará hacer que nuestro equipo no pierda el control del balón en ningún momento. En defensa, se optará por presionar al rival para que pierda la pelota lo antes posible.

**Gegenpressing:** En el estilo de gegenpressing, todos los jugadores del equipo participarán en una presión intensa con la intención de asfixiar al equipo rival y evitar que este pueda crear ocasiones peligrosas. En ataque, el estilo será ofensivo y rápido, intentando pillar a la defensa descolocada tras haber recuperado el balón. En defensa, todos los jugadores participarán en la presión, por lo que estos deben de contar con buenas capacidades físicas para llegar a la defensa lo antes posible.

**Contraataques:** En el estilo de contraataque, la principal peculiaridad es que intentaremos realizar ataques muy rápidos para intentar sorprender a la defensa e intentando aprovechar la velocidad de los atacantes de nuestro equipo. En ataque, la mentalidad será cautelosa, ya que no es necesario que todos los jugadores se sumen al ataque. En defensa, se puede o bien jugar a presionar al rival y realizar contraataques rápidos o bien esperar en el área y aprovechar el hecho de que los defensas están muy adelantados para sorprender.

**Ataque Por Bandas:** En el estilo por bandas, el equipo se especializa en acumular muchos jugadores en las bandas para obtener superioridades a lo largo del campo de juego. En ataque, la clave son los jugadores de banda, cuyo rol será tanto colocar centros como regatear para intentar marcar gol. En defensa, se aplicará una estrategia cautelosa inicialmente, aunque en algunas situaciones sí que se podrá presionar al equipo contrario.

**Catenaccio:** En este estilo de juego, el principal objetivo es que a nuestro equipo no le lleguen a marcar ningún objetivo mediante el uso de una defensa férrea. En ataque, no se arriesgará las posiciones de los jugadores defensivos, atacando con el menor número de jugadores posibles. En defensa, no se tomará ningún tipo de riesgo, despejando el balón siempre que sea necesario.



-Una vez explicados estos conceptos, voy a explicar cada una de las partes de la aplicación.

### **Base de datos:**

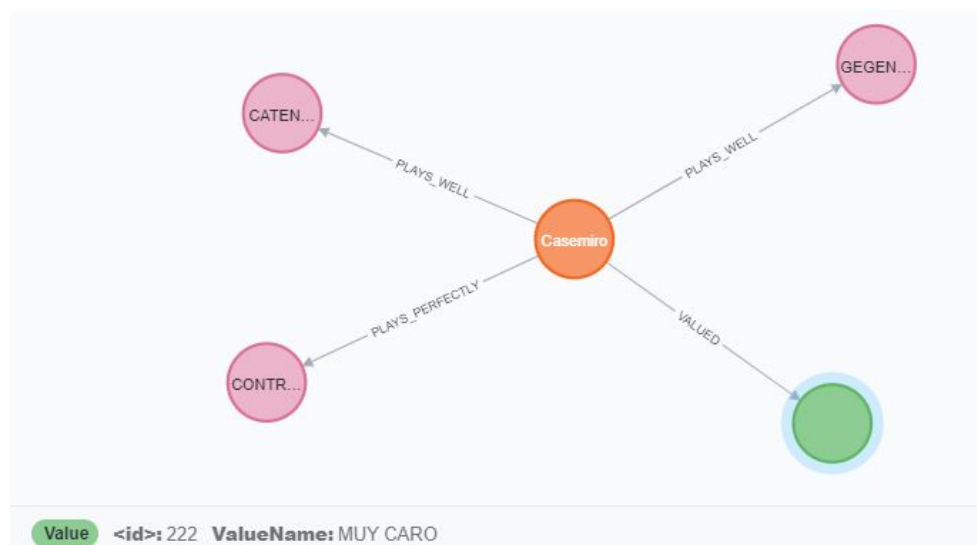
Como en Kaggle no fui capaz de encontrar una base de datos que fuera suficientemente adecuada al trabajo que iba a realizar, decidí que mi única solución era crear yo mismo la base de datos de jugadores.

En mi caso decidí optar por utilizar el videojuego **Football Manager** ya que, no sólo tiene más de 60000 jugadores para escoger en su base de datos, sino que además es muy fácil realizar la exportación de los jugadores con los atributos que prefieras para poder crear la base de datos de una forma muy rápida.

Tras hablarlo con el profesor, decidimos que lo ideal era coger el **once inicial** de los 20 equipos de la **primera división**, que nos iba a ofrecer alrededor de 200 jugadores, cifra más que suficiente para comenzar a trabajar en la aplicación.

Una vez obtenido el archivo .csv con el que iba a realizar la importación de los datos a neo4j, realicé la **creación de los nodos**, tanto de los jugadores como de otros elementos que se van a utilizar más adelante como su táctica más adecuada o el rango de precio en el que se encuentran.

En la siguiente captura se puede ver un nodo de jugador y las diferentes conexiones que posee este:



En este ejemplo aparece el jugador Casemiro, cuyo valor está en el rango Muy Caro y además juega perfectamente en estilos de contraataque y juega bien en catenaccio y gegenpressing.

Como se puede ver en la captura, hay 3 tipos de nodos:

**Nodo jugador:** Este nodo es el que contiene la información básica acerca de los jugadores, como por ejemplo sus atributos, su edad, su valor de mercado o la adecuación de cada uno a los estilos de juego.

**Nodo estilo de juego:** Este nodo tiene como principal funcionalidad unirse con el nodo de jugador mediante enlaces PLAYS\_PERFECTLY, PLAYS\_WELL, PLAYS\_AVERAGE y PLAYS\_BAD, que nos permitirán saber rápidamente si un jugador puede jugar perfectamente, bien o mal en una táctica en concreto

**Nodo rango de valores de mercado:** En la aplicación existe un filtro de valores de mercado, y por lo tanto es útil relacionar a los jugadores con su rango de valor de mercado mediante la relación VALUED para poder realizar queries que sean mucho más sencillas y rápidas que si tuviéramos que evaluar cada vez el rango de valores del jugador.

### **Backend:**

Como ya he explicado anteriormente, el backend es el **enlace entre la interfaz de usuario y la base de datos**, y en mi caso resulta muy útil para no llenar a la interfaz con trabajos que no debe realizar como por ejemplo lanzar las queries a la base de datos.

En el caso de mi aplicación, el backend realizará acciones cuando sea llamado desde el frontend mediante una petición **POST** a la formación en la que se encuentra el usuario.

```
app.post('/433', (req, res) => {  
  console.log('Se ha recibido una petición post en 433')
```

En las siguientes variables recogemos cuál es el estilo del equipo que ha calculado nuestra aplicación en el frontend y el valor de mercado que ha establecido el usuario. Si el propio usuario no establece ningún valor, entonces se pone por defecto el valor más caro de todos.

```
var team = req.body.z12  
var value = req.body.z13
```

Ahora tenemos que obtener la **cadena exacta** que tenemos que colocar en la query para asegurarnos de que todos los jugadores que cumplen las condiciones antes mencionadas se buscan.

```
if(team.localeCompare(posesionLenta) == 0){
    estiloEquipo = "POSESION LENTA"
}
else if(team.localeCompare(posesionRapida) == 0){
    estiloEquipo = "POSESION RAPIDA"
}
else if(team.localeCompare(gegenpressing) == 0){
    estiloEquipo = "GEGENPRESSING"
}
else if(team.localeCompare(counter) == 0){
    estiloEquipo = "CONTRAATAQUE"
}
else if(team.localeCompare(bandas) == 0){
    estiloEquipo = "ATAQUE POR BANDAS"
}
else{
    estiloEquipo = "CATENACCIO"
}
var valorJugador = ""
if(value.localeCompare("do") == 0){
    valorJugador = "MUY BARATO"
}
else if(value.localeCompare("re") == 0){
    valorJugador = "BARATO"
}
else if(value.localeCompare("mi") == 0){
    valorJugador = "CARO"
}
else{
    valorJugador = "MUY CARO"
}
```

EstiloEquipo será la String que define al estilo en la base de datos de neo4j y valorJugador es la String que define el valor de cada jugador también en neo4j.

Lo siguiente será realizar la **query**. En este ejemplo vamos a mirar la query que busca al portero a recomendar.

Aquí definimos una variable que va a ser la query que se envíe para buscar el portero.

```
let queryPor = "MATCH (v:Value)<-[:VALUED]-(n:Player)-[:PLAYS_PERFECTLY]->(p:Playstyle)
WHERE n.position = 'PO' AND p.Type = '"
```

Tenemos que concatenar tanto el estilo del equipo como el valor del jugador, que son los strings que hemos calculado antes.

```
queryPor = queryPor.concat(estiloEquipo + "' AND v.ValueName ='"  
+ valorJugador + "' RETURN n limit 1")
```

Tendremos que repetir esto para cada una de las posiciones que nos podemos encontrar en el campo, obteniendo todas las queries necesarias para poder realizar la llamada a la base de datos.

Una vez que están las queries, realizamos la **llamada** mediante **run**:

```
session  
  .run(queryPor)  
  .then(function(result){  
    result.records.forEach(function(record){
```

Como se puede ver en este trozo de código, cogeremos la sesión y haremos un `.run`, que ejecutará la query en la base de datos, y luego en la zona de `.then` **recogeremos el resultado** en “result”, cogiendo en el caso de que lo necesitemos varios resultados mediante el uso de un `forEach`.

En mi caso fue necesario **encadenar varias queries seguidas** (una para cada jugador), y, como no es algo que sea muy común en neo4j, ya que este sistema está pensado para realizar las queries de una en una, tuve que investigar hasta llegar a la conclusión de que tenía que colocar un nuevo `session.run` dentro del `.then` de la anterior query, realizando una cadena bastante larga de queries.

```
session  
  .run(queryPor)  
  .then(function(result){  
    result.records.forEach(function(record){  
      console.log(record._fields[0].properties.Nombre)  
      portero.push(record._fields[0].properties.Nombre);  
    })  
    session  
      .run(queryLatIzq)  
      .then(function(result){  
        result.records.forEach(function(record){  
          console.log(record._fields[0].properties.Nombre)  
          lateralIzquierdo.push(record._fields[0].properties.Nombre);  
        })  
        session  
          .run(queryCen)  
          .then(function(result){  
            result.records.forEach(function(record){  
              console.log(record._fields[0].properties.Nombre)  
              central.push(record._fields[0].properties.Nombre);  
            })
```

En el .then de la query mi objetivo es **añadir todos los atributos** que necesito de los jugadores para luego **devolverlos al frontend**. Es por ello que tengo un array para cada posición y lo que hago es añadir mediante push a este los datos necesarios.

```
portero.push(record._fields[0].properties.Nombre);portero.push(record._fields[0].properties.age.low)
```

Es importante remarcar el hecho de que para recibir los valores de tipo enteros la base de datos nos devolverá su valor low y su valor high, en nuestro caso elegiremos el valor low ya que este es el que nos va a servir.

En el caso de mi aplicación, el siguiente paso es evaluar **qué datos tengo que enviar desde el backend hasta el frontend** y cuáles no son necesarios, ya que si por ejemplo el jugador que introdujo el usuario en su aplicación encajaba en la táctica adecuadamente no era necesario que recomendemos ningún jugador sustituto.

```
if(estiloEquipo.localeCompare("POSESION LENTA") == 0){
    //Cogemos el valor del jugador de comparación y comprobamos si
    if(poslen.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
else if(estiloEquipo.localeCompare("POSESION RAPIDA") == 0){
    if(posRap.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
else if(estiloEquipo.localeCompare("GEGENPRESSING") == 0){
    if(gegen.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
else if(estiloEquipo.localeCompare("CONTRAATAQUE") == 0){
    if(count.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
else if(estiloEquipo.localeCompare("ATAQUE POR BANDAS") == 0){
    if(ban.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
else{
    if(caten.localeCompare(perfect) != 0){
        porteroMostrar = true
    }
}
```

Lo que hago para cada uno de los jugadores es evaluar primero cuál es la táctica general del equipo para saber en qué caso "if" me encuentro. Una vez realizado esto, tengo que evaluar si el jugador original que introdujo el usuario encaja bien en la táctica o no lo hace. Si no lo hace, ponemos porteroMostrar a true, lo que significa



que más adelante enviaremos los datos que hemos obtenidos desde neo4j a la interfaz de usuario para mostrarlos.

Una vez que hemos realizado estas acciones para cada uno de los jugadores, lo siguiente que haremos es declarar 3 variables que van a ser las cadenas de texto que se van a mostrar dependiendo de si no es necesario realizar una recomendación, si realizamos una recomendación o si no hemos encontrado ningún jugador en la base de datos que cumpla las características pedidas.

```
var notFound = "No se ha encontrado ningún elemento en la base de datos que mejore el resultado."
var innecesario = "El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación."
var recomendacion = "La recomendación de jugador que encaja mejor en la táctica es la siguiente:"
```

Ahora lo que necesitaremos es formar un JSON que nos permita enviar los datos desde el backend hasta el frontend, y en nuestro caso lo denominaremos respuesta.

Para que tenga una forma balanceada, en los casos en los que no sea necesario enviar el array que contiene los atributos del jugador a la hora de realizar la recomendación lo que haremos será enviar un array de ceros que no vamos a utilizar.

Tenemos 3 casos que se pueden dar:

**1-Tenemos que realizar una recomendación.** En este caso enviamos la string "recomendacion" y el array de valores del jugador en concreto.

**2-No se ha encontrado ningún elemento en la base de datos.** En este caso enviamos la string "notFound" y un array de 0s.

**3-El jugador que introdujo el usuario originalmente es suficientemente bueno y no hay que recomendarlo.** En este caso enviamos la string "innecesario" y un array de 0s.

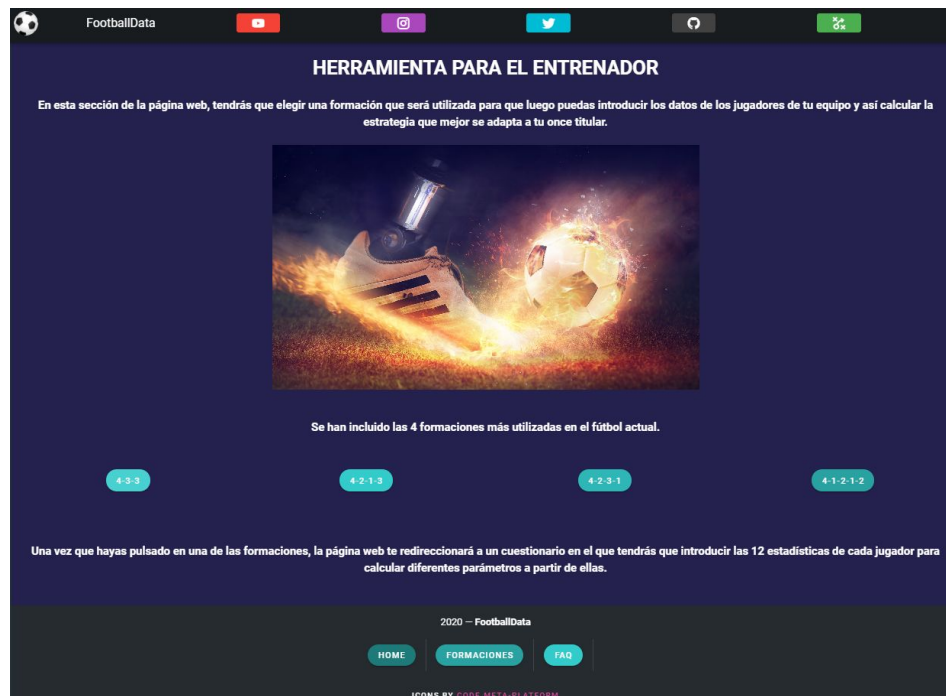
```
if(porteroMostrar == true){
  //Si portero está empty, entonces no podemos
  if(porteroEmpty == true){
    string = notFound
    respuesta.portero = {string, arrayCeros}
  }
  else{
    string = recomendacion
    respuesta.portero = {string, portero}
  }
}
```

Una vez realizada esta tarea, podemos por fin enviar el JSON al frontend y se termina el trabajo del backend.

```
res.send(respuesta)
```

## Frontend:

Lo primero que se encuentra un usuario nada más entrar a la página web es el **apartado de formaciones**, en el que el usuario puede elegir qué formación quiere utilizar:



Una vez que se ha elegido la formación a utilizar, se llevará a la página destinada a tal fin, cuyo diseño de la parte de arriba es el siguiente:



Como se puede ver, arriba se muestra un **banner** en el que hay links a diferentes páginas webs, y debajo encontramos una explicación de la formación tanto a nivel gráfico como en forma de texto para ayudar al usuario a saber si la formación que han elegido es la adecuada o es mejor utilizar otra diferente.

El siguiente apartado con el que nos encontramos es el de **introducir los datos del once inicial de nuestro equipo**. En función de la media de los 12 atributos que existen en nuestra aplicación se calculará la táctica global más adecuada para el equipo, mientras que en función de los atributos de cada jugador se calculará la adecuación de cada uno de ellos para cada una de las 6 tácticas.

LATERAL IZQUIERDO	
Nombre del jugador	Apellido del jugador
0 / 10	0 / 10
VEL: 17	RES: 16
REM: 13	PAS: 13
MAR: 16	ENT: 13
DEC: 15	COL: 16
	PAR: 1
	FUE: 14
	REG: 14
	CAB: 8
	EDA: 27

Como se puede apreciar en la captura, existe **un slider** para cada uno de los atributos, además de para la edad, el nombre y el apellido del jugador.

Una vez que se ha realizado la introducción de los datos, se pulsará el siguiente botón:



Una vez pulsado este botón, lo que va a suceder es que se calcularán las **medias de los atributos de los jugadores** en el propio frontend y las tácticas recomendadas tanto para el equipo como para los jugadores.

En la siguiente imagen se puede ver una overview de esta parte de la interfaz. en la que, para cada uno de los apartados, hay un menú desplegable para evitar que se muestre demasiada información a la vez.

Los 4 primeros apartados tienen que ver con la táctica recomendada al equipo principal, mientras que los últimos 11 son las recomendaciones de las tácticas de los jugadores de forma individual.

Lo primero que se va a mostrar son las medias de cada atributo: ▼

Ahora se van a desglosar las recomendaciones para cada una de las 6 estrategias posibles: ▼

Aquí se va a mostrar la estrategia recomendada: ▼

Esta es la explicación de la estrategia elegida: ▼

PORTERO ▼

LATERAL IZQUIERDO ▼

DEFENSA CENTRAL ▼

DEFENSA CENTRAL ▼

LATERAL DERECHO ▼

PIVOTE ▼

MEDIOCENTRO ▼

MEDIOCENTRO ▼

EXTREMO IZQUIERDO ▼

EXTREMO DERECHO ▼

DELANTERO ▼

Los primeros datos que se van a mostrar son las **medias de los atributos** de los jugadores.

Lo primero que se va a mostrar son las medias de cada atributo: ^

VEL 15	RES 14	FUE 15
REM 14	PAS 15	REG 13
MAR 13	ENT 13	CAB 14
DEC 15	COL 15	PAR 12

Una vez mostradas las medias, se mostrarán los **ratings** para cada una de las tácticas y la descripción de la táctica recomendada:

Ahora se van a desglosar las recomendaciones para cada una de las 6 estrategias posibles: ^

POSESIÓN LENTA 1.5550037414491136	POSESIÓN RÁPIDA 1.5627628926427473	GEGENPRESSING 1.5498025036064962
POR BANDAS 1.6268032481734842	CONTRAATAQUE 1.61612234631672	CATENACCIO 1.6181572711619898

Como se puede ver, en este ejemplo que he propuesto el valor mayor es el de Por Bandas, mientras que el menor es el de Gegenpressing.

El cálculo de esto se ha realizado mediante las siguientes fórmulas:

```
var velx = parseFloat(this.mediaVelocidad) / 14.5;
var resx = parseFloat(this.mediaResistencia) / 12.5;
var fuex = parseFloat(this.mediaFuerza) / 13;
var remx = parseFloat(this.mediaRemate) / 10.5;
var pasx = parseFloat(this.mediaPase) / 12.5;
var regx = parseFloat(this.mediaRegate) / 12;
var marx = parseFloat(this.mediaMarcaje) / 9.5;
var entx = parseFloat(this.mediaEntradas) / 9.5;
var cabx = parseFloat(this.mediaCabeza) / 10.5;
var decx = parseFloat(this.mediaDecisiones) / 12.5;
var colx = parseFloat(this.mediaColocacion) / 12.5;
var parx = (parseFloat(this.mediaParadas) * 11 - 10) / 15;

this.posRapidaRec = 0.06 * velx + 0.06 * resx + 0.06 * fuex + 0.08 * remx + 0.15 * pasx + 0.1 * regx + 0.08 * marx + 0.07 * entx + 0.07 * cabx + 0.1 * decx + 0.12 * colx + 0.05 * parx;
this.posLentaRec = 0.1 * velx + 0.08 * resx + 0.05 * fuex + 0.1 * remx + 0.12 * pasx + 0.1 * regx + 0.08 * marx + 0.07 * entx + 0.05 * cabx + 0.12 * decx + 0.08 * colx + 0.05 * parx;
this.gegenRec = 0.09 * velx + 0.15 * resx + 0.07 * fuex + 0.08 * remx + 0.08 * pasx + 0.08 * regx + 0.08 * marx + 0.07 * entx + 0.05 * cabx + 0.08 * decx + 0.12 * colx + 0.05 * parx;
this.counterRec = 0.15 * velx + 0.07 * resx + 0.07 * fuex + 0.08 * remx + 0.07 * pasx + 0.09 * regx + 0.08 * marx + 0.08 * entx + 0.05 * cabx + 0.1 * decx + 0.1 * colx + 0.06 * parx;
this.wingPlayRec = 0.12 * velx + 0.05 * resx + 0.1 * fuex + 0.08 * remx + 0.08 * pasx + 0.1 * regx + 0.08 * marx + 0.08 * entx + 0.1 * cabx + 0.07 * decx + 0.08 * colx + 0.06 * parx;
this.catenaccionRec = 0.08 * velx + 0.07 * resx + 0.1 * fuex + 0.07 * remx + 0.07 * pasx + 0.08 * regx + 0.08 * marx + 0.08 * entx + 0.08 * cabx + 0.12 * decx + 0.1 * colx + 0.06 * parx;
```

Esto se muestra de la siguiente forma:

Aquí se va a mostrar la estrategia recomendada:

RECOMENDACIÓN DE ESTRATEGIA

El estilo recomendado es ataque por las bandas.

Esta es la explicación de la estrategia elegida:

EXPLICACIÓN DE LA ESTRATEGIA

En el estilo por bandas, el equipo se especializará en acumular muchos jugadores en las bandas para obtener superioridades a lo largo del campo de juego. En ataque, la clave son los jugadores de banda, cuyo rol será tanto colocar centros como regatear para intentar marcar gol. En defensa, se aplicará una estrategia cautelosa inicialmente, aunque en algunas situaciones sí que se podrá presionar al equipo contrario.

Una vez explicada la táctica general del equipo, lo que se va a hacer es **mostrar la recomendación** de cada uno de los jugadores:

<p>DELANTERO</p> <p>POSESIÓN LENTA</p> <p>Encaja perfectamente en esta táctica</p>	<p>POSESIÓN RÁPIDA</p> <p>Encaja bien en esta táctica</p>	<p>GEGENPRESSING</p> <p>Encaja bien en esta táctica</p>
<p>POR BANDAS</p> <p>Encaja mal en esta táctica</p>	<p>CONTRAATAQUE</p> <p>Encaja perfectamente en esta táctica</p>	<p>CATENACCIO</p> <p>Encaja perfectamente en esta táctica</p>

Como se puede ver en esta captura, se explica si el jugador va a encajar perfectamente en la táctica, si va a encajar bien, si va a encajar mal o si no va a encajar en absoluto.



El código que hemos utilizado por ejemplo para calcular la **táctica recomendada** del delantero centro es el siguiente:

```
var vel = this.velocidad11.val / 14;  
var res = this.resistencia11.val / 13.5;  
var fue = this.fuerza11.val / 13.5;  
var rem = this.remate11.val / 14.5;  
var pas = this.pase11.val / 12.5;  
var reg = this.regate11.val / 12.5;  
var mar = this.marcaje11.val / 6;  
var ent = this.entradas11.val / 7.5;  
var cab = this.cabeza11.val / 13.5;  
var dec = this.decisiones11.val / 13;  
var col = this.colocacion11.val / 8;  
var par = this.paradas11.val;  
  
var posesionLenta = 0.09 * vel + 0.1 * res + 0.09 * fue + 0.13 * rem + 0.15 * pas + 0.1 * reg + 0.03 * mar + 0.02 * ent + 0.07 * cab + 0.1 * dec + 0.12 * col + 0 * par;  
var posesionRapida = 0.13 * vel + 0.11 * res + 0.09 * fue + 0.15 * rem + 0.12 * pas + 0.1 * reg + 0.03 * mar + 0.02 * ent + 0.05 * cab + 0.12 * dec + 0.08 * col + 0 * par;  
var gegenpressing = 0.12 * vel + 0.18 * res + 0.11 * fue + 0.13 * rem + 0.08 * pas + 0.08 * reg + 0.03 * mar + 0.02 * ent + 0.05 * cab + 0.08 * dec + 0.12 * col + 0 * par;  
var contraataque = 0.18 * vel + 0.11 * res + 0.1 * fue + 0.14 * rem + 0.07 * pas + 0.09 * reg + 0.03 * mar + 0.03 * ent + 0.05 * cab + 0.1 * dec + 0.1 * col + 0 * par;  
var bandas = 0.15 * vel + 0.09 * res + 0.13 * fue + 0.13 * rem + 0.08 * pas + 0.08 * reg + 0.03 * mar + 0.03 * ent + 0.13 * cab + 0.07 * dec + 0.08 * col + 0 * par;  
var catenaccio = 0.12 * vel + 0.1 * res + 0.14 * fue + 0.13 * rem + 0.07 * pas + 0.08 * reg + 0.03 * mar + 0.03 * ent + 0.08 * cab + 0.12 * dec + 0.1 * col + 0 * par;
```

En la siguiente imagen podemos ver la siguiente parte del programa, que es la parte en la que se realiza una **recomendación utilizando datos de la base de datos**.

Una vez mostradas las recomendaciones en cuanto a las tácticas tanto del equipo como las de cada jugador se refiere, ahora se procederá a recomendar jugadores para reforzar las posiciones más débiles

Lo primero es seleccionar si se quiere tener en cuenta el valor y, en el caso de que sí se quiera tener en cuenta, se elegirá cuál es el rango de valor deseado.

☒ ¿Tener en cuenta el precio?

☐ 1-3 MILLONES ☐ 4-9 MILLONES ☐ 10-25 MILLONES ☒ +25 MILLONES

Ahora, una vez que se pulse el botón de recomendación automática, se recomendarán jugadores que cubran las posiciones en los que el jugador no coincide con la táctica.

RECOMENDACIÓN AUTOMÁTICA

Como se puede ver en esta captura de pantalla, el usuario puede elegir si quiere tener en cuenta o no el precio del jugador. En caso de que no quiera tenerlo en cuenta el rango de precio será el más caro, es decir, +25 millones de euros.

Todos los datos que se van a enviar al backend, entre los que se encuentran el precio del jugador a recomendar y los datos de **todos los jugadores los tendremos que poner en formato JSON**, quedando de la siguiente forma:

```
let por = {a: "PO", b: this.porteroPosLen, c: this.porteroPosRap, d: this.porteroGegen, e: this.porteroCounter, f: this.porteroBandas, g: this.porteroCaten,};  
let lti = {a: "LI", b: this.latIzqPosLen, c: this.latIzqPosRap, d: this.latIzqGegen, e: this.latIzqCounter, f: this.latIzqBandas, g: this.latIzqCaten,};  
let dfc1 = {a: "DC1", b: this.defPosLen1, c: this.defPosRap1, d: this.defGegen1, e: this.defCounter1, f: this.defBandas1, g: this.defCaten1,};  
let dfc2 = {a: "DC2", b: this.defPosLen2, c: this.defPosRap2, d: this.defGegen2, e: this.defCounter2, f: this.defBandas2, g: this.defCaten2,};  
let ltd = {a: "LD", b: this.latDerPosLen, c: this.latDerPosRap, d: this.latDerGegen, e: this.latDerCounter, f: this.latDerBandas, g: this.latDerCaten,};  
let piv = {a: "PI", b: this.pivPosLen, c: this.pivPosRap, d: this.pivGegen, e: this.pivCounter, f: this.pivBandas, g: this.pivCaten,};  
let mc1 = {a: "MC1", b: this.medPosLen1, c: this.medPosRap1, d: this.medGegen1, e: this.medCounter1, f: this.medBandas1, g: this.medGegen1,};  
let mc2 = {a: "MC2", b: this.medPosLen2, c: this.medPosRap2, d: this.medGegen2, e: this.medCounter2, f: this.medBandas2, g: this.medGegen2,};  
let ei = {a: "EI", b: this.extIzqPosLen, c: this.extIzqPosRap, d: this.extIzqGegen, e: this.extIzqCounter, f: this.extIzqBandas, g: this.extIzqCaten,};  
let ed = {a: "ED", b: this.extDerPosLen, c: this.extDerPosRap, d: this.extDerGegen, e: this.extDerCounter, f: this.extDerBandas, g: this.extDerCaten,};  
let dlc = {a: "DL", b: this.delPosLen, c: this.delPosRap, d: this.delGegen, e: this.delCounter, f: this.delBandas, g: this.delCaten,};  
let json = { z1: por, z2: lti, z3: dfc1, z4: dfc2, z5: ltd, z6: piv, z7: mc1, z8: mc2, z9: ei, z10: ed, z11: dlc, z12: this.textoRecomendatorio, z13: letsyou,};
```

En mi caso, para realizar las llamadas al backend he utilizado Axios, un plugin muy sencillo que nos permite enviar peticiones de http como por ejemplo POST o GET a un servidor o backend.

Para realizar las llamadas, se puede o bien utilizar **las llamadas “Promises” de JavaScript** o bien utilizar **el método de la función asíncrona**. En mi caso yo he decidido optar por la segunda opción (data es el JSON que se ha creado en la anterior captura):

```
async callPost(deta) {  
  try {  
    var datos = await axios.post("http://127.0.0.1:3000/433", deta);  
    this.setState({  
      data: datos.data,  
    });  
  } catch (error) {  
    console.log(error);  
  }  
}
```

En “datos” se encuentran todos los datos que nos ha enviado el backend una vez hemos realizado las consultas en la base de datos.

Para acceder, por ejemplo, a los datos del portero que vamos a recomendar, podemos hacerlo de la siguiente forma:

```
this.porteroMostrarString = datos.data.portero.string;
```

Por último, lo que tenemos que decidir es **qué tenemos que mostrar** en cada uno de los cuadros de texto que corresponden a cada uno de los jugadores.

Si el String que hemos enviado de un jugador desde el backend **nos dice que no tenemos que recomendar a ningún jugador**, entonces imprimimos exclusivamente esta cadena de texto.

Si el String que hemos enviado desde el backend nos dice que **no hemos encontrado ningún jugador con las características pedidas en la base de datos**, entonces mostraremos también esta cadena de texto de forma exclusiva.

Por último, si desde el backend **hemos recibido que sí que necesitamos hacer la recomendación**, lo que haremos será mostrarla al usuario de la siguiente forma:

RECOMENDACIÓN PORTERO	▼
RECOMENDACIÓN LATERAL IZQUIERDO	▼
RECOMENDACIÓN LATERAL DERECHO	▼
RECOMENDACIÓN CENTRAL	▼
RECOMENDACIÓN PIVOTE	▼
RECOMENDACIÓN MEDIOCENTRO	▼
RECOMENDACIÓN EXTREMO IZQUIERDO	▼
RECOMENDACIÓN EXTREMO DERECHO	▼
RECOMENDACIÓN DELANTERO	▼

Si abrimos algunos de los desplegables se pueden ver los distintos tipos de recomendación que encontramos:

**Como el delantero no encajaba en la táctica, el programa nos ha recomendado un jugador**, en este caso Diego Costa, y nos ha mostrado su información.

#### RECOMENDACIÓN DELANTERO

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Diego Costa - Edad: 30 - Valor: 39M - Equipo: AT MADRID

Velocidad: 15 - Resistencia: 13 - Fuerza: 16

Remate: 15 - Pase: 12 - Regate: 12

Marcaje: 5 - Entradas: 9 - Cabeza: 15

Decisiones: 15 - Colocación: 6 - Paradas: 1

En el caso de portero **si nos encaja**, por lo que no hemos tenido que hacer una recomendación:

#### RECOMENDACIÓN PORTERO

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

En la sección 5 de análisis de resultados se verán con mayor detalle otras opciones que se pueden dar en la aplicación.

En esta parte he mostrado todos los datos a partir de una formación de 4-3-3. Sin embargo, para darle más flexibilidad a los entrenadores a la hora de elegir a los jugadores, se han incluido otras formaciones que funcionan exactamente igual pero tienen un dibujo diferente con distintas posiciones. Por ejemplo, el 4-2-1-3 tiene el siguiente aspecto:



#### 4-Explicación algoritmo:

En mi aplicación no es que exista un sólo algoritmo, sino que para cada posición existe una fórmula para conocer en qué táctica encaja mejor cada jugador. Además, obviamente, también existe una fórmula más genérica que nos permite calcular la táctica recomendada para el equipo en global.

Las fórmulas generales son las siguientes (teniendo en cuenta que estamos utilizando la media de los 11 jugadores de nuestro equipo):

**Poseción Rápida** =  $0.06 \cdot \text{Velocidad} + 0.06 \cdot \text{Resistencia} + 0.06 \cdot \text{Fuerza} + 0.08 \cdot \text{Remate} + 0.15 \cdot \text{Pase} + 0.10 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.07 \cdot \text{Entradas} + 0.07 \cdot \text{Cabeza} + 0.1 \cdot \text{Decisiones} + 0.12 \cdot \text{Colocación} + 0.05 \cdot \text{Paradas}$ .

**Poseción lenta** =  $0.10 \cdot \text{Velocidad} + 0.08 \cdot \text{Resistencia} + 0.05 \cdot \text{Fuerza} + 0.10 \cdot \text{Remate} + 0.12 \cdot \text{Pase} + 0.10 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.07 \cdot \text{Entradas} + 0.05 \cdot \text{Cabeza} + 0.12 \cdot \text{Decisiones} + 0.08 \cdot \text{Colocación} + 0.05 \cdot \text{Paradas}$ .

**Gegenpressing** =  $0.09 \cdot \text{Velocidad} + 0.15 \cdot \text{Resistencia} + 0.07 \cdot \text{Fuerza} + 0.08 \cdot \text{Remate} + 0.08 \cdot \text{Pase} + 0.08 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.07 \cdot \text{Entradas} + 0.05 \cdot \text{Cabeza} + 0.08 \cdot \text{Decisiones} + 0.12 \cdot \text{Colocación} + 0.05 \cdot \text{Paradas}$ .

**Contraataques** =  $0.15 \cdot \text{Velocidad} + 0.07 \cdot \text{Resistencia} + 0.07 \cdot \text{Fuerza} + 0.08 \cdot \text{Remate} + 0.07 \cdot \text{Pase} + 0.09 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.08 \cdot \text{Entradas} + 0.05 \cdot \text{Cabeza} + 0.10 \cdot \text{Decisiones} + 0.10 \cdot \text{Colocación} + 0.06 \cdot \text{Paradas}$ .

**Ataque Por Bandas** =  $0.12 \cdot \text{Velocidad} + 0.05 \cdot \text{Resistencia} + 0.10 \cdot \text{Fuerza} + 0.08 \cdot \text{Remate} + 0.08 \cdot \text{Pase} + 0.10 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.08 \cdot \text{Entradas} + 0.10 \cdot \text{Cabeza} + 0.07 \cdot \text{Decisiones} + 0.08 \cdot \text{Colocación} + 0.06 \cdot \text{Paradas}$ .

**Catenaccio** =  $0.08 \cdot \text{Velocidad} + 0.07 \cdot \text{Resistencia} + 0.10 \cdot \text{Fuerza} + 0.07 \cdot \text{Remate} + 0.07 \cdot \text{Pase} + 0.08 \cdot \text{Regate} + 0.08 \cdot \text{Marcaje} + 0.08 \cdot \text{Entradas} + 0.08 \cdot \text{Cabeza} + 0.12 \cdot \text{Decisiones} + 0.10 \cdot \text{Colocación} + 0.06 \cdot \text{Paradas}$ .

Para cada una de las posiciones que existen hay 6 fórmulas que permiten calcular la adecuación de cada jugador en cada una de las tácticas. Así, por ejemplo en los defensas es más importante el atributo de entradas que el atributo de remate a portería ya que su principal objetivo es el de intentar robar la pelota al equipo enemigo.

Hay un caso particular que es el del portero, que tiene como características principal que hay atributos como por ejemplo el remate o el regate que no tiene ninguna importancia en el cálculo de la táctica ya que sólo en contadas excepciones va a tener que realizar este tipo de acciones, y su atributo principal es el de paradas.

## 5-Análisis de resultados.

En esta sección de la memoria se van a mostrar **diferentes ejecuciones del programa** para mostrar su funcionalidad y las diferentes opciones que este ofrece.

Al contrario que en otros sistemas de recomendación, en los que el usuario tiene control completo de qué quiere que se vaya a recomendar, en nuestro sistema de recomendación este proceso está bastante automatizado, ya que es el propio programa el que calcula la mejor táctica, por lo que es mucho más difícil conseguir ejecuciones muy distintas o que puedan presentar problemas, ya que la mayoría de las ejecuciones suelen resultar en recomendaciones de unos 5-6 jugadores y es difícil encontrar una distinta.

El primer ejemplo que voy a incluir es el que existe por defecto, es decir, **con todos los sliders con las estadísticas de los jugadores a 20**. Este es un ejemplo que nos permite mostrar tanto que en la mayoría de los casos vamos a encontrar que sólo tenemos que realizar unas pocas recomendaciones como que en la mayoría de los casos de ejecución sí que se van a encontrar jugadores en la base de datos y sólo van a ser unos pocos los casos en los que no lo consigamos.

Como se puede ver, la media de todos los atributos es obviamente de 20:

The screenshot shows a web interface with a dark blue background. It is divided into two main sections, each with a title and a set of input fields.

**Section 1: Lo primero que se va a mostrar son las medias de cada atributo:**

VEL	RES	FUE
20	20	20

REM	PAS	REG
20	20	20

MAR	ENT	CAB
20	20	20

DEC	COL	PAR
20	20	20

**Section 2: Ahora se van a desglosar las recomendaciones para cada una de las 6 estrategias posibles:**

POSESIÓN LENTA	POSESIÓN RÁPIDA	GEENIPRESSING
2.3230245374709986	2.3312367390625104	2.3165720933633818

POR BANDAS	CONTRAATAQUE	CATENACCIO
2.4537293100124318	2.4330500122986516	2.4360807057431373

También podemos ver que en las recomendaciones, la táctica que ha salido elegida es la estrategia por bandas. (La parte del texto explicativo ya la he mostrado en la parte de Frontend del punto 3)



Los únicos jugadores que no encajan con esta táctica son los dos extremos y el delantero, y por lo tanto el programa nos ha recomendado jugadores que **nos permiten sustituirlos**:

#### RECOMENDACIÓN PORTERO

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN LATERAL IZQUIERDO

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN LATERAL DERECHO

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN CENTRAL

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN PIVOTE

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN MEDIOCENTRO

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

#### RECOMENDACIÓN EXTREMO IZQUIERDO

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Eden Hazard - Edad: 28 - Valor: 82M - Equipo: REAL MADRID  
Velocidad: 16 - Resistencia: 17 - Fuerza: 11  
Remate: 15 - Pase: 15 - Regate: 20  
Marcaje: 3 - Entradas: 4 - Cabeza: 7  
Decisiones: 17 - Colocación: 7 - Paradas: 1

#### RECOMENDACIÓN EXTREMO DERECHO

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Nabil Fekir - Edad: 25 - Valor: 38M - Equipo: BETIS  
Velocidad: 15 - Resistencia: 14 - Fuerza: 14  
Remate: 15 - Pase: 16 - Regate: 16  
Marcaje: 5 - Entradas: 5 - Cabeza: 6  
Decisiones: 15 - Colocación: 8 - Paradas: 1

#### RECOMENDACIÓN DELANTERO

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Diego Costa - Edad: 30 - Valor: 39M - Equipo: AT MADRID  
Velocidad: 15 - Resistencia: 13 - Fuerza: 16  
Remate: 15 - Pase: 12 - Regate: 12  
Marcaje: 5 - Entradas: 9 - Cabeza: 15  
Decisiones: 15 - Colocación: 6 - Paradas: 1

El segundo ejemplo va a ser un caso en el que tenemos que **recomendar muchos jugadores** ya que, si bien sí que tenemos una táctica que canjara bastante bien para la mayoría, no es la mejor táctica para los jugadores y por lo tanto tenemos que conseguir jugadores que cumplan mejor los requisitos establecidos.

Un buen ejemplo es que, cuando ponemos en **todos los jugadores la velocidad a 1 y el resto de los atributos a 20**, se producen resultados más interesantes que en la anterior ejecución.

(Ejemplo del extremo izquierdo, todos son iguales)



La media de los jugadores como se puede ver es de 1 en velocidad y 20 en el resto:

Lo primero que se va a mostrar son las medias de cada atributo:

VEL: 1	RES: 20	FUE: 20
REM: 20	PAS: 20	REG: 20
MAR: 20	ENT: 20	CAB: 20
DEC: 20	COL: 20	PAR: 20

Ahora se van a desglosar las recomendaciones para cada una de las 6 estrategias posibles:

POSESIÓN LENTA: 2.191990054712378	POSESIÓN RÁPIDA: 2.252616049407338	GEGENPRESSING: 2.1986410588806233
POR BANDAS: 2.296487930702087	CONTRAATAQUE: 2.2364982881607203	CATENACCIO: 2.3312531195362407

La estrategia que nos ha recomendado esta vez es **catenaccio**, como se puede ver en las siguientes capturas:

Aquí se va a mostrar la estrategia recomendada:

RECOMENDACIÓN DE ESTRATEGIA

El estilo recomendado es catenaccio.

Esta es la explicación de la estrategia elegida:

EXPLICACIÓN DE LA ESTRATEGIA

En este estilo de juego, el principal objetivo es que a nuestro equipo no le lleguen a marcar ningún objetivo mediante el uso de una defensa férrea. En ataque, no se arriesgarán las posiciones de los jugadores defensivos, atacando con el menor número de jugadores posibles. En defensa, no se tomará ningún tipo de riesgo, depejando el balón siempre que sea necesario.

Si abrimos 3 jugadores al azar, vemos que algunos encajan bien, otros perfectamente y otros mal:

PORTERO

POSESIÓN LENTA Encaja perfectamente en esta táctica	POSESIÓN RÁPIDA Encaja bien en esta táctica	GEGENPRESSING Encaja perfectamente en esta táctica
POR BANDAS Encaja perfectamente en esta táctica	CONTRAATAQUE No encaja en esta táctica	CATENACCIO Encaja bien en esta táctica

LATERAL IZQUIERDO

POSESIÓN LENTA Encaja bien en esta táctica	POSESIÓN RÁPIDA No encaja en esta táctica	GEGENPRESSING No encaja en esta táctica
POR BANDAS No encaja en esta táctica	CONTRAATAQUE No encaja en esta táctica	CATENACCIO Encaja perfectamente en esta táctica

DEFENSA CENTRAL

POSESIÓN LENTA Encaja perfectamente en esta táctica	POSESIÓN RÁPIDA Encaja mal en esta táctica	GEGENPRESSING No encaja en esta táctica
POR BANDAS Encaja bien en esta táctica	CONTRAATAQUE No encaja en esta táctica	CATENACCIO No encaja en esta táctica

Esta vez vamos a pedir una **recomendación de jugadores** que se encuentren en el rango de valores de 4 a 9 millones.

☒ ¿Tener en cuenta el precio?

☐ 1-3 MILLONES
 ☒ 4-9 MILLONES
 ☐ 10-25 MILLONES
 ☐ +25 MILLONES

Ahora, una vez que se pulse el botón de recomendación automática, se recomendarán jugadores que cubran las posiciones en los que el jugador no coincide con la táctica.


Ahora vamos a ver las recomendaciones de cada uno de los jugadores:

RECOMENDACIÓN PORTERO	^
La recomendación de jugador que encaja mejor en la táctica es la siguiente:	
Nombre: Marko Dmitrovic - Edad: 27 - Valor: 7M - Equipo: EIBAR	
Velocidad: 10 - Resistencia: 12 - Fuerza: 11	
Remate: 5 - Pase: 3 - Regate: 8	
Marcaje: 8 - Entradas: 8 - Cabeza: 9	
Decisiones: 14 - Colocación: 14 - Paradas: 15	
RECOMENDACIÓN LATERAL IZQUIERDO	^
El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.	
RECOMENDACIÓN LATERAL DERECHO	^
El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.	
RECOMENDACIÓN CENTRAL	^
La recomendación de jugador que encaja mejor en la táctica es la siguiente:	
Nombre: Zouhair Feddal - Edad: 30 - Valor: 5M - Equipo: BETIS	
Velocidad: 13 - Resistencia: 14 - Fuerza: 14	
Remate: 7 - Pase: 11 - Regate: 9	
Marcaje: 14 - Entradas: 14 - Cabeza: 13	
Decisiones: 13 - Colocación: 14 - Paradas: 1	
RECOMENDACIÓN PIVOTE	^
No se ha encontrado ningún elemento en la base de datos que mejore el resultado.	
RECOMENDACIÓN MEDIOCENTRO	^
La recomendación de jugador que encaja mejor en la táctica es la siguiente:	
Nombre: Recio - Edad: 28 - Valor: 6M - Equipo: LEGANÉS	
Velocidad: 8 - Resistencia: 15 - Fuerza: 13	
Remate: 11 - Pase: 13 - Regate: 14	
Marcaje: 14 - Entradas: 12 - Cabeza: 11	
Decisiones: 13 - Colocación: 14 - Paradas: 1	
RECOMENDACIÓN EXTREMO IZQUIERDO	^
La recomendación de jugador que encaja mejor en la táctica es la siguiente:	
Nombre: Nolito - Edad: 32 - Valor: 6M - Equipo: SEVILLA	
Velocidad: 12 - Resistencia: 14 - Fuerza: 10	
Remate: 14 - Pase: 13 - Regate: 15	
Marcaje: 4 - Entradas: 10 - Cabeza: 9	
Decisiones: 15 - Colocación: 10 - Paradas: 1	
RECOMENDACIÓN EXTREMO DERECHO	^
No se ha encontrado ningún elemento en la base de datos que mejore el resultado.	
RECOMENDACIÓN DELANTERO	^
La recomendación de jugador que encaja mejor en la táctica es la siguiente:	
Nombre: Joselu - Edad: 29 - Valor: 6M - Equipo: ALAVÉS	
Velocidad: 10 - Resistencia: 13 - Fuerza: 15	
Remate: 13 - Pase: 12 - Regate: 11	
Marcaje: 7 - Entradas: 10 - Cabeza: 15	
Decisiones: 13 - Colocación: 7 - Paradas: 1	

En este caso, en la mayoría de los jugadores que hemos encontrado que nos ha ofrecido una **recomendación adecuada a las características**, es decir, en la posición correcta y con el valor adecuado y que encajen perfectamente en la táctica que hemos pedido.

Sin embargo, en el caso del pivote **no hemos encontrado a ningún jugador en la base de datos que encaje con las características pedidas**. Esto se debe a que en la base de datos que he utilizado apenas hay pivotes (ni mediapuntas), por lo que al programa le cuesta encontrar los resultados pedidos.

```
neo4j$ match (n) where n.position = 'PI' return n
```

The image shows a Neo4j query result. On the left is a small icon of a person. To its right is a purple pill-shaped badge containing the text "(16)". Further right is an orange pill-shaped badge containing the text "Player(16)".

Si miramos en la base de datos, encontramos que sólo hay 16 resultados de más de 200 jugadores que se encuentran en la base de datos, por lo que no es fácil encontrar que coincida tanto el valor de mercado como la táctica preferencial.

Si por ejemplo cambiamos el rango de precios al rango de 10-25, podemos ver que en todos los casos en los que se debe realizar una recomendación funciona correctamente:

**RECOMENDACIÓN PORTERO** ^

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Fernando Pacheco - Edad: 27 - Valor: 12M - Equipo: ALAVÉS  
Velocidad: 12 - Resistencia: 10 - Fuerza: 14  
Remate: 1 - Pase: 13 - Regate: 1  
Marcaje: 3 - Entradas: 1 - Cabeza: 8  
Decisiones: 12 - Colocación: 13 - Paradas: 16

**RECOMENDACIÓN LATERAL IZQUIERDO** ^

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

**RECOMENDACIÓN LATERAL DERECHO** ^

El jugador encaja con la táctica, por lo que no es necesario ofrecer ninguna recomendación.

**RECOMENDACIÓN CENTRAL** ^

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Gabriel Paulista - Edad: 28 - Valor: 23M - Equipo: VALENCIA  
Velocidad: 15 - Resistencia: 15 - Fuerza: 15  
Remate: 8 - Pase: 12 - Regate: 10  
Marcaje: 15 - Entradas: 16 - Cabeza: 15  
Decisiones: 11 - Colocación: 15 - Paradas: 1



<p>RECOMENDACIÓN PIVOTE</p> <p>La recomendación de jugador que encaja mejor en la táctica es la siguiente:</p> <p>Nombre: David López - Edad: 29 - Valor: 24M - Equipo: ESPANYOL  Velocidad: 13 - Resistencia: 15 - Fuerza: 14  Remate: 8 - Pase: 12 - Regate: 9  Marcaje: 14 - Entradas: 15 - Cabeza: 14  Decisiones: 15 - Colocación: 15 - Paradas: 1</p>	^
<p>RECOMENDACIÓN MEDIOCENTRO</p> <p>La recomendación de jugador que encaja mejor en la táctica es la siguiente:</p> <p>Nombre: Dani García - Edad: 29 - Valor: 22M - Equipo: ATHLETIC  Velocidad: 13 - Resistencia: 15 - Fuerza: 14  Remate: 8 - Pase: 13 - Regate: 11  Marcaje: 14 - Entradas: 15 - Cabeza: 13  Decisiones: 15 - Colocación: 16 - Paradas: 1</p>	^
<p>RECOMENDACIÓN EXTREMO IZQUIERDO</p> <p>La recomendación de jugador que encaja mejor en la táctica es la siguiente:</p> <p>Nombre: Iñigo Córdoba - Edad: 22 - Valor: 14M - Equipo: ATHLETIC  Velocidad: 14 - Resistencia: 13 - Fuerza: 11  Remate: 12 - Pase: 12 - Regate: 14  Marcaje: 5 - Entradas: 7 - Cabeza: 8  Decisiones: 13 - Colocación: 9 - Paradas: 1</p>	^
<p>RECOMENDACIÓN EXTREMO DERECHO</p> <p>La recomendación de jugador que encaja mejor en la táctica es la siguiente:</p> <p>Nombre: Lucas Ocampos - Edad: 24 - Valor: 12M - Equipo: SEVILLA  Velocidad: 16 - Resistencia: 15 - Fuerza: 13  Remate: 13 - Pase: 12 - Regate: 15  Marcaje: 5 - Entradas: 7 - Cabeza: 13  Decisiones: 11 - Colocación: 7 - Paradas: 1</p>	^
<p>RECOMENDACIÓN DELANTERO</p> <p>La recomendación de jugador que encaja mejor en la táctica es la siguiente:</p> <p>Nombre: Raúl De Tomás - Edad: 24 - Valor: 10M - Equipo: ESPANYOL  Velocidad: 13 - Resistencia: 13 - Fuerza: 13  Remate: 13 - Pase: 12 - Regate: 13  Marcaje: 5 - Entradas: 8 - Cabeza: 14  Decisiones: 13 - Colocación: 8 - Paradas: 1</p>	^

Como tercera ejecución, podemos probar a meter **datos más realistas que futbolistas**, es decir, que los delanteros sean buenos rematadores y malos defensores, que los extremos sean muy rápidos o que los defensas sean muy fuertes.

**Vamos a probar con datos en los que los dos jugadores que juegan en la misma posición** (defensas centrales, mediocentros,...) **tengan estadísticas diferentes entre ellos** ya que es lo normal en el fútbol, intentar tener jugadores que se complementen entre ellos.

Cómo mostrar todos los sliders ocuparía mucho espacio en este documento, voy a mostrar directamente los cálculos que nos ofrece la aplicación.

Lo primero que se va a mostrar son las medias de cada atributo:

VEL 14	RES 13	FUE 13
REM 12	PAS 14	REG 14
MAR 11	ENT 11	CAB 11
DEC 15	COL 14	PAR 2

Ahora se van a desglosar las recomendaciones para cada una de las 6 estrategias posibles:

POSESIÓN LENTA 1.0947692679975802	POSESIÓN RÁPIDA 1.0978438164376458	GEGENPRESSING 1.0837236193933109
POR BANDAS 1.077182369717397	CONTRAATAQUE 1.0801002679111573	CATENACCIO 1.0748473943479389

Aquí se va a mostrar la estrategia recomendada:

RECOMENDACIÓN DE ESTRATEGIA  
El estilo recomendado es posesión rápida.

Esta es la explicación de la estrategia elegida:

EXPLICACIÓN DE LA ESTRATEGIA  
En el estilo de posesión rápida, si bien se intentará mantener el control del balón la mayoría de las veces, en los momentos en los que se pueda realizar un ataque rápido se realizará. La zona de juego normalmente será por el centro, si bien se utilizarán las bandas para dar profundidad al equipo. En ataque la mentalidad será ofensiva, subiendo a atacar la mayoría de los jugadores del equipo. En defensa, se realizará una presión intensa para recuperar el balón lo antes posible.

Vamos a mirar el caso de los **defensas centrales**:

DEFENSA CENTRAL

POSESIÓN LENTA Encaja bien en esta táctica	POSESIÓN RÁPIDA Encaja bien en esta táctica	GEGENPRESSING Encaja mal en esta táctica
POR BANDAS Encaja perfectamente en esta táctica	CONTRAATAQUE Encaja bien en esta táctica	CATENACCIO Encaja bien en esta táctica

DEFENSA CENTRAL

POSESIÓN LENTA Encaja perfectamente en esta táctica	POSESIÓN RÁPIDA Encaja perfectamente en esta táctica	GEGENPRESSING Encaja mal en esta táctica
POR BANDAS Encaja perfectamente en esta táctica	CONTRAATAQUE Encaja bien en esta táctica	CATENACCIO Encaja bien en esta táctica

Podemos ver que, si bien el segundo defensa central encaja bien en la táctica que nos ha recomendado el programa, el primer defensa central sólo encaja bien, por lo que necesitaremos ofrecer una recomendación de la base de datos.

### RECOMENDACIÓN CENTRAL

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Stefan Savic - Edad: 28 - Valor: 27M - Equipo: AT MADRID

Velocidad: 14 - Resistencia: 15 - Fuerza: 15

Remate: 9 - Pase: 12 - Regate: 9

Marcaje: 16 - Entradas: 14 - Cabeza: 14

Decisiones: 14 - Colocación: 15 - Paradas: 1

En este caso nos ha recomendado a Stefan Savic, que encaja mejor que el jugador que hemos introducido (utilicé el filtro de precio de +25 millones).

Podemos también mirar el caso de los **mediocentros**, que es similar a los defensas:

#### MEDIOCENTRO

POSESIÓN LENTA Encaja perfectamente en esta táctica	POSESIÓN RÁPIDA Encaja perfectamente en esta táctica	GEGENPRESSING Encaja bien en esta táctica
POR BANDAS Encaja perfectamente en esta táctica	CONTRAATAQUE Encaja bien en esta táctica	CATENACCIO Encaja bien en esta táctica

#### MEDIOCENTRO

POSESIÓN LENTA Encaja mal en esta táctica	POSESIÓN RÁPIDA Encaja bien en esta táctica	GEGENPRESSING Encaja perfectamente en esta táctica
POR BANDAS Encaja bien en esta táctica	CONTRAATAQUE Encaja perfectamente en esta táctica	CATENACCIO Encaja bien en esta táctica

En este caso nos recomienda a Toni Kroos:

### RECOMENDACIÓN MEDIOCENTRO

La recomendación de jugador que encaja mejor en la táctica es la siguiente:

Nombre: Toni Kroos - Edad: 29 - Valor: 62M - Equipo: REAL MADRID

Velocidad: 11 - Resistencia: 13 - Fuerza: 14

Remate: 13 - Pase: 20 - Regate: 14

Marcaje: 8 - Entradas: 9 - Cabeza: 7

Decisiones: 17 - Colocación: 12 - Paradas: 1

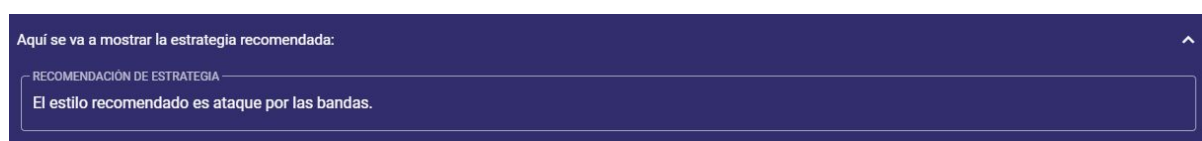
Vamos a mirar el **cuarto ejemplo**, en el cual en vez de trabajar con la formación 4-3-3 vamos a utilizar **4-2-1-3**, que incluye una nueva posición que hasta ahora no hemos utilizado, **el mediapunta**.

En este caso simplemente voy a forzar al programa a recomendar **un jugador que juegue en la posición de mediapunta** para comprobar los casos que se nos pueden dar en nuestro programa.

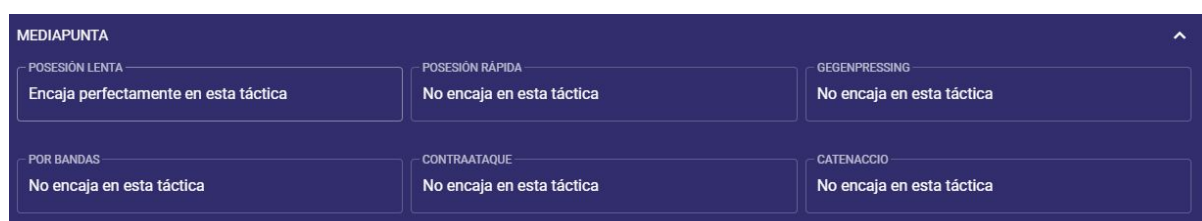
Para ello por ejemplo podemos poner todas las estadísticas a 1 del mediapunta y a 1 del resto:



La táctica del equipo sigue siendo por bandas (es la que sale por defecto y en el resto de jugadores no hemos tocado nada):



Por lo que el programa nos tendrá que recomendar un nuevo jugador, ya que como se puede ver el único estilo en el que encaja el que hemos introducido es en posesión lenta:



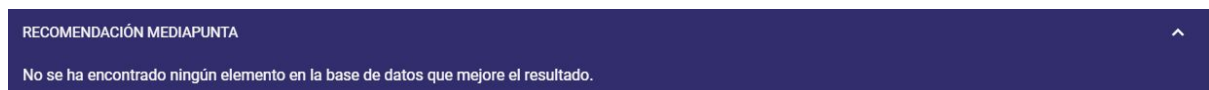
Como ya había expuesto anteriormente, no hay muchos pivotes ni mediapuntas, por lo que igual nos resulta más difícil encontrar a un jugador que cumpla las características en la base de datos.

Si por ejemplo probamos con el siguiente rango de valores:



A horizontal bar with a dark blue background containing four radio button options for player value ranges: "1-3 MILLONES", "4-9 MILLONES", "10-25 MILLONES", and "+25 MILLONES". The "4-9 MILLONES" option is selected, indicated by a yellow dot in the center of the radio button.

podemos comprobar que no encontramos a ningún jugador adecuado:



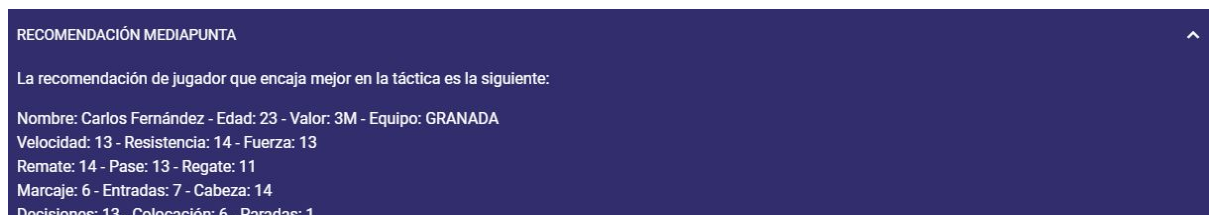
A dark blue rectangular box with the title "RECOMENDACIÓN MEDIAPUNTA" and a small upward arrow icon on the right. The text inside reads: "No se ha encontrado ningún elemento en la base de datos que mejore el resultado."

Sin embargo, si refinamos la búsqueda y probamos con jugadores en el rango más barato que existe en la aplicación:



A horizontal bar with a dark blue background containing four radio button options for player value ranges: "1-3 MILLONES", "4-9 MILLONES", "10-25 MILLONES", and "+25 MILLONES". The "1-3 MILLONES" option is selected, indicated by a red dot in the center of the radio button.

ya nos realizar una recomendación adecuada a lo que hemos pedido:



A dark blue rectangular box with the title "RECOMENDACIÓN MEDIAPUNTA" and a small upward arrow icon on the right. The text inside reads: "La recomendación de jugador que encaja mejor en la táctica es la siguiente:" followed by a list of player statistics: "Nombre: Carlos Fernández - Edad: 23 - Valor: 3M - Equipo: GRANADA", "Velocidad: 13 - Resistencia: 14 - Fuerza: 13", "Remate: 14 - Pase: 13 - Regate: 11", "Marcaje: 6 - Entradas: 7 - Cabeza: 14", and "Decisiones: 13 - Colocación: 6 - Paradas: 1".

Como se puede ver en los sliders, los campos de **Nombre, Apellido y Edad** no los he modificado en ningún momento ya que no son necesarios para esta aplicación. La razón es simplemente para que si se diera el caso de que quisiera ampliar la aplicación para que tuviera en cuenta características como la edad, entonces estos campos se podrían utilizar, y también se podrían utilizar para poder añadir nuevos jugadores a la base de datos.

En la sección de **líneas de futuro** se pueden ver con más detalle este tipo de apreciaciones.



## 6-DAFO:

El análisis DAFO consiste en buscar las debilidades, las amenazas, las fortalezas y las oportunidades del programa para realizar un análisis crítico de este.

### Debilidades:

1-Debido a la gran cantidad de datos que se deben de introducir en el programa, si bien la solución de utilizar múltiples sliders es bastante efectiva, **para el usuario puede resultar muy tedioso introducir todos los datos en una única sección.**

2-Si bien el diseño de la aplicación es funcional, probablemente se podría **mejorar bastante su aspecto** mediante un mejor uso de la paleta de colores o de unos componentes más atractivos.

3-En algunos casos la **base de datos se me ha quedado un poco corta** aun habiendo introducido más de 200 nodos debido a que a veces no es capaz de encontrar ningún jugador que cumpla con las características pedidas.

4-Debido a la gran cantidad de datos que se ofrecen en la aplicación, un usuario que entre por primera vez posiblemente se pierda navegando en los menús o se **sienta abrumado por la gran cantidad de información que se muestra.**

### Amenazas:

1-Debido a estar utilizando una aplicación con una nueva tecnología, inicialmente tenía cierta **incertidumbre sobre si podría aplicar mi idea inicial** utilizando este tipo de bases de datos, que al final ha sido un éxito.

2-Otro de los problemas que se podrían encontrar es el hecho de que al tener que unir tantas piezas muchas veces se **producen errores que pueden ralentizar el progreso de la aplicación** durante varios días.

3-Una vez terminada la aplicación, otra de las amenazas que nos pueden surgir es el hecho de que, si bien este tipo de sistemas aún no están muy extendidos, **los equipos de fútbol cada vez más están invirtiendo en estos sistemas y por lo tanto podemos perder una ventaja competitiva.**

### Fortalezas:

1-Es una aplicación **relativamente novedosa con respecto a otros tipos de sistemas**, ya que es a la vez un sistema de recomendación de jugadores y un sistema de cálculo y obtención de tácticas.

2-Los sistemas que se están utilizando están en auge, por lo que, si en algún momento decido desarrollar más adelante la aplicación, **posiblemente los sistemas de programación estarán más desarrollados**.

3-La **velocidad de ejecución en relación con la cantidad de datos que se van a guardar es bastante rápida**, por lo que la experiencia de usuario es buena en este aspecto.

### Oportunidades:

1-La oportunidad que estamos aprovechando en esta asignatura es la de poder **utilizar una nueva tecnología con neo4j** y las **bases de datos de grafos** que pueden ser de gran importancia en el futuro.

2-También estamos aprovechando el hecho de que **no existen aplicaciones similares** a la que estoy desarrollando para poder tener más oportunidades a nivel de mercado.

3-Otra de las oportunidades que hemos podido aprovechar es el hecho de que en este tipo de trabajos, si bien son individuales, tenemos la oportunidad de “**socializar conocimiento**”, es decir, hablar con los compañeros para poder solucionar mutuamente los problemas que nos puedan surgir.

4-También ha sido un buen momento para aprender a utilizar **los cuadernos de trabajo**, en los que hemos ido documentando las actualizaciones que estaba teniendo nuestro proyecto.



## 7-Líneas de futuro.

1-Si bien se ha utilizado una base de datos de Football Manager, lo ideal sería **utilizar datos de jugadores reales para poder calcular los atributos de los jugadores** y que el usuario no tuviera por qué adoptar el formato de atributos de 1 al 20. Así, se podrían calcular algunos atributos de la siguiente manera:

-En el caso de la velocidad, se podrían hacer pruebas de Sprint entre los jugadores para comprobar cuáles son los más rápidos y cuáles son los más lentos.

-En el caso de los pases, se podría comprobar el porcentaje de pases completados de cada uno de los jugadores para poder realizar comprobaciones entre ellos.

-En el caso de los remates a portería, se podrían calcular la cantidad de goles que están marcando los jugadores en comparación a los tiros a puerta que realizan.

También se podrían hacer **nuevas fórmulas basadas en estas estadísticas** en vez de los algoritmos utilizados actualmente.



2-Otra de las líneas de futuro que se podrían proseguir a partir de la aplicación que he desarrollado sería ampliar su funcionalidad a **otros ámbitos del fútbol**.

-Un buen ejemplo sería conseguir que, mediante los atributos del jugador y su edad, se creara una calculadora que permita **hallar el valor de mercado de un jugador**.

-Otra de las características que se podrían añadir sería la **comparación entre dos equipos**. Por un lado, se realizaría una comparación entre las dos estrategias que llevan los equipos, y por otro lado se realizarían comparaciones entre los jugadores de forma individual.

-Si dispusiéramos de suficientes datos, también se podría intentar realizar una página web que se dedicara exclusivamente a realizar una **funcionalidad de base de datos de jugadores de todo el mundo**, aunque esto también requeriría de poder coordinarnos con APIs externas y sería muy difícil.



3-Por último, tendríamos que considerar que esta aplicación se podría utilizar como **base para realizar otras de temáticas diferentes**, ya sea dentro del mundo del deporte (baloncesto, Fórmula 1, ..) o incluso fuera de este, ya que en muchos lugares nos podemos encontrar que los algoritmos de recomendación pueden llegar a ser muy útiles, sobre todo acompañado de sistemas de bases de datos con grafos como por ejemplo neo4j.



## 8-Lecciones aprendidas:

Una de las lecciones más importantes que he aprendido es que no sólo se deben utilizar las tecnologías que ya se han utilizado una y mil veces, sino que es una buena idea de vez en cuando **probar tecnologías en auge** que pueden tener un gran potencial como por ejemplo neo4j y los grafos de conocimiento, ya que si en algún momento este tipo de tecnología llega a explotar estaremos mucho más preparados que la mayoría de la gente para afrontar este desafío.

También me he dado cuenta de que hay **muchas formas de plantear un proyecto de una asignatura**, ya que en esta, al contrario que otras asignaturas en las que encontramos que normalmente hay una parte teórica y otra práctica, en SIBI comenzamos a trabajar desde el principio investigando individualmente, compartiendo los conocimientos con los compañeros y hablando con el profesor para que nos aconseje y nos guíe, lo cual nos da mucha libertad para realizar el trabajo a nuestro ritmo y eso se agradece en comparación a otras asignaturas más rígidas en las que todo el mundo tiene que acabar las mismas tareas a la vez.

Algo que también me puedo llevar es el uso de los **cuadernos de trabajo**, ya que son una excelente herramienta para poder organizar qué es lo que has realizado hasta ahora y qué te falta por realizar, además de recopilar los documentos más relevantes del trabajo que se está realizando.





## 9-Bibliografía:

Curso de neo4j preparatorio:

<https://neo4j.com/graphacademy/online-training/v4/00-intro-neo4j-about/>

Web de Neo4j:

<https://neo4j.com/>

Web de Vuetify:

<https://next.vuetifyjs.com/en/introduction/why-vuetify/>

Web de Node.js:

<https://nodejs.org/es/>

Web de Vue.js:

<https://vuejs.org/>

