



universidad
de león



Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Memoria Football Player Match



FootballPlayerMatch

Autor: Miguel Turrión Gutiérrez

01 de noviembre del 2021

ÍNDICE

Introducción	3
Descripción del problema	4
Herramientas de desarrollo	5
Descripción de la aplicación	12
General	12
Frontend	13
Backend	18
Base de datos	23
Explicación de los algoritmos	35
Análisis de los resultados	42
Ejemplo 1: Búsqueda de mediocentro con experiencia	42
Ejemplo 2: Búsqueda de portero sin ninguna especificación	44
Ejemplo 3: Recomendación de delanteros por búsquedas anteriores	46
Ejemplo 4: Recomendación de carrilero derecho sin búsquedas anteriores	47
Ejemplo 5: Búsqueda de defensa central por recomendación en perfil	48
DAFO	51
Problemas y soluciones en el desarrollo	53
Mejoras futuras	55
Lecciones aprendidas	56
Bibliografía y referencias	57

Introducción

En este documento se explicará de forma detallada todo lo relacionado con el desarrollo, funcionamiento, tecnologías que se utilizan, problemas que resuelve, posibles mejoras futuras, problemas que han ido apareciendo en el progreso de la aplicación y las soluciones que se han ido tomando, entre otra mucha información.

La aplicación es una página web que servirá como herramienta de ayuda a los diferentes equipos de fútbol para facilitar el trabajo de búsqueda de jugadores para su posible fichaje. La aplicación tendrá varias secciones, pero la más importante es la de búsqueda de jugadores, la cual permitirá al usuario hacer primero una búsqueda concisa del tipo de jugador que quiere, aplicando los diferentes filtros, una vez hecha esta búsqueda se podrá acceder al perfil de alguno de los jugadores en los que esté interesado y dentro del perfil de éste, la aplicación recomendará otros 3 jugadores similares que se asemejen al jugador que estamos viendo.

La aplicación está orientada a todo tipo de equipos de fútbol, ya sean equipos grandes, como podrían ser el Real Madrid Club de Fútbol o el Fútbol Club Barcelona que tienen grandes recursos económicos hasta equipos más modestos como podría ser el Levante Unión Deportiva o el Elche Club de Fútbol. También se tiene información sobre jugadores de casi todas las ligas profesionales de fútbol que hay en el mundo, por lo que se pueden realizar búsquedas para casi cualquier característica de un jugador.

Descripción del problema

El dinero que se invierte en fichajes es enorme y confundirse con el fichaje de un jugador puede ser un problema para cualquier equipo, pero sobre todo para los que no tengan grandes ingresos económicos, más ahora que debido a la pandemia muchos equipos de fútbol se vieron afectados, ya que su principal fuente de ingresos como los derechos televisivos, partidos de fútbol, fueron suspendidos e incluso hasta hace poco muchos partidos eran jugados a puerta cerrada. Por lo que es normal que estos equipos no inviertan en contratar ojeadores para buscar jugadores específicos

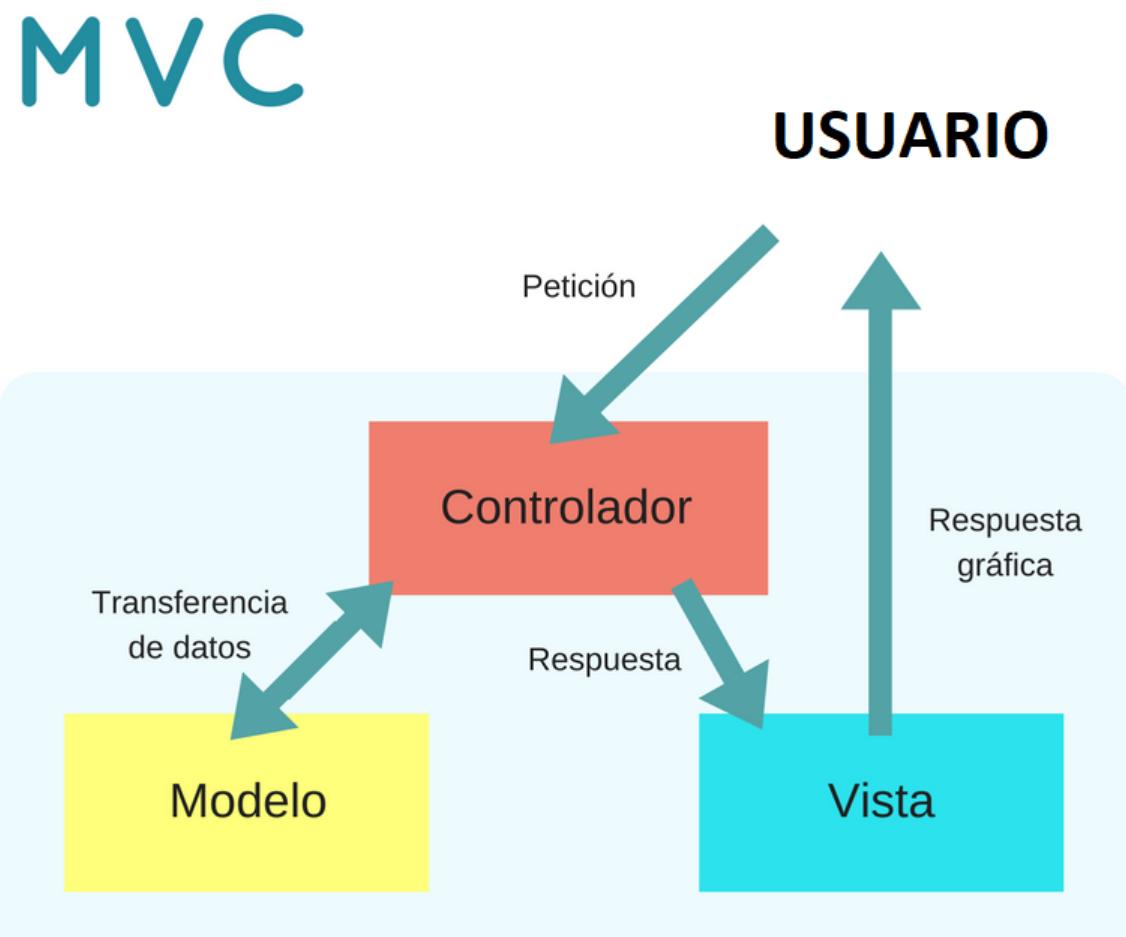
Otro factor es la cantidad de jugadores y ligas profesionales que hay en el mundo, seguir la pista a un jugador se vuelve una tarea difícil, más si éste no es un jugador muy popular.

Estas son las razones por las que decidí crear un sistema de recomendación para fichajes, y que los equipos de fútbol puedan utilizarlo de manera simple y sin casi gastos económicos. La aplicación mediante una selección de varios filtros, y un par de clicks no sólo mostrará los jugadores que están dentro de esos rangos, sino que también te recomendará jugadores similares basándose en tus búsquedas anteriores o jugadores similares basándose en el jugador que estés ojeando en ese mismo momento, permitiendo así al usuario ver una gran cantidad de jugadores similares en un breve periodo de tiempo.



Herramientas de desarrollo

El desarrollo se ha orientado a hacer una aplicación web, que permita la conexión a la misma desde cualquier dispositivo ya sea móvil u ordenador, la aplicación sigue el patrón de diseño de **Modelo-Vista-Controlador** conocido como Model-View-Controller en inglés (MVC), por el que cada componente tendrá la siguiente funcionalidad.



- **Vista:** es la parte visual de la aplicación donde se mostrará la información que se solicite.
- **Controlador:** es donde está la lógica del sistema, se reciben las peticiones del usuario desde la vista, se analizan y se ejecuta las diferentes funciones para enviar una respuesta.
- **Modelo:** es donde se almacena toda la información de la aplicación, y recibirá peticiones desde el controlador según la información solicitada.

En este caso la vista corresponde con el frontend, el controlador sería el backend y el modelo es la base de datos.

Todo el desarrollo se ha hecho bajo un ordenador con distribución linux, más concretamente con ubuntu (**versión 20.04**) que es un sistema operativo de software libre y código abierto, y las herramientas para programadores que ofrece este tipo de sistema operativo frente a otros, facilitan mucho el trabajo de desarrollo.

Y por último para la creación de la propia aplicación se han utilizado las siguientes herramientas:

Para el **backend** se ha utilizado **Node.js**, es una tecnología con la que estoy familiarizado y me ha dado buenos resultados en otros trabajos realizados, para la elección de ésta no sólo me he basado en la experiencia de trabajos anteriores sino que también tiene otras características como la facilidad para crear un nuevo proyecto, una gran sencillez y rapidez a la hora de crear la comunicación con el frontend ya que está optimizado para realizar esta tarea y por último y no menos importante, tiene compatibilidad con la tecnología que vamos a utilizar en la base de datos, que mediante la instalación de un driver ya tendremos a nuestra disposición todas las funcionalidades que tiene esta tecnología.



Para iniciar un nuevo proyecto mediante Node.js, lo único que tendríamos que hacer es descargarlo desde la página oficial, una vez descargado lo instalamos y por último, utilizaremos en un terminal el comando:

```
miguel@miguel-Aspire-V3-571G:~$ npm init
```

Después, nos pedirá diferente información relacionada con nuestro proyecto y una vez acabado, ya podemos configurarlo para que actúe como un **servidor web** e iniciararlo mediante el siguiente comando:

```
miguel@miguel-Aspire-V3-571G:~$ npm start
```

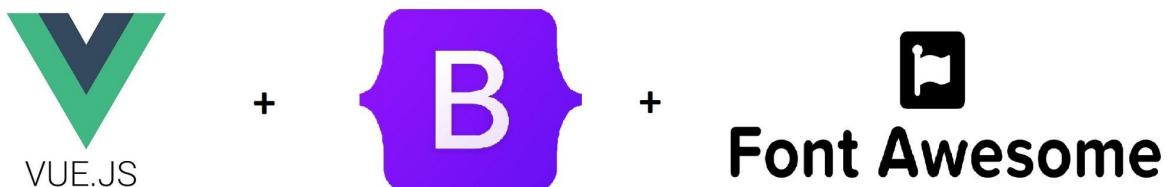
Aquí ya podemos añadir a nuestro proyecto **diferentes módulos** para que nos ayuden en el desarrollo de la aplicación, mediante el comando:

```
miguel@miguel-Aspire-V3-571G:~$ npm install nombre-modulo
```

En mi caso se han utilizado los siguientes módulos:

- **Nodemon**, es una utilidad que monitorea los cambios en el código que se está desarrollando y reinicia el servidor automáticamente.
- **Cors**, nos permite el intercambio de recursos entre los diferentes componentes, en nuestro caso entre el frontend y el backend.
- **Express**, nos permite estructurar la aplicación de una manera ágil y nos proporciona otras muchas funcionalidades como gestión de sesiones, cookies...
- **Neo4j-driver**, nos permite comunicarnos desde el backend a la base de datos, abriendo una sesión para realizar consultas.

En el área del **frontend** se ha utilizado **Vue.js**, que es un framework que conocía pero nunca lo había usado en ningún proyecto, por lo que me he animado a aprender un poco más sobre éste y usarlo para desarrollar la parte visual de la aplicación web. Como apoyo para dar una mayor calidad a los elementos visuales se ha utilizado **Bootstrap** que es un framework de css que me permitirá crear los estilos de los elementos de la página de una forma sencilla y también se usará **FontAwesome** que es un conjunto de iconos que me servirá para decorar la aplicación.



Al iniciar Vue en nuestro proyecto viene con lo básico para que puedas elegir qué es lo que quieras y lo que no y modeles el proyecto a tu gusto, esto es una de las características que más me ha gustado de usar Vue, también trabaja con componentes lo que es muy útil si un mismo elemento lo queremos mostrar en varias vistas de nuestra página web con crearlo una vez y añadirlo a estas se mostrará de forma correcta en las que lo necesitemos. Aparte de todo esto también es sencillo de aprender y con una sintaxis no muy compleja.

Para crear nuestro proyecto con Vue tenemos que utilizar el siguiente comando:

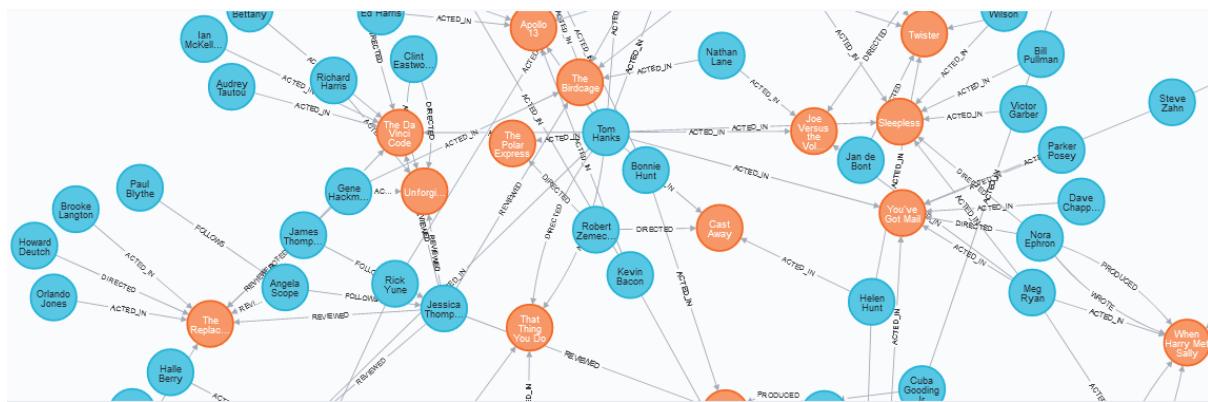
```
miguel@miguel-Aspire-V3-571G:~$ vue create nombre-del-proyecto
```

Seleccionamos la configuración que deseemos y ya podemos empezar a trabajar.

Por último la base de datos, en la cual se utilizará la tecnología **Neo4j**, es una base de datos orientada a grafos, por lo que podremos ver cómo está almacenada la información de una manera fácil y detallada. Neo4j utiliza **Cypher** como lenguaje propio para realizar consultas, es similar a SQL pero con mucho más potencial, es bastante intuitivo y por lo tanto fácil de utilizar.



Así se vería la representación de la información de una base de datos creada mediante Neo4j.



Los círculos representan los diferentes nodos que existen y las flechas que los unen las relaciones que hay entre ellos, tanto los nodos como las relaciones pueden tener propiedades lo que es muy útil para tener una mayor y mejor información de toda nuestra base de datos.

En mi caso al estar utilizado ubuntu como sistema operativo para poder acceder a trabajar con la base de datos necesitaremos abrir un terminal de cypher mediante el comando:

```
miguel@miguel-Aspire-V3-571G:~$ cypher-shell
```

Esto nos pedirá una verificación de usuario y contraseña que debimos indicar durante la instalación.

Una vez dentro de la consola podemos escribir `:help` para conocer la lista de comandos disponibles que podemos utilizar desde aquí, siendo `:exit` uno de los más importantes ya que nos permitirá salir. También podemos escribir las diferentes consultas que queramos hacer a la base de datos, todas ellas deberán terminar con un punto y coma para que sean consideradas.

La aplicación quedaría definida así con las tecnologías y componentes explicados anteriormente.



Se ha utilizado también en el desarrollo un sistema de control de versiones que es **GitHub**, en donde se ha creado un proyecto y se ha ido actualizando semana a semana durante todo el proceso de creación de la aplicación web.



GitHub permite almacenar tu proyecto con todo tipo de archivos, también facilita para que varias personas trabajen en un mismo proyecto y su utilización es muy sencilla.

Para empezar a utilizar GitHub lo primero que debemos es clonar el proyecto en la carpeta donde lo tengamos mediante el comando:

```
miguel@miguel-Aspire-V3-571G:~$ git clone repositorio
```

Una vez hecho esto ya podemos empezar a modificar los archivos y una vez que queramos actualizar nuestro repositorio con los cambios pertinentes utilizaremos la siguiente secuencia de comandos:

```
miguel@miguel-Aspire-V3-571G:~$ git add nombre-archivo
```

Aquí indicamos que archivos queremos que se actualicen.

```
miguel@miguel-Aspire-V3-571G:~$ git commit -m "mensaje"
```

El mensaje que aparecerá cuando subamos los archivos, aquí escribiremos de forma resumida qué es lo que hemos hecho.

```
miguel@miguel-Aspire-V3-571G:~$ git push
```

Para que los cambios se suban al repositorio, nos pedirá verificación de nuestros datos para ello.

Otro comando importante para cuando estemos trabajando en algún proyecto en grupo es:

```
miguel@miguel-Aspire-V3-571G:~$ git pull
```

Para tener la aplicación actualizada con los cambios que se hayan hecho y empezar a trabajar.

Descripción de la aplicación

General



Como se puede observar en la imagen, la aplicación está compuesta por 3 apartados que están disponibles en la barra de navegación en todo momento para que el usuario pueda moverse a través de las secciones como desee. En esta barra de navegación también se verá el logo de la aplicación a la izquierda.

- La página de **inicio**, se mostrará una imagen con un texto explicando de forma breve las diferentes características que tiene la aplicación.
- La página de **búsqueda de jugadores**, aquí el usuario podrá empezar a usar la aplicación como tal, esta página contará con diferentes filtros que el usuario puede seleccionar al gusto y ver qué jugadores están dentro de estos pulsando en el botón de buscar. Se explicará detalladamente todo su funcionamiento más adelante.
- La página de **perfil del usuario**, aquí el usuario podrá comprobar que jugadores tiene guardados como favoritos mediante el botón de mostrar favoritos y también podrá eliminar los jugadores que tiene como favoritos y las visitas a los perfiles de los jugadores para reiniciar las recomendaciones y que haga diferentes basándose en las nuevas búsquedas que haga el usuario.

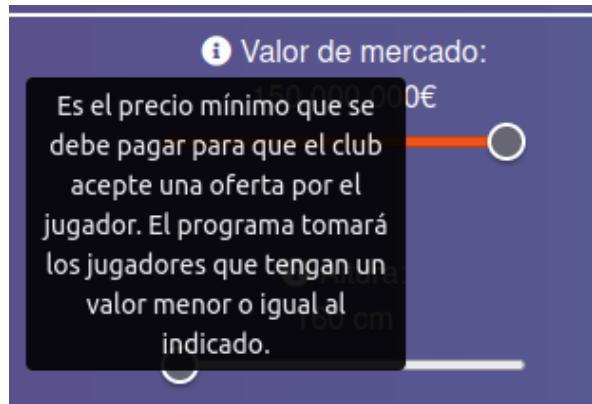
Y por último, el pie de página que mostrará las diferentes redes sociales que cuenta la aplicación y el copyright, como se puede ver en la siguiente imagen.



Frontend

Aquí nos vamos a centrar en el área de **búsqueda de jugadores** que es donde se realiza la mayoría de acciones de la aplicación, en esta sección lo primero que nos encontramos es la selección de diferentes filtros, los cuales nos ayudarán a buscar el jugador que queramos.

Cada filtro tiene a su lado un ícono con una *i* que contiene una breve explicación del mismo, como podemos ver en la siguiente imagen:



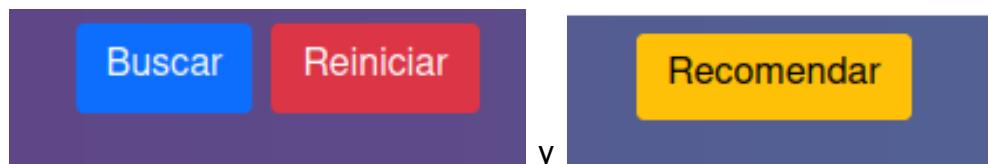
El único filtro obligatorio que debemos seleccionar es el de la posición, todos los demás son opcionales que podemos modificar para obtener mejores resultados.

La aplicación consta de los siguientes filtros:

- **Posición** donde juega el jugador que estamos buscando.
- **Valor de mercado** está comprendido entre 0 y 150.000.000 euros.
- **Sueldo** está comprendido entre 0 y 600.000 euros.
- **Edad** está comprendida entre 18 y 45 años.
- **Altura** está comprendida entre 160 y 220 cm.
- **Pierna** que puede estar sin definir, zurdo, diestro o ambidiestro.
- **Nacionalidad** puede estar sin definir o seleccionar una de las nacionalidades que existen.
- **Las demás características** pueden estar en estado sin definir, bajo, medio y alto, medio y alto o alto.
Sin definir corresponde al valor 0.
Bajo, medio y alto corresponde al valor 25.
Medio y alto corresponde al valor 50.
Alto corresponde al valor 75.
El sistema buscará que la característica sea mayor o igual al valor seleccionado.

Los atributos de portero sólo aparecerán cuando la posición seleccionada en el filtro de posición sea la de portero.

Después de los filtros encontraremos 3 botones.



- **Buscar**, nos permitirá buscar los jugadores según los filtros que hayamos aplicado.
- **Reiniciar**, para reiniciar todos los filtros, se colocarán por defecto.
- **Recomendar**, después de utilizar la aplicación varias veces y observar diferentes jugadores la aplicación podrá hacer una recomendación al usuario basándose en sus búsquedas anteriores, el funcionamiento se explicará más adelante en el apartado de explicación de algoritmos.

Si hemos dado al botón de buscar, a continuación de los botones nos aparecerá una tabla con los resultados obtenidos de la misma. Esta tabla mostrará por cada página 7 jugadores y podemos filtrar de mayor a menor según los títulos de la tabla.

Resultados totales: 116													
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial		
Jan Oblak	28	188	Atlético Madrid	Slovenia	75,000,000€	125,000€	2023	159,400,000€	Diestro	91	93		
Marc-André ter Stegen	29	187	FC Barcelona	Germany	69,500,000€	260,000€	2022	147,700,000€	Ambidestro	90	93		
Alisson Ramsés Becker	29	191	Liverpool	Brazil	62,500,000€	160,000€	2024	120,300,000€	Diestro	90	91		
Thibaut Courtois	29	199	Real Madrid	Belgium	56,000,000€	250,000€	2024	119,000,000€	Zurdo	89	90		
Manuel Neuer	35	193	FC Bayern München	Germany	29,000,000€	125,000€	2023	47,900,000€	Ambidestro	89	89		
Ederson Santana de Moraes	28	188	Manchester City	Brazil	53,500,000€	195,000€	2024	103,000,000€	Zurdo	88	91		
Samir Handanović	37	193	Inter	Slovenia	16,000,000€	100,000€	2021	26,400,000€	Diestro	88	88		
« < < 1 2 3 4 ... > »													

Por ejemplo si queremos que aparezcan primero los de menor edad pulsamos en el título de edad y se reorganizará la tabla de forma automática, mostrando los resultados de una forma fácil y sencilla.

Resultados totales: 116												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Gianluigi Donnarumma	22	196	Milan	Italy	41,500,000€	34,000€	2021	78,900,000€	Diestro	85	92	
Luís Manuel Arantes Maximiano	22	190	Sporting CP	Portugal	13,500,000€	7,000€	2023	29,700,000€	Diestro	78	88	
Alban Lafont	22	193	FC Nantes	France	11,000,000€	14,000€	2021	0€	Diestro	78	84	
Dean Henderson	24	190	Manchester United	England	16,000,000€	70,000€	2025	33,600,000€	Diestro	80	87	
Unai Simón Mendibil	24	190	Athletic Club de Bilbao	Spain	15,500,000€	18,000€	2025	35,700,000€	Diestro	80	86	
Alex Meret	24	190	Napoli	Italy	12,500,000€	36,000€	2023	23,800,000€	Zurdo	78	87	
Paul Bernardoni	24	190	Angers SCO	France	10,500,000€	16,000€	2024	23,100,000€	Zurdo	78	84	

« < 1 2 3 4 ... > »

Si pulsamos de nuevo en la edad nos colocará a los jugadores de mayor a menor.

Resultados totales: 116												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Gianluigi Buffon	43	192	Juventus	Italy	2,200,000€	41,000€	2021	3,600,000€	Diestro	82	82	
José Manuel Reina Páez	39	188	Lazio	Spain	1,300,000€	30,000€	2023	2,200,000€	Diestro	79	79	
Ben Foster	38	193	Watford	England	1,300,000€	12,000€	2022	2,300,000€	Zurdo	79	79	
Antonio Mirante	38	193	Roma	Italy	1,700,000€	24,000€	2021	2,900,000€	Diestro	78	78	
Samir Handanović	37	193	Inter	Slovenia	16,000,000€	100,000€	2021	26,400,000€	Diestro	88	88	
Miguel Angel Moyá Rumbo	37	188	Real Sociedad	Spain	1,700,000€	15,000€	2021	3,600,000€	Diestro	78	78	
Steve Mandanda	36	185	Olympique de Marseille	France	7,000,000€	28,000€	2024	13,300,000€	Diestro	83	83	

« < 1 2 3 4 ... > »

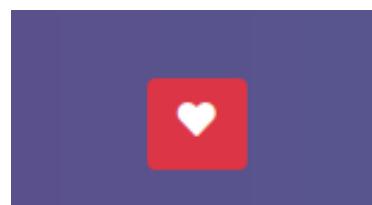
Esto pasará cuando pulsemos en cualquier título de la tabla, que nos colocará a los jugadores de forma que hayamos seleccionado así el usuario puede verlo a su gusto.

Si pulsamos dentro de la tabla en algún jugador, nos llevará al perfil del mismo donde el usuario podrá ver información más detallada del mismo.

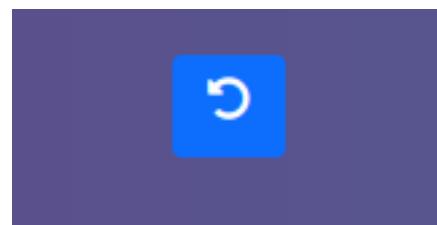
The screenshot shows a player profile for Gianluigi Donnarumma. At the top is a placeholder profile picture. Below it, the player's name is displayed as "Gianluigi Donnarumma (Italy)". Underneath the name are the birth date "1999-02-25 (22 años)" and height "196 cm". A red heart-shaped button is located below this information. The profile is divided into three main sections: "Milan" (team), "Habilidad" (Ability), and "Mentalidad" (Mental). The "Milan" section contains financial details: "Sueldo: 34,000€ / mes", "Contrato hasta: 2021", "Clausula de rescisión: 78,900,000€", and "Valor del mercado: 41,500,000€". The "Habilidad" section includes "Diestro" (Right-footed), "Habilidad general: 85 / 100", "Potencial: 92 / 100", and a yellow star icon followed by the text "★ Potencial de futuro". The "Mentalidad" section displays five green progress bars with numerical values: "Agresividad" (Aggression) at 30, "Posición" (Positioning) at 14, "Compostura" (Composure) at 68, "Penaltis" (Penalties) at 24, and another unnamed bar at 68.

Aquí veremos todas las características del jugador y sus atributos definidos de manera cuantitativa, con valores comprendidos entre el 0 y el 100.

Mediante este botón con forma de corazón podremos añadir al jugador como favorito.



Si seguimos bajando veremos este botón, que nos permitirá volver a la pantalla de búsqueda de jugadores para seguir buscando si no nos convence éste o los jugadores similares que nos ha recomendado el sistema.



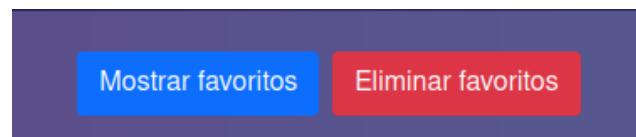
Por último los jugadores recomendados del sistema, aparecerán con una breve descripción y un botón de visitar perfil para poder ir directamente al perfil de estos.

Jugadores similares:

The screenshot shows a dark blue header with the text "Jugadores similares:". Below it, there are three player cards. Each card has a small placeholder profile picture, the player's name, position, birthdate, and height. At the bottom of each card is a red "Ver perfil" button.

Nombre	Posición	Nacionalidad
Thomas Strakosha (Albania)	GK	Bernd Leno (Germany)
1995-03-19 (26 años)		1992-03-04 (29 años)
193 cm		190 cm

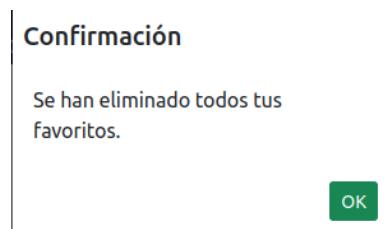
También tenemos el perfil del propio usuario, cuando accedemos al mismo nos encontraremos con dos botones.



El botón de mostrar favoritos hará que nos aparezca una tabla debajo de los botones con los jugadores que hemos ido añadiendo a esta lista.

Nombre	Posiciones	Equipo	Nacionalidad
Gianluigi Donnarumma	GK	Milan	Italy

Y el botón de eliminar favoritos hará que se eliminen todos los jugadores que hemos añadido.



Backend

Como se ha explicado anteriormente el backend es donde está la lógica de la aplicación y aquí se realizan las operaciones que son requeridas desde el frontend, tendremos diferentes rutas que son las que realizarán según qué función dependiendo de la ruta que sea, por lo que se explicará qué es lo que solicita y qué devuelve cada una.

Se ha configurado para que el backend esté escuchando las peticiones que se envían desde el frontend en el **puerto 3000** como podemos ver en la siguiente imagen:

```
// Configuraciones  
  
app.set('port', 3000);
```

Y aquí configuramos para que escuche en el puerto indicado.

```
// INICIAR EL SERVIDOR  
app.listen(app.get('port'), () => {  
    console.log('Servidor abierto en puerto', app.get('port'));  
});
```

Por lo que todas las solicitudes tendrán que estar dirigidas hacia este puerto, para que sean recibidas correctamente y se realicen las diferentes operaciones.

Todas las llamadas que se realizan desde el frontend están utilizando el método **POST**, este método permite enviar la información sin que el usuario la vea, ya que no se muestra en la URL como con el método GET sino que se almacena en el cuerpo de la petición.

Por lo que en la aplicación tenemos diferentes rutas, todas con este método pero cada una con funciones diferentes como veremos a continuación, aunque la estructura en todos es muy parecida.

Tenemos la ruta de **/buscarJugador**, ésta es llamada cuando se pulsa el botón de buscar en la sección de Búsqueda de jugadores.

```
// DIRECCIÓN PARA BUSCAR UN JUGADOR SEGÚN LA SELECCIÓN EN LOS FILTROS
app.post('/buscarJugador', (req, res) => {

    // LEEMOS LOS VALORES DE LA PETICIÓN

    var posicion = req.body.posicion;
    var edad = utils.calcularYearNacimiento(req.body.edad);
    var altura = req.body.altura;
    var pierna = req.body.pierna;
    var nacionalidad = req.body.nacionalidad;
    var valor = req.body.valor * 1000000;
    var sueldo = req.body.sueldo * 1000;

    var finalizacion = req.body.finalizacion;
    var regates = req.body.regates;
    var remate = req.body.remate;
    var tiroFalta = req.body.tiroFalta;
    var pasesCortos = req.body.pasesCortos;
    var pasesLargos = req.body.pasesLargos;
    var control = req.body.control;
    var disparoLejano = req.body.disparoLejano;
    var marcaje = req.body.marcaje;

    var agresividad = req.body.agresividad;
```

Lo primero que vemos son las diferentes variables que corresponden a los diferentes filtros que se han seleccionado.

```
// CREAMOS LA PETICIÓN

var query = 'MATCH (j:Jugador)-[:JUEGA_COMO]->(p:Posicion {posicion: \''+posicion+'\'})';
query += 'WHERE ';

//EDAD
query += 'datetime(j.fechaNacimiento) > datetime(\''+edad+'\') ';

// ALTURA
query += 'AND j.altura >= '+altura' ';

// PIERNA
if(pierna != null) {
    query += 'AND j.pierna = \''+pierna+'\' ';
}

// NACIONALIDAD
if(nacionalidad != null) {
    query += 'AND j.nacionalidad = \''+nacionalidad+'\' ';
```

Luego creamos una petición para la base de datos que nos devuelvan los jugadores que entran dentro de los requisitos de estos filtros.

```
results.records.forEach(function(record) {
    var jugador = {
        // CARACTERÍSTICAS GENERALES DE LOS JUGADORES
        nombre: record._fields[0].properties.nombre,
        edad: utils.calcularEdad(record._fields[0].properties.fechaNacimiento),
        altura: record._fields[0].properties.altura,
        pierna: record._fields[0].properties.pierna,
        equipo: record._fields[0].properties.equipo,
        nacionalidad: record._fields[0].properties.nacionalidad,
        valor: utils.numeroConSeparador(record._fields[0].properties.valor) + '€',
        sueldo: utils.numeroConSeparador(record._fields[0].properties.sueldo) + '€',
        clausula: utils.numeroConSeparador(record._fields[0].properties.clausula) + '€',
        contrato: record._fields[0].properties.contrato,
        habilidad: record._fields[0].properties.general,
        potencial: record._fields[0].properties.potencial
    };
    jugadores.push(jugador);
})
// DEVUELVE EL ARRAY PARA MOSTRARLO
res.json(jugadores);
```

Por último recogemos y enviamos la información de estos jugadores para que el usuario pueda verla.

Otra ruta que tenemos es **/recomendarSimilares**, que entra en juego cuando le damos al botón de recomendar en la sección de Búsqueda de jugadores. La estructura es muy parecida a la ruta anterior, primero obtenemos las variables para configurar la petición y luego devolvemos la lista de jugadores. Como dentro de esta ruta está uno de los algoritmos de recomendación de la aplicación se explicará con más detalle posteriormente en el área de **Explicación de algoritmos**.

Ruta de **/infoJugador**, esta ruta es llamada cuando seleccionamos un jugador para ver su perfil.

```
// DIRECCIÓN DONDE SE DEVOLVERÁ INFORMACIÓN MÁS ESPECÍFICA DEL JUGADOR
app.post('/infoJugador', (req, res) => {
    // OBTENEMOS LOS VALORES DE LA PETICIÓN
    var jugador = req.body.nombre;
    var usuario = req.body.usuario;
    var posicion = req.body.posicion;
```

Lo primero que hacemos es crear unas variables que almacenarán el jugador que queremos ver el perfil, el nombre del usuario que va a ver el perfil y la posición que juega este jugador.

```
// BUSCAMOS EL USUARIO
var queryVisitaPerfil = 'MATCH (u:Usuario { nombre: \''+usuario+'\'} ) ';

// BUSCAMOS EL JUGADOR
queryVisitaPerfil += 'MATCH (j:Jugador { nombre: \''+jugador+'\'} ) ';

// CREAMOS RELACIÓN
queryVisitaPerfil += 'MERGE (u)-[v:VISITA_PERFIL]->(j) ';

// CREA LA RELACIÓN SI NO EXISTE
queryVisitaPerfil += 'ON CREATE SET v.veces = 1 ';

// ACTUALIZA LA RELACIÓN
queryVisitaPerfil += 'ON MATCH SET v.veces = v.veces + 1;'
```

Luego creamos una petición a la base de datos que creará una relación entre el usuario y el jugador con **VISITA_PERFIL**, si esta relación ya está creada se aumentará un contador de veces que se ha visitado el perfil de este jugador.

```
// CONEXION PARA VISITAR EL PERFIL DEL JUGADOR
var queryVerPerfil = 'MATCH (j:Jugador { nombre: \''+jugador+'\'} )-[JUEGA_COMO]->(p:Posicion) R
```

Una vez hecho esto, crearemos otra petición que nos devuelva la información del jugador que queremos ojar.

```
// JUGADORES SIMILARES

var queryRecomendados = 'MATCH (j1:Jugador {nombre: \''+jugador.nombre+'\'} )-[:TIENE]->(skill1)
queryRecomendados += 'WITH j1, collect(id(skill1)) AS j1Skill ';
queryRecomendados += 'MATCH (j2:Jugador)-[:JUEGA_COMO]->(:Posicion { posicion: \''+posicion+
queryRecomendados += 'MATCH (j2)-[:TIENE]->(skill2) WHERE j1 <-> j2 ';
queryRecomendados += 'WITH j2, gds.alpha.similarity.jaccard(j1Skill, collect(id(skill2))) AS similitud
queryRecomendados += 'RETURN j2 ORDER BY similitud DESC LIMIT 3;'
```

Y por último, crearemos otra petición buscando jugadores similares al que vamos a ojar, aquí utilizamos otro de los algoritmos de recomendación, por ello esta parte se explicará en el área de **Explicación de algoritmos**.

Ruta de **/borrarBusquedas**, esta ruta es llamada cuando pulsamos el botón de eliminar favoritos desde el perfil del usuario.

```
// DIRECCIÓN PARA ELIMINAR TODAS TUS BÚSQUEDAS Y FAVORITOS
app.post('/borrarBusquedas', (req, res) => {

    // OBTENEMOS LOS VALORES DE LA PETICIÓN
    var usuario = req.body.usuario;

    // BUSCAMOS LA RELACIÓN DEL USUARIO CON LOS PERFILES DE LOS JUGADORES
    var query = 'MATCH (u:Usuario { nombre: \''+usuario+'\'} )-[v:VISITA_PERFIL]->(j:Jugador) ';

    // BORRAMOS ESAS RELACIONES
    query += 'DELETE v';

    // SEGUIMOS USANDO EL USUARIO
    query += 'WITH u';

    // BUSCAMOS LA RELACIÓN DE FAVORITOS
    query += 'MATCH (u)-[l:LIKES]->(j:Jugador) ';

    // BORRAMOS ESAS RELACIONES
    query += 'DELETE l;'
```

Lo primero que hacemos es guardar en una variable el nombre del usuario del que vamos a borrar sus favoritos, después creamos la petición que hará que se eliminen todas las relaciones que relacionan al usuario con los jugadores de la forma **VISITA_PERFIL** y **LIKES**. Una vez eliminadas, respondemos al usuario que el trabajo se ha realizado correctamente.

Ruta de **/addFavorito**, esta ruta se llama cuando pulsamos el botón de mostrar favoritos desde el perfil del usuario.

```
💡 DIRECCIÓN CUANDO UN USUARIO MARCA COMO FAVORITO A UN JUGADOR
app.post('/addFavorito', (req, res) => {

    // OBTENEMOS LOS VALORES DE LA PETICIÓN
    var usuario = req.body.usuario;
    var jugador = req.body.jugador;

    // BUSCAMOS EL USUARIO
    var query = 'MATCH (u:Usuario { nombre: \''+usuario+'\'} ) ';

    // BUSCAMOS EL JUGADOR
    query += 'MATCH (j:Jugador { nombre: \''+jugador+'\'} ) ';

    // CREAMOS RELACIÓN
    query += 'MERGE (u)-[l:LIKES]->(j) RETURN l;'
```

Lo primero que hacemos es obtener el nombre del usuario y el jugador que se ha marcado como favorito, después creamos la petición para crear la relación entre estos como una relación de **LIKES**, cuando ésta está creada respondemos al usuario con que el jugador se ha añadido a favoritos correctamente.

Base de datos

La base de datos inicial, utilizada en la aplicación ha sido obtenida en la página de **kaggle** y pertenece al juego **FIFA 21**, basándose en los datos en el modo carrera que tiene el propio juego. Esta base de datos se compone de unos 20.000 jugadores con más 50 columnas de atributos para cada uno, de los cuales se han ido eliminando o manteniendo según la importancia que tuvieran para el sistema de recomendación. Por lo que la limpieza de la base de datos se han realizado las siguientes tareas:

Eliminación de **columnas duplicadas** como pueden ser:

- 'gk_diving' y 'goalkeeper_diving'
- 'gk_handling' y 'goalkeeper_handling'
- Y otras características de los porteros, que se han unido en su formato largo (goalkeeper_habilidad) para su mejor entendimiento.

Eliminación de **columnas vacías**, que no tenían ningún valor asignado:

- defending_marking

Eliminación de columnas que se ha considerado que no tenían **ningún valor** para la aplicación:

- sofifa_id
- player_url
- short_name
- weight_kg
- league_name
- league_rank
- international_reputation
- work_rate
- body_type
- real_face
- player_tags
- team_position
- team_jersey_number
- loaned_from
- joined
- nation_position
- nation_jersey_number
- pace
- shooting
- passing

- dribbling
- defending
- physic

Unión de atributos con información redundante:

- La edad y la fecha de nacimiento, ya que de una se puede obtener la otra.

Después de la limpieza, los jugadores quedaron con **37 atributos** que son los que los definirán y ayudarán a la búsqueda y recomendación de estos.

La medida de las estadísticas que definen qué tipo de jugador son, va enlazado con un número comprendido entre **0 al 100**, siendo 0 el peor de los casos y siendo 100 el mejor de los casos. Esta medida vendrá dada en los atributos individuales de los jugadores.

Tenemos características **generales** de los jugadores que son:

- **Nombre del jugador:**
El propio nombre del jugador que lo identificará en la base de datos.
- **Fecha de nacimiento:**
Es un factor clave a la hora de la búsqueda de un jugador, ya que nos puede interesar un jugador joven con gran potencial de futuro, uno de mayor edad que esté más experimentado o uno de mediana edad que no disponga de tanta experiencia pero que tenga algo más de recorrido.
- **Altura:**
Dependiendo del estilo de juego que tenga nuestro equipo, por ejemplo, un jugador alto vendría bien cuando lo que buscamos es que se centren balones por arriba.
- **Nacionalidad:**
 - Es importante conocer la nacionalidad de los jugadores, ya que algunas ligas tienen límite de jugadores extranjeros a la hora de inscribirlos en la competición.
 - O por ejemplo el Athletic de Bilbao que sólo puede tener jugadores con nacionalidad española.
- **Club en el que juegan:**
Puede limitar el fichaje de un jugador si éste juega en un equipo que es rival directo del nuestro.

Las siguientes características están centradas en el **valor que tiene el jugador** e información del contrato que tiene con su club actual.

- **Valor de mercado:**

Mínimo coste del jugador que se debería pagar para que el otro club aceptara una oferta por el mismo.

- **Sueldo:**

Cantidad de dinero que recibe en su equipo actual al mes.

- **Valor de la cláusula de rescisión:**

Precio en euros de la cláusula de rescisión que si es pagada por otro club el jugador podrá abandonar su equipo actual sin importar la duración de su contrato.

- **Año de finalización del contrato:**

El año en que finaliza el contrato del jugador en su equipo actual, es importante a la hora de querer fichar un jugador, si el año de finalización es cercano una de las opciones podría ser esperar a que termine.

Atributos **generales** como jugador:

- **Posición:**

Es el lugar o lugares que el jugador se encuentra cómodo jugando en el campo, por lo que nos permite buscar un jugador acorde a la posición que necesitamos en nuestro equipo.

Se pueden encontrar en la aplicación las siguientes posiciones:



- **Habilidad total:**

Valoración general que se tiene del jugador.

- **Potencial del jugador:**

Habilidad máxima que puede llegar a obtener el jugador si sigue entrenando correctamente.

- **Pierna preferente:**

La pierna que tiene tendencia a usar.

- **Habilidad con pierna mala:**

Capacidad del jugador de utilizar correctamente su pierna mala.

Atributos **físicos** de los jugadores:

- **Aceleración:**
Capacidad del jugador de llegar a su velocidad máxima.
- **Velocidad:**
Mide qué tan rápido puede correr el jugador.
- **Salto:**
Habilidad del jugador de saltar, cuanto mayor sea más alto podrá llegar.
- **Resistencia:**
Capacidad del jugador de aguantar todo el partido sin cansarse.
- **Agilidad:**
Mide la facilidad del jugador de moverse o girar.

Atributos **técnicos** de los jugadores:

- **Control de balón:**
Capacidad de un jugador de controlar el balón cuando lo recibe.
- **Remate de cabeza:**
Mide la capacidad de un jugador de hacer remates con la cabeza.
- **Finalización:**
Precisión de los disparos con el pie, dentro del área grande.
- **Tiros lejanos:**
Mide la precisión de los disparos desde fuera del área.
- **Penaltis:**
Mide la precisión y acierto de los tiros de penal.
- **Precisión en faltas:**
Mide la precisión y acierto de un tiro libre directo a puerta.
- **Pases largos:**
Clasifica la precisión de un jugador que realice un pase largo a su compañero de equipo.
- **Pases cortos:**
Determina la precisión de un jugador para pasar el balón a una corta

distancia.

- **Regates:**

Habilidad del jugador para llevar la pelota y regatear a un oponente.

- **Marcaje:**

Capacidad de cubrir a un jugador contrario.

Atributos **mentales** de los jugadores:

- **Agresividad:**

Cualidad que mide la cantidad de veces y frecuencia que el jugador hace empujones, entradas y entradas fuertes.

- **Compostura:**

Capacidad del jugador, de no ponerse nervioso y fallar en un pase o disparo a puerta.

- **Posicionamiento:**

Capacidad de un jugador para detectar espacios abiertos y moverse a buenas posiciones que ofrecen una ventaja de ataque.

Atributos de **portero** de los jugadores:

- **Colocación:**

Capacidad del portero de posicionarse correctamente al detener disparos.

- **Estirada:**

Capacidad del portero de alcanzar balones por el aire.

- **Paradas:**

Mide qué tan limpiamente atrapa la pelota y la mantiene entre sus manos, sin que se le escape.

- **Golpeo:**

Utilizado para medir la longitud y la precisión de los saques, ya sea con las manos o desde el suelo.

- **Reflejos:**

Agilidad del portero para detener un tiro.

Con estos atributos se crearán diferentes características para poder identificar de manera más sencilla a los jugadores. Podemos encontrar las siguientes características algunas son combinación de uno o varios atributos.

- **Joven:**

Cuando la edad del jugador es menor o igual a 22 años.

- **Experimentado:**

Cuando la edad del jugador es mayor o igual a 30 años.

- **Joven promesa:**

Cuando la edad del jugador es menor o igual a 22 años y su potencial de habilidad es mayor al actual.

- **Polivalente:**

Cuando un jugador puede jugar correctamente en 3 o más posiciones.

- **Playmaker:**

Cuando el jugador tiene el atributo de regate mayor o igual a 85, significa que el jugador tiene facilidad para hacer jugadas arriesgadas con éxito.

- **Alto:**

Cuando la altura del jugador supera los 185 cm.

- **Bajo:**

Cuando la altura del jugador es menor a 165 cm.

- **Buen cabeceador:**

Cuando la altura del jugador supera los 185 cm y su habilidad para rematar de cabeza es igual o mayor a 75.

- **Agresivo:**

Cuando el atributo de agresividad es mayor o igual a 75.

- **Calmado:**

Cuando el atributo de agresividad es menor o igual a 25.

- **Buen tirador de falta:**

Cuando el jugador tiene el atributo de tirador de falta mayor o igual a 75.

- **Buen tirador de penaltis:**

Cuando el jugador tiene el atributo de tirador de penaltis mayor o igual a 75.

- **Buen tirador lejano:**

Cuando el jugador tiene el atributo de tiros lejanos mayor o igual a 75.

- **Killer:**

Cuando el jugador tiene la facilidad de meter goles con tiros dentro del área, el atributo debe ser mayor o igual a 80.

- **Escurridizo:**

Cuando el jugador es rápido y tiene la habilidad de moverse a posiciones para crear peligro, los atributos de velocidad, posicionamiento y agilidad deben ser mayor o igual a 75.

- **Acierto en pases:**

Cuando el jugador tiene los atributos de pases cortos y pases largos mayor o igual a 75.

- **Diestro:**

Cuando el jugador puede usar correctamente su pierna derecha.

- **Zurdo:**

Cuando el jugador puede usar correctamente su pierna izquierda.

- **Defensivo:**

Cuando el jugador tiene en el atributo de marcaje igual o mayor a 75.

- **Coste barato:**

Cuando el precio de mercado del jugador es menor o igual a 20.000.000 de euros.

- **Coste medio:**

Cuando el precio de mercado del jugador está entre 20.000.000 y 60.000.000 de euros.

- **Coste alto:**

Cuando el precio de mercado del jugador es mayor o igual a 60.000.000 de euros.

- **Veloz:**

Cuando el jugador tiene el atributo de velocidad mayor o igual a 75.

- **Ágil:**

Cuando el jugador tiene el atributo de agilidad mayor o igual a 75.

- **Buen aguante físico:**
Cuando el jugador tiene el atributo de resistencia mayor o igual a 80.
- **Buen saque:**
Cuando el jugador tiene en el atributo de portero golpeo mayor o igual a 80.
- **Parada aérea:**
Cuando el jugador tiene en el atributo de portero salto mayor o igual a 80.
- **Buenos reflejos:**
Cuando el jugador tiene en el atributo de portero reflejos mayor o igual a 80.
- **Contrato de larga duración:**
Cuando el jugador tiene un contrato con su equipo actual superior o igual a 4 años.

Estas características ayudarán a la hora de implantar los algoritmos de recomendación del sistema, que se explicará su funcionamiento más adelante.

Como nuestra base de datos está en un archivo.csv lo primero que debemos hacer es exportar la información para crear la nueva base de datos en cypher, en donde creamos los siguientes nodos:

- **Nodo (Jugador)** este nodo será el principal de la aplicación ya que contendrá los atributos anteriormente definidos, su propiedad principal, la que lo identifique y lo diferencie con el resto será el nombre.
- **Nodo (Posición)** este nodo serán las diferentes posiciones que el jugador puede tener, siendo esta característica la principal a la hora de buscar jugadores, como podrían ser las temáticas en una página de películas. Cada jugador puede estar relacionado con uno o varios de estos nodos.
- **Nodo (Característica)** este nodo contendrá diferentes características de los jugadores basándonos en los atributos que tenemos, facilitando así la recomendación de jugadores. Las diferentes características que existen fueron definidas anteriormente.
- **Nodo (Usuario)** este nodo contendrá a los usuarios que estarán usando la aplicación, se crearán diferentes relaciones con el **nodo Jugador** para los sistemas de recomendación de la aplicación.

Y existirán las siguientes relaciones, que unirán los nodos y darán sentido a la base de datos:

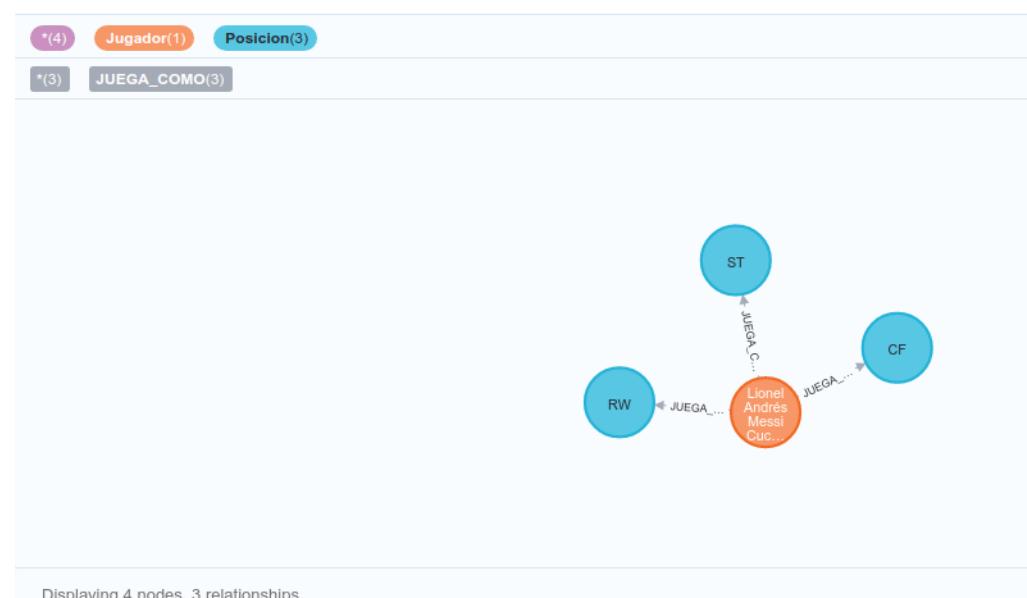
- **Relación (JUEGA_COMO)** entre el jugador y la posición, indicará a la aplicación qué lugares puede tomar éste dentro del campo.
- **Relación (VISITA_PERFIL)** entre el usuario y el jugador, esta relación también almacenará la cantidad de veces que el usuario visita el perfil de un determinado jugador, siendo muy útil para luego realizar recomendaciones.
- **Relación (LIKES)** entre el usuario y el jugador, esta relación también almacenará cuando un usuario está interesado en un jugador y quiere guardarlo en una selección de favoritos, después los jugadores almacenados aquí aparecerán en el perfil del usuario.
- **Relación (TIENE)** entre el jugador y la característica, cuando un jugador tiene los atributos necesarios para tener una característica específica se creará esta relación.

Algunos ejemplos de los nodos con las relaciones:

Mediante la consulta:

```
MATCH (j:Jugador {nombre: 'Lionel Andrés Messi Cuccittini'})-[r:JUEGA_COMO]->(p:Posicion)
RETURN j, r, p;
```

Podemos ver la relación que tiene el jugador **Lionel Andrés Messi Cuccittini** con las diferentes posiciones que puede jugar, en este caso CF, ST y RW.

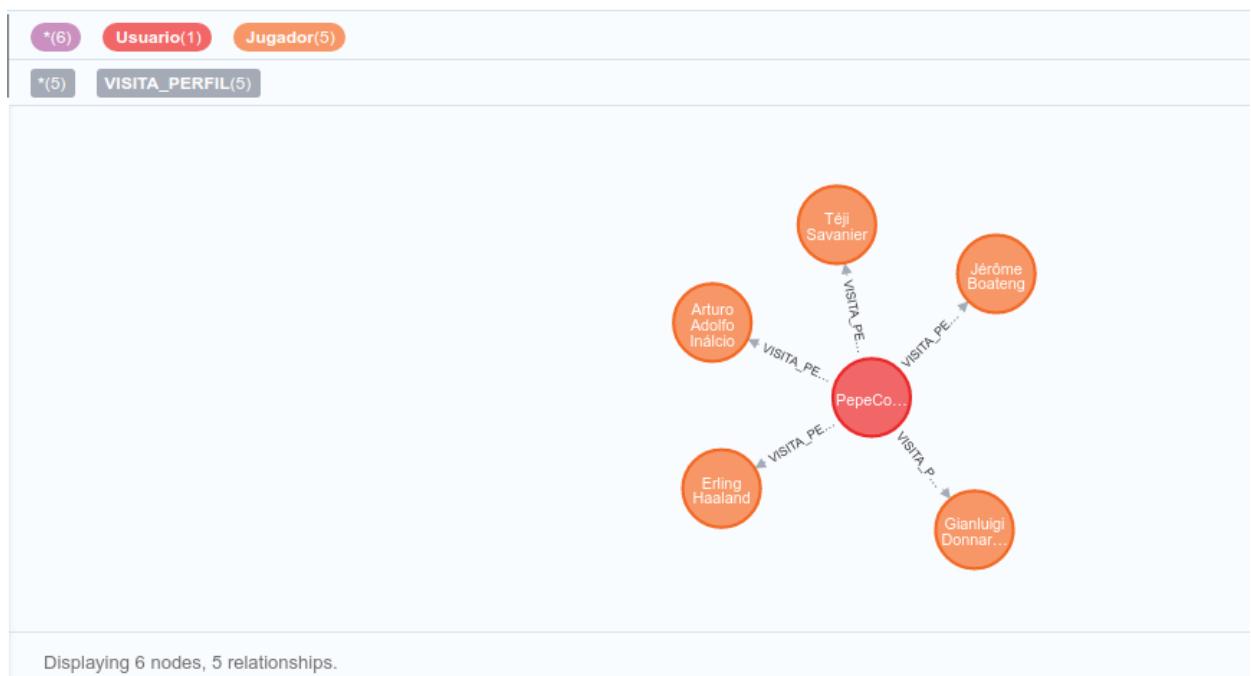


Displaying 4 nodes, 3 relationships.

Mediante la consulta:

```
MATCH (u:Usuario {nombre: 'PepeCortez'})-[v:VISITA_PERFIL]->(j:Jugador)
RETURN u, j, v;
```

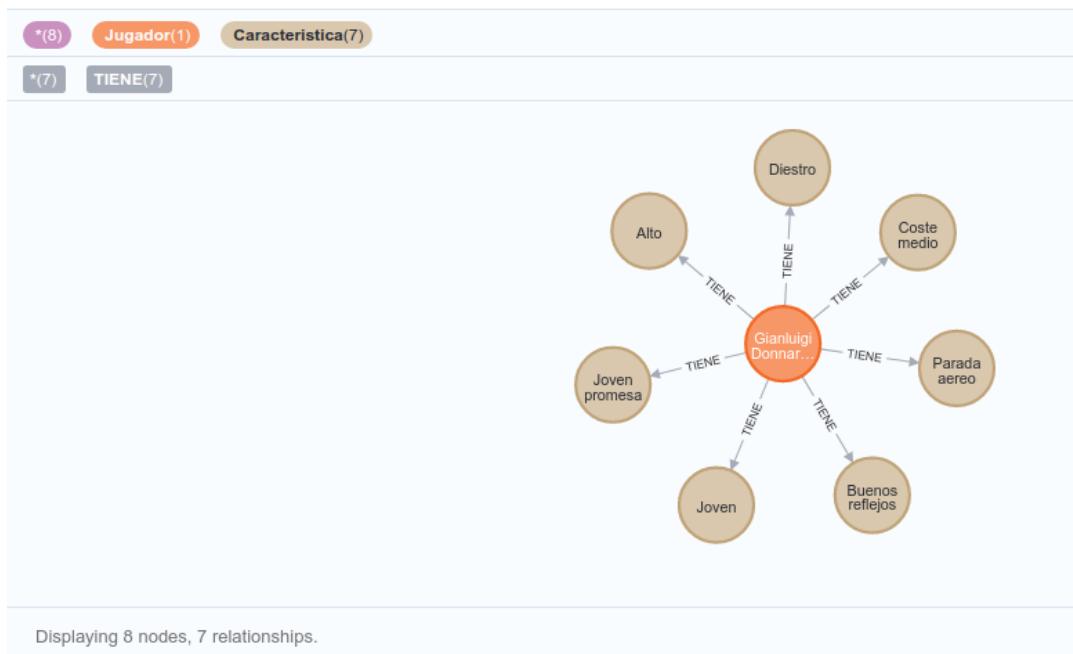
Podemos ver la relación que se crea cuando el usuario **PepeCortez** visita los diferentes perfiles de los jugadores. Podemos observar que ha visitado los perfiles de los jugadores Gianluigi Donnarumma, Jérôme Boateng, Téji Savanier, Arturo Adolfo Inálcio Dutra y Erling Haaland.



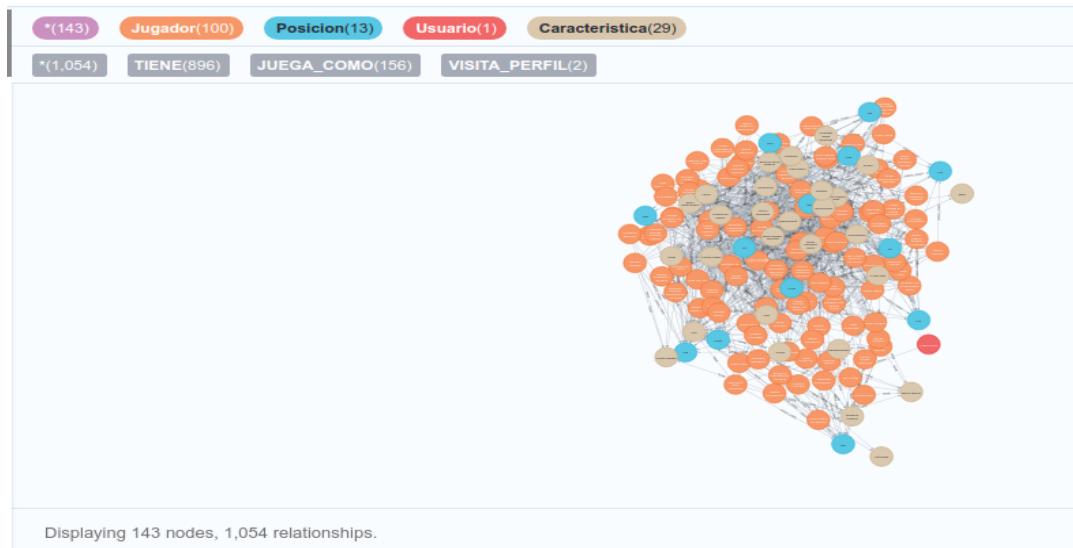
Mediante la consulta:

```
MATCH (j:Jugador {nombre: 'Gianluigi Donnarumma'})-[t:TIENE]->(c:Caracteristica)
RETURN j, t, c;
```

Podemos ver la relación que tiene el jugador **Gianluigi Donnarumma** con las diferentes características que hemos definido. En este caso el jugador tiene las características de Joven, Joven promesa, Buenos reflejos, Parada aéreo, Alto, Diestro y Coste medio.



Visión general de la base de datos:



Explicación de los algoritmos

Aquí se explicarán de forma detallada los diferentes algoritmos que se han utilizado para los sistemas de recomendación que existen en la aplicación, pero lo primero que debemos hacer es instalar la librería de Neo4J **Graph Data Science** (gds) para poder trabajar con ellos, la debemos descargar desde la página de descargas de Neo4j.

Cuando esté descargada, para instalarla debemos añadirla en la siguiente ruta **/var/lib/neo4j/plugins** y una vez añadida reiniciamos el servicio de neo4j para aplicar los cambios con el comando:

```
miguel@miguel-Aspire-V3-571G:~$ service neo4j restart
```

La aplicación tiene un buscador donde se aplican diferentes filtros para que los usuarios puedan hacer una búsqueda inicial de qué tipo de jugador está interesado, esto ayudará luego a que el sistema pueda realizar recomendaciones. Ambos algoritmos de recomendación están basados en la similitud entre los diferentes jugadores para hacer sugerencias.

El **1º algoritmo** que se va a explicar es el que nos dará los jugadores similares al entrar al perfil de un jugador, este algoritmo está basado en el algoritmo de **similitud de Jaccard**, que mide la semejanza entre conjuntos y utiliza la siguiente fórmula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

En nuestro caso se ha modificado un poco el algoritmo para limitar la búsqueda a jugadores que tengan la misma posición al jugador que estamos ojeando, por lo que el algoritmo primero consultará que tienen la misma posición una vez hecho esto mirará la similitud de los jugadores basándose en los nodos Característica que tienen en común, a mayor cantidad de nodos iguales y menos distintos la similitud será mayor.

Busca las relaciones del jugador que estamos ojeando con el nodo **Característica**.

```
var queryRecomendados = 'MATCH (j1:Jugador {nombre: \''+jugador.nombre+'\'} )-[:TIENE]->(skill1) ';
```

Crea una lista de las relaciones del jugador anterior con el nodo **Característica**.

```
queryRecomendados += 'WITH j1, collect(id(skill1)) AS j1Skill ';
```

Busca jugadores que jueguen en la misma posición al jugador que estamos ojeando.

```
queryRecomendados += 'MATCH (j2:Jugador)-[:JUEGA_COMO]->(:Posicion { posicion: \''+posicion+'\'} ) ';
```

Busca las relaciones de estos jugadores con el nodo **Característica** siendo este jugador diferente al que estamos ojeando.

```
queryRecomendados += 'MATCH (j2)-[:TIENE]->(skill2) WHERE j1 <> j2 ';
```

Por último, calcula la similitud con el algoritmo de jaccard, devolviendo la información de los 3 jugadores más similares al que estamos ojeando de manera descendente.

```
queryRecomendados += 'WITH j2, gds.alpha.similarity.jaccard(j1Skill, collect(id(skill2))) AS similitud  
queryRecomendados += 'RETURN j2 ORDER BY similitud DESC LIMIT 3;';
```

El 2º algoritmo es el que entrará en acción cuando pulsemos el botón de recomendar, para que éste funcione primero debemos ojear jugadores así el sistema tendrá una referencia del jugador en el que estamos interesados y hará una recomendación en base a éste.

Este algoritmo es más complejo que el anterior ya que para cada posición se definen cuales son los atributos más importantes, para compararlos con los demás jugadores y usará la **distancia euclídea** para determinar la similitud de los jugadores.

La distancia euclídea utiliza la siguiente fórmula:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_i - q_i)^2 + \dots + (p_n - q_n)^2}.$$

Esto nos permitirá calcular la similitud entre los conjuntos de datos, primero se verá cuáles son los atributos utilizados para calcular la recomendación para cada posición y su explicación de la elección de esos atributos.

Para la posición de **portero** al tener atributos especiales se tienen en cuenta los mismos que son:

- Edad
- Altura
- Salto de portero
- Parada
- Golpeo de portero
- Posicionamiento de portero
- Reflejos de portero
- Potencial

Para la posición de **defensa central**, es una posición defensiva y tienen que tener buenas habilidades para marcar a los delanteros, también se debe tener en cuenta la capacidad de sacar el balón desde atrás ya sea mediante pases cortos o largos y por último su agresividad para conocer si es propenso a cometer muchas faltas, por ello se tienen en cuenta los siguientes atributos.

- Edad
- Altura
- Marcaje
- Pases largos
- Pases cortos
- Agresividad
- Potencial

Para la posición de **lateral** ya sea por la derecha o por la izquierda, es una posición defensiva que sirve de apoyo a los defensas centrales, por ello se tiene en cuenta su habilidad para marcar a los jugadores, también suelen ser jugadores que se les pide que tengan un buen juego aéreo para defender faltas y corners y por último su resistencia ya que su mayor parte del partido tienen una labor defensiva pero también pueden realizar jugadas ofensivas, por ello se piden los siguientes atributos:

- Edad
- Altura
- Marcaje
- Salto
- Resistencia
- Potencial

Para la posición de **carrilero** tanto derecho como izquierdo es considerado una posición ofensiva sobretodo pero que también debe realizar tareas defensivas, por lo que el jugador debe tener una buena resistencia y velocidad para subir y bajar en

las jugadas de ataque y defensa durante todo el partido y un buen posicionamiento para buscar huecos entre las líneas defensivas enemigas y crear jugadas de peligro, por ello se piden los siguientes atributos:

- Edad
- Altura
- Resistencia
- Velocidad
- Posicionamiento
- Potencial

Para la posición de **mediocentro defensivo** es una posición defensiva, por lo que al igual que los defensas necesitará una buena capacidad de marcaje, también que tenga buena resistencia para aguantar todo el partido, aceleración y velocidad para interceptar los ataques del equipo enemigo, capacidad para sacar el balón jugado mediante pases largos o cortos, por ello se piden los siguientes atributos:

- Edad
- Altura
- Resistencia
- Velocidad
- Aceleración
- Marcaje
- Pases largos
- Pases cortos
- Potencial

Para la posición de **mediocentro/derecho/izquierdo** es una posición neutral que debe hacer labores tanto defensivas como ofensivas y una de las cosas más importantes es la capacidad de visión del campo que tiene el jugador y poder sacar el balón jugado por lo que debe hacer buenos pases cortos y largos, buen control de balón y agilidad para moverse rápidamente y sin perder el balón en errores, capacidad para aguantar todo el partido moviéndose y una buena mentalidad para no cometer fallos durante el partido a pesar del resultado, por ello se piden los siguientes atributos:

- Edad
- Altura
- Resistencia
- Pases largos
- Pases cortos
- Control de balón
- Agilidad
- Compostura

- Potencial

Para la posición de **mediocentro ofensivo** es una posición ofensiva que es parecida a la de mediocentro pero con una mayor cercanía al área rival, como también debe realizar tareas defensivas es importante que tenga una buena resistencia, buenos pases cortos para no perder el balón, capacidad de controlar el balón y agilidad para moverlo rápidamente y al estar cerca del área que tenga posibilidad de hacer buenos disparos de lejos, por ello se piden los siguientes atributos:

- Edad
- Altura
- Resistencia
- Pases cortos
- Control de balón
- Agilidad
- Disparos lejanos
- Potencial

Para la posición de **extremo** tanto derecho como izquierdo es una posición ofensiva que se juega por las bandas por lo que deben tener una gran velocidad y aceleración para poder moverse rápidamente y agilidad para llevar el balón de forma rápida sin perderlo, también capacidad de regate para superar a los defensores, buscar huecos en la línea defensiva enemiga y suelen ser los encargados de tirar las faltas y los saques de esquina, por ello se piden los siguientes atributos:

- Edad
- Altura
- Velocidad
- Aceleración
- Agilidad
- Regates
- Tiros de falta
- Posicionamiento
- Potencial

Para la posición de **mediapunta** es una posición ofensiva muy parecido a la posición de delantero pero es más como un apoyo para este, por lo que debe tener un buen pase de balón en corto, agilidad y control de balón para moverse rápido con el mismo, resistencia para aguantar el partido, capacidad de regate para crear ocasiones de gol y al estar cerca de la portería enemiga que pueda hacer buenos disparos lejanos, por ello se utilizan los siguientes atributos:

- Edad
- Altura
- Resistencia
- Pases cortos
- Control de balón
- Agilidad
- Disparos lejanos
- Regates
- Potencial

Para la posición de **delantero** es una posición ofensiva por lo que es necesario que tenga una velocidad y aceleración alta para poder escapar de los defensores, capacidad para llevar rápidamente el balón pegado al pie sin perderlo, buscar huecos libres para crear peligro, disparos lejanos y finalización para marcar goles y remate para hacer buenos golpes de cabeza, por ello se piden los siguientes atributos:

- Edad
- Altura
- Velocidad
- Aceleración
- Agilidad
- Finalización
- Disparos lejanos
- Remate
- Posicionamiento
- Potencial

Ahora que ya conocemos los atributos que se utilizarán para cada posición vamos a ver cómo funciona el algoritmo internamente.

Lo primero que hacemos es buscar jugadores que jueguen en la posición que queremos hacer la recomendación.

```
var query = 'MATCH (j:Jugador)-[:JUEGA_COMO]->(p:Posicion {posicion: \''+posicion+'\'})';
```

Ahora filtramos esos jugadores con los que el usuario haya visitado su perfil.

```
// CONTROBAMOS QUE HAYA UNA RELACION DE QUE EL USUARIO HA VISITADO EL PERFIL DEL JUGADOR
query += 'MATCH (u:Usuario)-[v:VISITA_PERFIL]->(j) WHERE u.nombre = \''+usuario+'\' ;'
```

Devolvemos esos jugadores que el usuario ha visitado su perfil ordenándolos de mayor a menor según la cantidad de veces que el usuario ha visitado su perfil.

```
// ATRIBUTOS DEDICADOS AL VISTADO MÁS RESES  
query += 'RETURN j.nombre ORDER BY v.vectores DESC LIMIT 1;'
```

Ahora que ya tenemos el jugador que nos interesa para hacer la recomendación, debemos buscar los demás que se asemejen a éste.

Buscamos ese jugador para crear la recomendación.

```
var queryRecomendados = 'MATCH (j1:Jugador {nombre: \''+ results.records[0]._fields +'\'} ) ';
```

Buscamos todos los jugadores que juegan en la misma posición.

```
queryRecomendados += 'MATCH (j2:Jugador)-[:JUEGA_COMO]->(:Posicion { posicion: \''+posicion+'\'} ) ';
```

Ahora según la posición utilizamos los atributos que hemos explicado anteriormente que utilizaremos para el algoritmo y utilizamos la distancia euclídea para hacer la comparación.

```
// ATRIBUTOS SEGÚN LA POSICIÓN  
if(posicion == 'GK') { // PORTERO  
    queryRecomendados += 'WITH j1, j2, \  
    gds.alpha.similarity.euclideanDistance([duration.between(date(j1.fechaNacimiento), date()).years,  
    [duration.between(date(j2.fechaNacimiento), date()).years, j2.altura, j2.porteroSalto, j2.porteroP  
} else if(posicion == 'CB') { // DEFENSA CENTRAL  
    queryRecomendados += 'WITH j1, j2, \  
    gds.alpha.similarity.euclideanDistance([duration.between(date(j1.fechaNacimiento), date()).years,  
    [duration.between(date(j2.fechaNacimiento), date()).years, j2.altura, j2.marcaje, j2.pasesCortos,  
} else if(posicion == 'RB' || posicion == 'LB') { // LATERAL DERECHO / IZQUIERDO  
    queryRecomendados += 'WITH j1, j2, \  
    gds.alpha.similarity.euclideanDistance([duration.between(date(j1.fechaNacimiento), date()).years,  
    [duration.between(date(j2.fechaNacimiento), date()).years, j2.altura, j2.marcaje, j2.salto, j2.res  
} else if(posicion == 'RWB' || posicion == 'LWB') { // CARRILERO DERECHO / IZQUIERDO  
    queryRecomendados += 'WITH j1, j2, \  
    gds.alpha.similarity.euclideanDistance([duration.between(date(j1.fechaNacimiento), date()).years,  
    [duration.between(date(j2.fechaNacimiento), date()).years, j2.altura, j2.resistencia, j2.velocidad  
} else if(posicion == 'CDM') { // MEDIO CENTRO DEFENSIVO  
    queryRecomendados += 'WITH j1, j2, \  
    gds.alpha.similarity.euclideanDistance([duration.between(date(j1.fechaNacimiento), date()).years,
```

Devolvemos la lista de jugadores ordenándolos de mayor a menor similitud.

```
queryRecomendados += 'RETURN j2 ';  
queryRecomendados += 'ORDER BY similitud LIMIT 7;';
```

Por último el backend enviará esa información al frontend para mostrarsela al usuario.

Análisis de los resultados

En este apartado se mostrarán los diferentes resultados según los intereses que tengamos en mente. Por lo que se expondrán distintos resultados basándonos en búsquedas reales que podría realizar cualquier equipo que estuviera interesado en utilizar la aplicación y ver el funcionamiento de la misma.

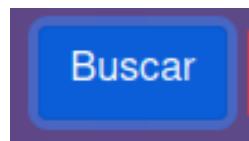
Ejemplo 1: Búsqueda de mediocentro con experiencia

Tenemos un equipo con bastantes jugadores jóvenes por lo que muchos de ellos no tienen mucha experiencia jugando partidos bajo gran presión ni en torneos internacionales, por lo que estamos buscando un mediocentro que tenga experiencia con buenos pases en corto y largo para poder sacar el balón jugado y que no se ponga nervioso cuando juegue bajo presión, y también que pueda guiar a los jugadores menos experimentados a poder controlar estas emociones durante los partidos. Añadiremos también que sea ambidiestro y que sepa guardar la compostura.

Características generales		
● Posición: Medio centro	● Valor de mercado: 150.000.000€	● Sueldo: 600.000€
● Edad: 45 años	● Altura: 160 cm	● Pierna: Ambidiestro

Atributos específicos		
● Agresividad: Sin definir	● Compostura: Alto	● Posicionamiento: Sin definir
● Penaltis: Sin definir		
● Acceleración: Sin definir	● Velocidad: Sin definir	● Agilidad: Sin definir
● Salto: Sin definir	● Resistencia: Sin definir	
● Finalización: Sin definir	● Regates: Sin definir	● Remate de cabeza: Sin definir
● Tiro de falta: Sin definir	● Pases cortos: Alto	● Pases largos: Alto
● Control de balón: Sin definir	● Disparo lejano: Sin definir	● Marcaje: Sin definir

Como podemos ver en las imágenes hemos colocado los filtros que nos interesan para buscar el jugador y los demás los hemos dejado como estaban. Ahora le damos al botón de buscar.



Una vez hayamos dado al botón nos muestra los siguientes resultados.

Resultados totales: 86												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Kevin De Bruyne	30	181	Manchester City	Belgium	87,000,000€	370,000€	2023	161,000,000€	Ambidestro	91	91	
Toni Kroos	31	183	Real Madrid	Germany	55,000,000€	310,000€	2023	112,800,000€	Ambidestro	88	88	
Bruno Miguel Borges Fernandes	27	179	Manchester United	Portugal	63,000,000€	195,000€	2025	124,400,000€	Ambidestro	87	90	
Luka Modrić	36	172	Real Madrid	Croatia	24,500,000€	260,000€	2021	50,200,000€	Ambidestro	87	87	
Paul Pogba	28	191	Manchester United	France	50,000,000€	190,000€	2021	98,800,000€	Ambidestro	86	87	
Marco Verratti	29	165	Paris Saint-Germain	Italy	50,000,000€	135,000€	2024	96,300,000€	Ambidestro	86	87	
Sergej Milinković-Savić	26	191	Lazio	Serbia	49,000,000€	94,000€	2023	87,000,000€	Ambidestro	85	89	

« < 1 2 3 4 ... > »

Como hemos dicho que queremos un jugador con experiencia vamos a buscar jugadores con edades comprendidas entre 28 años en adelante, para ellos en la primera fila pulsaremos dos veces en la columna de edad para que nos coloque a los jugadores con mayor edad primero.

Resultados totales: 86												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Andrés Iniesta Luján	37	171	Vissel Kobe	Spain	5,500,000€	12,000€	2021	7,200,000€	Ambidestro	81	81	
Luka Modrić	36	172	Real Madrid	Croatia	24,500,000€	260,000€	2021	50,200,000€	Ambidestro	87	87	
Marco Parolo	36	184	Lazio	Italy	4,000,000€	40,000€	2021	6,800,000€	Ambidestro	79	79	
José Ernesto Sosa	36	179	Fenerbahçe SK	Argentina	4,300,000€	47,000€	2022	8,200,000€	Ambidestro	78	78	
João Filipe Iria Santos Moutinho	35	170	Wolverhampton Wanderers	Portugal	15,000,000€	110,000€	2022	28,500,000€	Ambidestro	83	83	
James Philip Milner	35	175	Liverpool	England	7,000,000€	94,000€	2022	13,000,000€	Ambidestro	80	80	
Enzo Nicolás Pérez	35	178	River Plate	Argentina	5,500,000€	21,000€	2023	7,700,000€	Ambidestro	80	80	

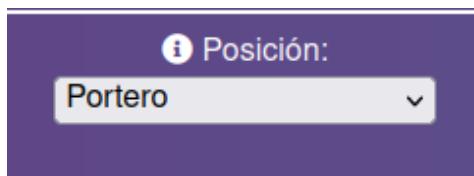
« < 1 2 3 4 ... > »

Una vez hecho esto, ya podemos ir accediendo a los perfiles de los jugadores que nos interesen.

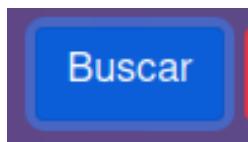
Ejemplo 2: Búsqueda de portero sin ninguna especificación

Estamos buscando un portero debido a que nuestro anterior portero ha sido fichado por otro equipo y los porteros reservas que tenemos no nos convencen para que jueguen de titulares, por lo que vamos a usar el sistema para ver si hay alguno que nos pueda interesar.

Lo primero que hacemos es seleccionar la posición de portero en los filtros para buscar jugadores y como no tenemos muy claro que es lo que estamos buscando vamos a dejar los demás filtros por defecto.



Bajamos más abajo y le damos al botón de buscar para que nos muestre los resultados.



Al darle al botón nos mostrará todos los jugadores que jueguen en la posición de portero que hay en la aplicación.

Resultados totales: 116												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Jan Oblak	28	188	Atlético Madrid	Slovenia	75,000,000€	125,000€	2023	159,400,000€	Diestro	91	93	
Marc-André ter Stegen	29	187	FC Barcelona	Germany	69,500,000€	260,000€	2022	147,700,000€	Ambidiestro	90	93	
Alisson Ramsés Becker	29	191	Liverpool	Brazil	62,500,000€	160,000€	2024	120,300,000€	Diestro	90	91	
Thibaut Courtois	29	199	Real Madrid	Belgium	56,000,000€	250,000€	2024	119,000,000€	Zurdo	89	90	
Manuel Neuer	35	193	FC Bayern München	Germany	29,000,000€	125,000€	2023	47,900,000€	Ambidiestro	89	89	
Ederson Santana de Moraes	28	188	Manchester City	Brazil	53,500,000€	195,000€	2024	103,000,000€	Zurdo	88	91	
Samir Handanović	37	193	Inter	Slovenia	16,000,000€	100,000€	2021	26,400,000€	Diestro	88	88	

Como vemos en la imagen anterior nos ha devuelto a 116 jugadores que juegan en esta posición, por lo que podemos ir pasando más páginas para ir viendo más.

Resultados totales: 116												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Pau López Sabata	27	189	Roma	Spain	25,500,000€	52,000€	2024	45,300,000€	Zurdo	83	88	
Koen Casteels	29	197	VfL Wolfsburg	Belgium	22,500,000€	53,000€	2024	39,900,000€	Zurdo	83	85	
Anthony Lopes	31	184	Olympique Lyonnais	Portugal	22,000,000€	69,000€	2023	43,500,000€	Zurdo	83	85	
Steve Mandanda	36	185	Olympique de Marseille	France	7,000,000€	28,000€	2024	13,300,000€	Diestro	83	83	
Kevin Trapp	31	189	Eintracht Frankfurt	Germany	18,500,000€	35,000€	2024	31,500,000€	Ambidiestro	83	83	
Lukáš Hrádecký	32	192	Bayer 04 Leverkusen	Finland	17,500,000€	63,000€	2023	29,800,000€	Diestro	83	83	
Aitor Fernández Abarisketa	30	182	Levante UD	Spain	18,500,000€	31,000€	2023	38,900,000€	Diestro	83	83	

« < ... 3 4 5 ... > »

Al ir usando la aplicación nos damos cuenta de que queremos un portero joven y que tenga buen potencial de futura, para que tenga una gran trayectoria en el equipo, por lo que en la tabla pulsamos dos veces en la primera fila en la columna de potencial, para que nos filtre los jugadores de mayor potencial a menor potencial.

Resultados totales: 116												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Jan Oblak	28	188	Atlético Madrid	Slovenia	75,000,000€	125,000€	2023	159,400,000€	Diestro	91	93	
Marc-André ter Stegen	29	187	FC Barcelona	Germany	69,500,000€	260,000€	2022	147,700,000€	Ambidiestro	90	93	
Gianluigi Donnarumma	22	196	Milan	Italy	41,500,000€	34,000€	2021	78,900,000€	Diestro	85	92	
Alisson Ramsés Becker	29	191	Liverpool	Brazil	62,500,000€	160,000€	2024	120,300,000€	Diestro	90	91	
Ederson Santana de Moraes	28	188	Manchester City	Brazil	53,500,000€	195,000€	2024	103,000,000€	Zurdo	88	91	
Thibaut Courtois	29	199	Real Madrid	Belgium	56,000,000€	250,000€	2024	119,000,000€	Zurdo	89	90	
Manuel Neuer	35	193	FC Bayern München	Germany	29,000,000€	125,000€	2023	47,900,000€	Ambidiestro	89	89	

« < 1 2 3 4 ... > »

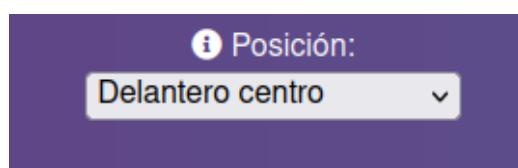
Como podemos ver es muy sencillo utilizar la aplicación y modificar los elementos para que nos vaya mostrando lo que queramos según vayamos sabiendo qué es lo que queremos específicamente, en nuestro caso en esta primera página el portero que más se asemeja a lo que estamos buscando sería **Gianluigi Donnarumma**, ya que es un jugador joven de 22 años y además con un buen potencial de futuro, aunque dependiendo de nuestra economía a lo mejor es un jugador algo caro, pero podemos seguir pasando de página para ir viendo lo que nos muestra la aplicación

o si queremos filtrar de manera más concisa podemos volver a los filtros anteriores e ir cambiandolos según lo que nos interese.

Ejemplo 3: Recomendación de delanteros por búsquedas anteriores

Estamos interesados en buscar un delantero para completar nuestro equipo de fútbol, pero las búsquedas que hemos realizado en la aplicación no nos ha convencido ninguno de los jugadores que nos ha mostrado, por lo que vamos a usar el botón de recomendar para que el sistema nos ofrezca una lista de jugadores similares al jugador que más hemos estado ojeando.

Lo primero que hacemos es seleccionar la posición de delantero centro en los filtros para buscar jugadores.



Los demás filtros los dejamos como están, ya que usaremos la herramienta de recomendación de la aplicación para buscar los jugadores, para ello bajamos abajo y le damos al botón de recomendar.



A continuación nos mostrará los resultados.

Resultados totales: 7												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
Erling Haaland	21	194	Borussia Dortmund	Norway	45,000,000€	56,000€	2024	85,500,000€	Zurdo	84	92	
Marcus Thuram	24	192	Borussia Mönchengladbach	France	20,500,000€	43,000€	2023	39,000,000€	Diestro	80	86	
Alexander Isak	22	192	Real Sociedad	Sweden	18,000,000€	28,000€	2024	41,400,000€	Ambidestro	79	86	
Mason Greenwood	20	181	Manchester United	England	14,500,000€	33,000€	2023	30,500,000€	Ambidestro	77	89	
Anthony Martial	26	184	Manchester United	France	41,000,000€	155,000€	2024	81,000,000€	Diestro	84	88	
Marcus Rashford	24	186	Manchester United	England	53,000,000€	150,000€	2023	111,300,000€	Ambidestro	85	91	
João Félix Sequeira	22	181	Atlético Madrid	Portugal	32,000,000€	51,000€	2026	72,000,000€	Ambidestro	81	93	

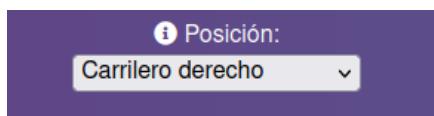
Como podemos observar en la imagen el primero jugador que nos recomendará será el jugador el cual hemos visitado más su perfil y los siguientes son la

recomendación que realiza el sistema basándose en la similitud que tiene con éste, en nuestro caso el perfil del jugador más visitado ha sido el de **Erling Haaland** y los siguientes jugadores son los que más similitud tienen.

Ejemplo 4: Recomendación de carrilero derecho sin búsquedas anteriores

En este caso estamos buscando un carrilero derecho para mejorar nuestro ataque por la banda derecha y que apoye a nuestro extremo derecho en jugadas ofensivas, pero no hemos visitado ningún perfil de ningún jugador en esta posición.

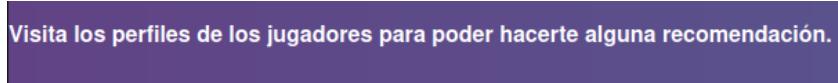
Lo primero que hacemos es seleccionar la posición de carrilero derecho en los filtros para buscar jugadores.



Los demás filtros no los tocamos, ya que volveremos a usar la herramienta de recomendación de la aplicación para buscar los jugadores, para ello bajamos abajo y le damos al botón de recomendar.



En este caso al no haber realizado búsquedas anteriores el sistema no puede hacer una recomendación sobre ningún jugador en esta posición, ya que no conoce nuestras preferencias de búsqueda por lo que nos pide primero que observemos a algún jugador para luego poder realizar la recomendación.



Este es el mensaje que recibiremos de la aplicación en este caso.

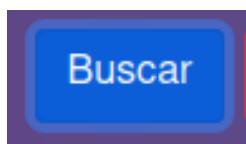
Ejemplo 5: Búsqueda de defensa central por recomendación en perfil

Estamos buscando un defensa central pero no tenemos muy claro qué características queremos que tenga, por lo que vamos a ir observando perfiles de jugadores hasta que encontremos alguno que nos interese.

Lo primero que hacemos es colocar en el filtro de posición para buscar un defensa central.



Le damos al botón de buscar para ver todos los posibles defensas centrales que hay en la aplicación.



Nos aparecerá la lista de estos.

Resultados totales: 170												
Nombre	Edad	Altura	Equipo	Nacionalidad	Valor	Sueldo	Contrato	Clausula	Pierna	Habilidad	Potencial	
A. Benjamin Chiamuloria Paes	21	182	Internacional	Brazil	10,000,000€	20,000€	2023	19,000,000€	Zurdo	78	78	
Ozan Muhammed Kabak	21	186	FC Schalke 04	Turkey	13,000,000€	19,000€	2024	24,700,000€	Diestro	77	87	
Matthijs de Ligt	22	189	Juventus	Netherlands	49,500,000€	80,000€	2024	91,600,000€	Ambidestro	85	92	
Boubacar Kamara	22	184	Olympique de Marseille	France	17,000,000€	29,000€	2022	35,700,000€	Diestro	79	87	
Dan-Axel Zagadou	22	196	Borussia Dortmund	France	16,000,000€	38,000€	2022	30,400,000€	Zurdo	79	86	
Ibrahima Konaté	22	194	RB Leipzig	France	15,500,000€	33,000€	2023	29,500,000€	Diestro	78	88	
Edmond Fayçal Tapsoba	22	192	Bayer 04 Leverkusen	Burkina Faso	15,500,000€	38,000€	2025	29,500,000€	Diestro	78	88	

Vamos pasando las páginas de los resultados y vemos que un jugador nos llama la atención y que podría entrar dentro de lo que estamos buscando. En nuestro caso el jugador es **Aymeric Laporte**, pulsamos dentro de la tabla en la fila correspondiente para acceder al perfil de éste.



Aymeric Laporte (France)
1994-05-27 (27 años)
189 cm

Manchester City	Habilidad	Mentalidad
Sueldo: 200,000€ / mes Contrato hasta: 2024 Clausula de rescisión: 107,800,000€ Valor del mercado: 56,000,000€	Zurdo Habilidad general: 87 / 100 Potencial: 90 / 100	Agresividad 81 Posición 54 Compostura 85 Penaltis 44
Físico	Técnico	Portero

Una vez en el perfil vemos detalladamente toda la información del jugador, pero no nos convence del todo, así que nos dirigimos abajo de todo de la página y nos hará 3 recomendaciones de jugadores similares a **Aymeric Laporte**.

Jugadores similares:



Alessio Romagnoli (Italy)
CB
1995-01-12 (26 años)
188 cm

[Ver perfil](#)


Presnel Kimpembe (France)
CB
1995-08-13 (26 años)
189 cm

[Ver perfil](#)


Stefan de Vrij (Netherlands)
CB
1992-02-05 (29 años)
189 cm

[Ver perfil](#)

Dándole a ver perfil de cada uno de los jugadores recomendados accederemos al perfil de ellos, en nuestro caso vamos a acceder al perfil de **Alessio Romagnoli** porque creemos que puede encajar en nuestro equipo.



Alessio Romagnoli (Italy)
1995-01-12 (26 años)
188 cm

Milan

Sueldo: 51,000€ / mes
Contrato hasta: 2022
Clausula de rescisión: 53,300,000€
Valor del mercado: 30,000,000€

Habilidad

Zurdo
Habilidad general: 83 / 100
Potencial: 87 / 100

Mentalidad

Agresividad: 85
Posición: 31
Compostura: 76
Penaltis: 34

En el perfil de éste se nos volverá a mostrar de forma detallada los atributos de este jugador y abajo del todo se volverá a hacer una recomendación basándose en la similitud que tiene con el jugador que estamos ojeando.

Jugadores similares:



Aymeric Laporte (France)
CB
1994-05-27 (27 años)
189 cm



Presnel Kimpembe (France)
CB
1995-08-13 (26 años)
189 cm



Rúben dos Santos Gato Alves Dias (Portugal)
CB
1997-05-14 (24 años)
187 cm

[Ver perfil](#)

[Ver perfil](#)

[Ver perfil](#)

Como vemos nos vuelve a recomendar a **Aymeric Laporte** y a **Presnel Kimpembe** ya que estos jugadores tendrán muchas similitudes entre ellos y un jugador nuevo llamado **Rúben dos Santos Gato Alves Dias** que no había salido recomendado anteriormente desde el perfil de **Aymeric Laporte**.

Desde aquí podemos seguir navegando entre los perfiles de los jugadores según las recomendaciones que nos va ofreciendo el sistema e ir valorando las características que tienen estos jugadores para finalmente realizar el fichaje, también se puede volver a atrás y con los filtros delimitar la búsqueda.

DAFO

El análisis DAFO se trata de una herramienta que permite evaluar de manera interna y externa las debilidades, amenazas, fortalezas y oportunidades de un proyecto y diseñar una estrategia para afrontar su futuro.

Consta de un **análisis interno** que se compone de las debilidades y fortalezas, donde se estudian las variables de producción, marketing, organización, recursos humanos o personal y finanzas.

Y de un **análisis externo** que se compone de las amenazas y oportunidades, donde se estudian factores que no dependen directamente del proyecto pero que le afectan, donde entran variables como el mercado, el sector, la competencia y el entorno.

- **Debilidades:**

La principal debilidad de la aplicación son los datos, ya que para crear un sistema de recomendación de fichajes avanzado se deberían tener información actualizada de los jugadores, actualizandolos cada vez que se use la aplicación ya que así tendríamos una mejor visión del estado actual del jugador en ese momento, no sólo basándonos en características que han sido definidas anteriormente según su trayectoria. Lo óptimo sería una combinación de datos actualizados y características del jugador.

- **Fortalezas:**

El uso fácil y sencillo de la aplicación la vuelve una de sus fortalezas más importantes comparándolo con lo difícil que sería buscar jugadores de forma manual sin tener ningún software de apoyo como éste, ya que con un par de clicks, rellenando los diferentes filtros te mostrará una gran cantidad de jugadores que entran dentro de los rangos definidos y de una forma sencilla puedes ir modificando estos filtros para buscar a tu jugador ideal, además el sistema ofrece recomendaciones para que esta búsqueda sea mucho más sencilla.

- **Amenazas:**

Muchas empresas se están dando cuenta de lo importante que es el uso de datos para mejorar el rendimiento de las mismas, y mucho más algunos equipos de fútbol con la gran capacidad económica que tienen, por lo que las amenazas pueden ser que ya estén utilizando este tipo de aplicaciones para el rendimiento de fichajes de manera interna, ya que de forma pública no hay mucha información al respecto, aunque ya se sabe que algunos clubes ya están implementando sistemas de **BIG DATA** para mejorar el rendimiento de sus jugadores.

- **Oportunidades:**

Las oportunidades van un poco por la línea de las amenazas, ya que como muchas empresas se están dando cuenta de la importancia de los datos para mejorar sus ingresos, esta puede ser nuestra virtud, ya que no todos los equipos tienen recursos económicos para hacer grandes inversiones en este tipo de software por lo que se podría ofrecer a equipos de menor renombre de forma asequible. Y también debido a la pandemia que muchos clubes estuvieron afectados han tomado la decisión de desprenderse de la figura del ojeador para tener menos gastos y quizás vean con buenos ojos aplicaciones de este estilo.

Problemas y soluciones en el desarrollo

Durante el desarrollo de la aplicación han ido surgiendo diferentes problemas que se han tenido que resolver para poder avanzar en el proyecto, por lo que en este apartado se van a describir estos problemas y la solución tomada para cada uno. Esto puede ayudar para futuros proyectos, ya que conociendo cómo se han resuelto los problemas aquí, si nos encontramos con una situación similar ya no es necesario pensar una solución, ya que ha sido resuelta aquí.

Problema 1: Al tener una base de datos con una gran cantidad de jugadores (20.000) con varias columnas para cada jugador (32 columnas), se ha dificultado la creación de los nodos en cypher ya que la capacidad de computación del ordenador donde se ha desarrollado no es demasiado potente para poder hacerlo.

Solución: Se ha optado por dividir la base de datos usando sólo los primeros 1.000 jugadores desde el código de creación en cypher, pero el archivo sigue manteniendo los 20.000 jugadores, el código los limitará a los 1.000 primeros para poder probar el funcionamiento de la aplicación y ver los resultados, aunque con 20.000 funcionará de la misma manera.

Problema 2: Durante la creación de los nodos en cypher saltó un error de que en la base de datos la columna de 'club_name' en algunos jugadores estaba vacía, y siendo ésta una columna importante para la aplicación hubo que buscar otra solución que no fuera eliminarla.

Solución: Se ha colocado 'Sin equipo' de forma manual a los jugadores que tuvieran esa columna vacía.

Problema 3: En la base de datos las columnas de 'release_clause_eur' y 'contract_valid_until' estaban vacías en algunos jugadores.

Solución: Se ha colocado un 0 a estas columnas.

Problema 4: La idea inicial era que la página de inicio tuviera varias imágenes mostrando las características de la aplicación, pero fue imposible adaptarlas a un tamaño común.

Solución: Se ha optado por colocar una sola imagen fija con un texto dentro.

Problema 5: Inicialmente se quería tener separado la elección de filtros con la muestra de resultados, pero la búsqueda de jugadores se volvía un proceso tedioso.

Solución: Se tomó la decisión de mostrar los datos en la misma vista que la de los filtros, así se podrán editar y ver los resultados de una manera más sencilla.

Problema 6: Cuando se hacía una petición para mostrar los jugadores que coincidieran con los filtros aplicados, se creaba una tabla demasiado grande con mucha información junta.

Solución: Se creó un sistema de paginación, que va dividiendo los jugadores en diferentes páginas para su visualización y poder filtrarlos por las características que aparecen en la tabla.

Problema 7: Los atributos como propiedades del nodo Jugador dificultaban la implementación de los algoritmos para los sistemas de recomendación.

Solución: Se decidió crear otro nodo llamado Característica que cada jugador tuviera diferentes cualidades basándose en los atributos iniciales.

Mejoras futuras

La aplicación actualmente es bastante completa para el tiempo de desarrollo que se ha tenido, pero a lo largo del avance han surgido ideas para mejorarla los diferentes sistemas, pero la complejidad de algunas y la limitación de tiempo, no ha sido posible implementarlas. Algunas de las ideas son las siguientes.

1. En la visualización del perfil de los jugadores, hacer alguna conexión con alguna **API de fútbol** que nos proporcione estadísticas actuales del jugador en ese momento, como podría ser los partidos que lleva jugados, goles que ha marcado, asistencias, tarjetas, entre otro tipo de información que sería útil para los usuarios conocerla, así tendrían una visión de la situación en este mismo instante del jugador, que están ojeando.
2. Añadir **imágenes para las banderas de la nacionalidades, escudos de equipo y foto de perfil a los jugadores**, esta idea es simplemente visual pero que ayudaría a los usuarios a poner cara al jugador que están ojeando y también mayor facilidad para reconocerlo.
3. Añadir **más características** a los jugadores para mejorar el sistema de recomendación, la aplicación actual cuenta con unas 29 características que pueden tener los jugadores, la idea sería añadir más para que los jugadores pudieran ser más distintos entre ellos y pudiendo así mejorar la eficacia de recomendación del sistema.
4. Crear un sistema de **registro e inicio de sesión**, al poder tener diferentes usuarios, el sistema como está diseñado almacenaría sus visitas a perfiles y a los jugadores que añade a favoritos, pudiendo hacer con esto mejores sistemas de recomendación basados en las preferencias de los usuarios que usan la aplicación.

Lecciones aprendidas

Desde mi punto de vista, siendo una persona que me gustan los retos y poder superarme, el desarrollo de esta aplicación me ha hecho aprender muchas cosas no sólo en el apartado de conocer nuevas tecnologías sino también a la hora de resolver los problemas y metodologías de trabajo, ya que en otras asignaturas existe mucha menos libertad a la hora de trabajar ya que el proyecto suele estar definido por un guión y unos pasos a seguir, en cambio aquí todo el desarrollo de la aplicación queda a elección del alumno, que por un lado tiene sus partes positivas como puede ser fomentar la creatividad o aprender a afrontar las dificultades que surgen y por otro lado las negativas como puede ser una mayor cantidad de trabajo y de razonamiento que se debe hacer. Por ello voy a mencionar las que para mí han sido más importantes y que me pueden servir en un futuro cuando me enfrente a algún problema similar.

Resolución de problemas, durante la realización de la aplicación me han ido surgiendo diferentes dificultades que era necesario resolverlas para poder continuar con el proyecto, por lo que me he ido adaptando o cambiando ideas que inicialmente tenía en mente para poder sacar el proyecto adelante sin quedarme atascado, dando como resultado la aplicación que vemos ahora mismo.

Centrarse en lo importante, tanta libertad que nos da esta asignatura nos permite que cada uno trabaje a su ritmo haciendo más o menos lo que cada uno quiere, pero es fundamental tener un poco de orden para que las cosas no se descontrolen, por lo que es necesario realizar un **plan de trabajo** para ir realizando las tareas semana a semana, esta es otra lección que he aprendido, el procedimiento que he tomado fue inicialmente preparar como iba a ir desarrollando el proyecto de manera general y luego planear las tareas a realizar durante la semana, si alguna tarea estaba incompleta se pasaba a la semana siguiente, teniendo así un orden y poder llegar a la fecha de entrega.

Neo4j, esta nueva tecnología me ha ido sorprendido más y más según iba conociendo más de ella, acostumbrado a las bases de datos de SQL y su representación mediante tablas, la forma de representar mediante grafos de Neo4j es simplemente espectacular, ya que te da la capacidad de ver toda la información y sus relaciones de una manera simple y sencilla. Además Cypher el lenguaje utilizado para las consultas es muy parecido a SQL, por lo que la adaptación a este ha sido muy fácil, aunque sí no tuviera ninguna referencia anterior el lenguaje es intuitivo y sencillo de aprender por lo que no ha supuesto ninguna dificultad.

Bibliografía y referencias

- Base de datos:
<https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset?select=Career+Mode+player+datasets+-+FIFA+15-21.xlsx>
- Cursos de Neo4j:
<https://graphacademy.neo4j.com/>
- Definición de atributos de los jugadores:
<https://es.fifauteam.com/atributos-en-fifa-21/>
- Ubuntu:
<https://ubuntu.com/download>
- Node.js
<https://nodejs.org/es/>
- Neo4j
<https://neo4j.com>
- Vue.js
<https://vuejs.org/>
- Bootstrap
<https://getbootstrap.com/>
- Font Awesome
<https://fontawesome.com/>
- GitHub
<https://github.com/>
- Lista de nacionalidades en inglés y español en JSON
<https://gist.githubusercontent.com/Yizack/bbfce31e0217a3689c8d961a356cb10d/raw/107e0bdf27918adea625410af0d340e8fc1cd5bf/countries.json>
- Manual de ayuda de Cypher
<https://neo4j.com/docs/cypher-manual/current/>
- Función para obtener la edad según la fecha de nacimiento
<https://es.stackoverflow.com/questions/31373/obtener-la-edad-a-partir-de-la-f>

<https://www.echa-de-nacimiento-con-javascript-y-php/31380>

- Web donde se muestran los atributos de los jugadores
<https://sofifa.com/player/158023/lionel-messi/210002?attr=career>
- Recursos para Bootstrap y Vue
<https://bootstrap-vue.org>
- Instalación librerías en NEO4J
<https://neo4j.com/docs/graph-data-science/current/installation/neo4j-server/>
- Librería gds de NEO4J
<https://neo4j.com/download-center/#algorithms>
- Pandemia y equipos de fútbol
<https://elpais.com/economia/2021-03-06/la-pandemia-pincha-la-burbuja-economica-del-futbol.html>
https://www.eldiario.es/economia/15-clubes-futbol-espanoles-suman-350-millones-deuda-aval-pandemia_1_7955472.html
- Obtener edad con la fecha de nacimiento en un MATCH en Neo4j
<https://community.neo4j.com/t/convert-date-of-birth-property-to-age-during-match/6451/4h>
- Jaccard Similarity algorithm
<https://neo4j.com/docs/graph-data-science/current/alpha-algorithms/jaccard/>
- Euclidean Distance
<https://neo4j.com/docs/graph-data-science/current/alpha-algorithms/euclidean/>
- Atributos importantes de un jugador según la posición
[https://www.topentrenador.es/blog/recursos/tactico-tecnica/caracteristicas-jugadores-seguin-posicion/](https://www.topentrenador.es/blog/recursos/tactico-tecnica/caracteristicas-jugadores-segun-posicion/)
- Patrón de diseño: Modelo-Vista-Controlador
<http://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve/mvc-modelo-vista-controlador>
- Explicación de DAFO:
<https://www.cerem.es/blog/claves-para-hacer-un-buen-dafo-o-foda>