

Índice

Descripción del problema.....	2
Herramientas utilizadas.....	3
Aplicación.....	4
Algoritmo de recomendación.....	13
Análisis DAFO.....	14
Líneas de futuro.....	14
Lecciones aprendidas.....	15
Bibliografía.....	15

Descripción del problema

Actualmente vivimos en una sociedad donde el fútbol es el centro deportivo del mundo, la mayoría de gente del mundo ve o es apasionada del fútbol, ya sea en sus respectivas ligas, las competiciones europeas o las competiciones internacionales.

En estos últimos años se ha puesto de moda aplicaciones como la liga fantasy, el comunio o la liga biwenger que son aplicaciones donde formas tu equipo, fichando jugadores de la liga y compitiendo con otros usuarios por hacer el máximo de puntos para ganar la liga.

Esos puntos se rigen por el papel que hace el jugador en el partido, si hace buen partido y tiene buenas estadísticas sumará más puntos que si hace mal partido.

MyTeam es un proyecto creado para ayudarte a formar tu equipo dependiendo de las estadísticas de los jugadores, dejarte claro en que facetas destacan para que tengas en cuenta eso y puedas administrar mejor tu plantilla y tus puntos.

La persona que ingrese en la página web tendrá la oportunidad de crear un nuevo jugador, crear una nueva estadística y poder relacionarlos en la base de datos, así podrán tener una base más extensa y tener las cosas más claras para el momento de la selección de jugadores, nuevos fichajes o posibles ventas.

Herramientas utilizadas

En este proyecto de recomendación de jugadores se han empleado diferentes herramientas/aplicaciones:

1. Visual Studio Code: Editor de texto.
2. Neo4j: Base de datos, ya sea desde la aplicación o desde la sandbox.
3. Node.js: Es uno de los entornos de ejecución de JavaScript, diseñado para crear aplicaciones network escalables.
4. Express: framework especializado para Node.js que trabaja el backend de una página.
5. Nodemon: herramienta que resetea automáticamente la aplicación “node” (app.js en nuestro proyecto) cuando se hace cualquier cambio en los archivos.
6. Postman: programa que se utiliza para el desarrollo de API's. Útil para realizar peticiones tipo get,post, etc.. al servidor creado.

Para el diseño de esta página web he utilizado un lenguaje HTML junto a CSS.

El CSS sirve para darle un diferente diseño a la página, es decir, para modificar las imágenes, textos, scripts, botones, etc..

El archivo funcional tiene lenguaje .ejs y los demás links de la página tiene lenguaje .html.

También he utilizado JQuery y JavaScript para aumentar la accesibilidad e interactividad de la página.

Para la base de datos como ya he mencionado antes he utilizado Neo4j.

Es un software sencillo orientado a grafos. En mi caso he usado la Sandbox ya que la aplicación de Neo4j me dió algunos problemas en el ordenador.

Aplicación

En esta parte de la memoria intentaré explicar tanto con código como con imágenes de la página web.

Antes de todo, haré un breve resumen de cómo surgió la idea y cómo he ido evolucionando en el proyecto.

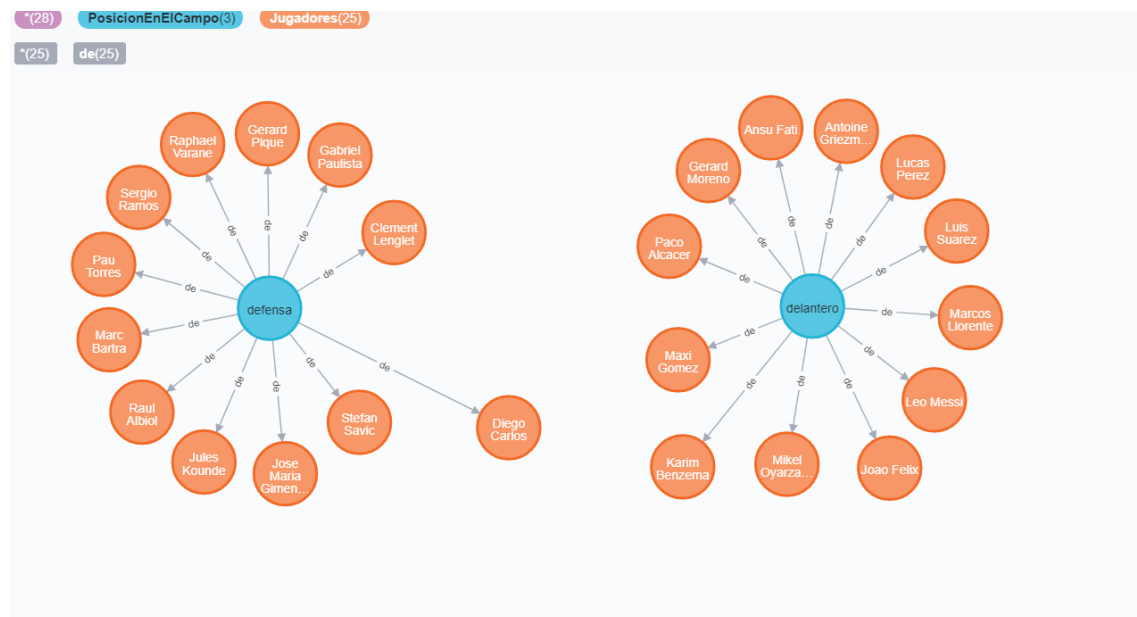
La idea me surgió porque soy un gran aficionado al fútbol y estoy metido en varias ligas con mis amigos haciendo nuestros propios equipos.

Cuando se me explico este trabajo rápidamente se me vino a la cabeza este proyecto ya que a mi siempre me surgen dudas sobre que jugador fichar y que jugadores pueden componer mi equipo.

Antes de continuar, mi conocimiento sobre Neo4j, HTML, CSS y JavaScript era nulo. He tenido suerte de encontrar varios videos introductorios para la consecución tanto de la base de datos como de la página web.

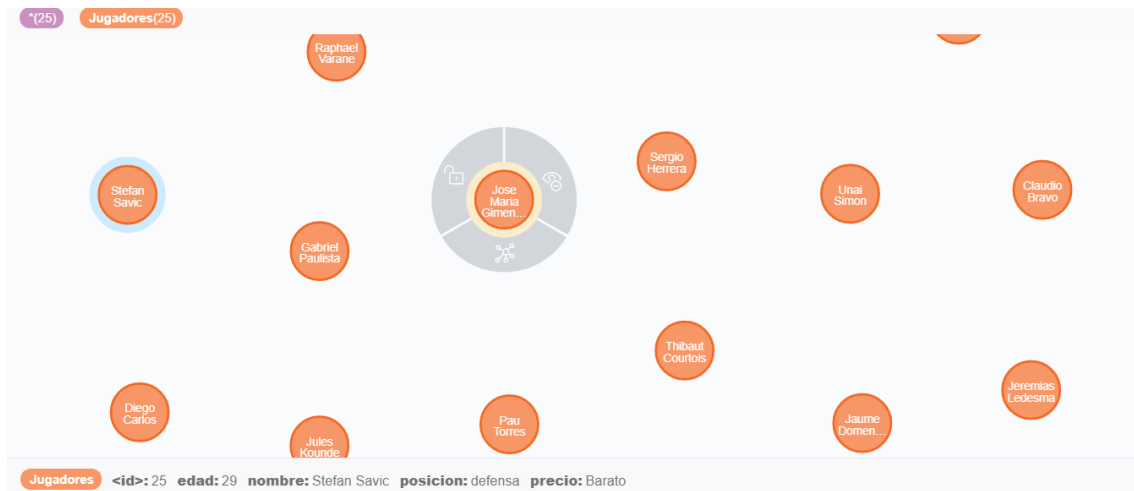
También he de decir que he tenido la suerte de contar con amigos con amplios conocimientos sobre los temas de la página y me han resultado muy útiles tanto la ayuda como sus consejos.

Primero de todo me gustaría enseñaros una parte de la base de datos que tengo:

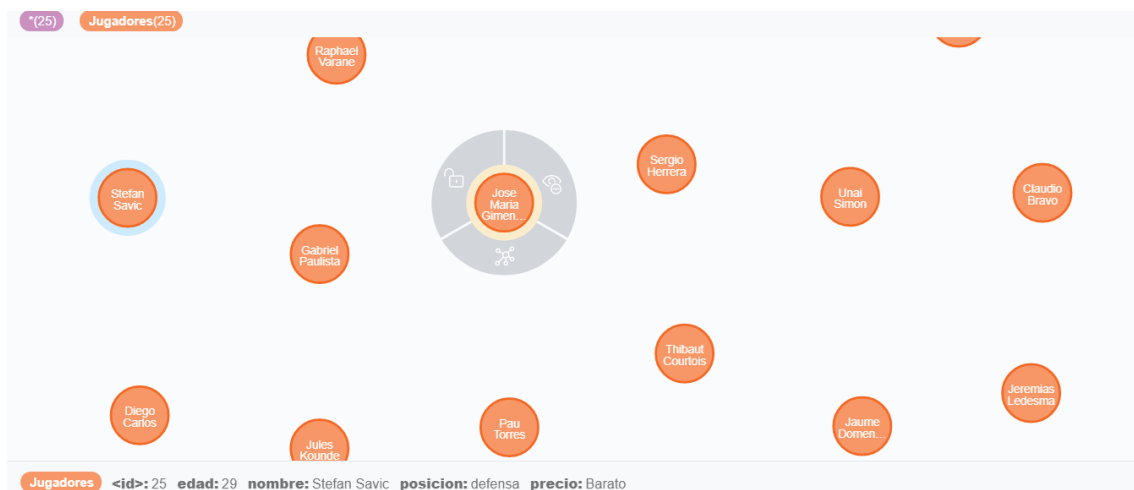


En esta imagen se muestra una parte de los jugadores relacionados con su posición.

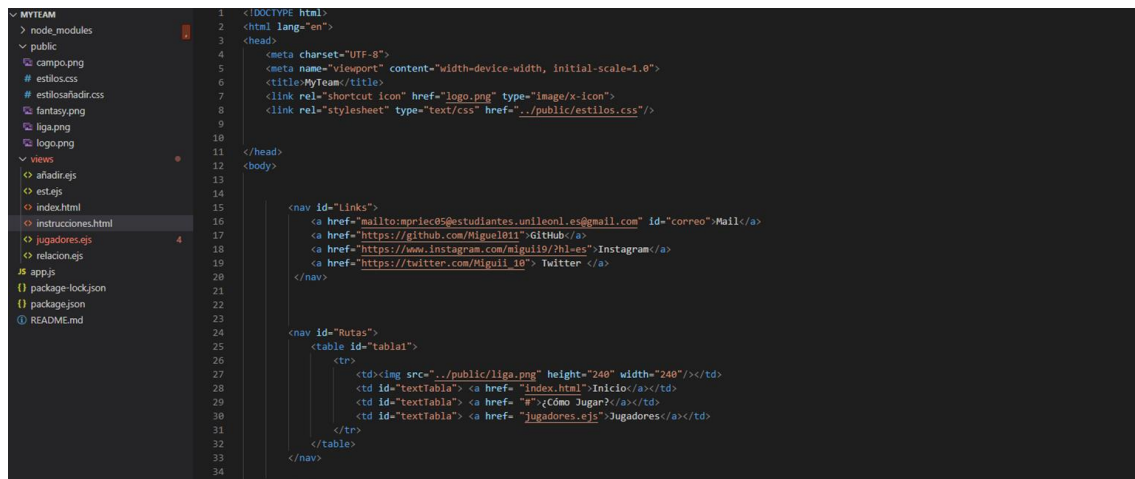
En la siguiente imagen se muestra los jugadores introducidos con sus “Property Keys”



En esta imagen se muestra las relaciones de los jugadores con sus estadísticas, ojo que esto va a ser una parte importante para lo que viene después.



A continuación lo que voy a mostrar es la parte del código HTML y su CSS. Con lo cual vamos ya con el código de la página web.

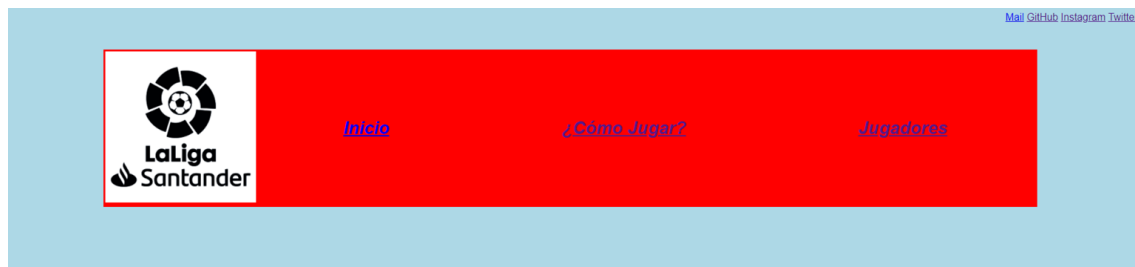


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>MyTeam</title>
7   <link rel="shortcut icon" href="logo.png" type="image/x-icon">
8   <link rel="stylesheet" type="text/css" href="public/estilos.css">
9
10 </head>
11 <body>
12
13   <nav id="Links">
14     <a href="mailto:mpriec05@estudiantes.unileon.es@gmail.com" id="correo">Mail</a>
15     <a href="https://github.com/Miguel011">GitHub</a>
16     <a href="https://www.instagram.com/miguel019/2hl-es">Instagram</a>
17     <a href="https://twitter.com/Miguel010">Twitter</a>
18   </nav>
19
20   <nav id="Rutas">
21     <table id="tabla1">
22       <tr>
23         <td></td>
24         <td id="textTabla"> <a href="index.html">Inicio</a></td>
25         <td id="textTabla"> <a href="¿Cómo Jugar?">¿Cómo Jugar?</a></td>
26         <td id="textTabla"> <a href="Jugadores.js">Jugadores</a></td>
27       </tr>
28     </table>
29   </nav>
30
31
32
33
34

```

En esta imagen se muestra el código necesario para crear tanto la parte de los links redireccionables a mi Twitter, Instagram, GitHub y mail.



Ahora vamos a verlo en la parte visual de la página web.
Lo siguiente que vamos a ver son unas imágenes de las partes más “explicativas” de la página y su respectivo código.

Esta es la parte visual donde se ven las páginas de inicio (imagen derecha) y la página de como jugar (imagen izquierda).



Ahora enseñaremos el código de ejemplo de imagen y texto.

```
<div id="container">
  <table id="tabla3">
    <tr>
      <td>
        
      </td>
      <td>
        <h1 id="titulo1">DEBERÍAS SABER...</h1>
        <p id="text2">
          Para el funcionamiento de esta página web es necesario estar en una liga compitiendo como
          por ejemplo "la liga fantasy" de Marca. Existen otro tipo de paginas que hacen lo mismo, simplemente
          escoge la que más te guste y ¡A Jugar!. <br/>
          En este tipo de ligas hay que estar muy atento al mercado para fichar jugadores para tu equipo por lo cual
          esta página te ofrece un sistema de búsqueda para fichar los jugadores en torno a tus expectativas.
          Al principio del juego te dan un dinero limitado por el cual tienes que escoger ciertos jugadores para formar tu
          plantilla y esta página (entre otras cosas) te hace una búsqueda de los jugadores correspondiente al precio que quieras
          gastar y a las estadísticas que tengan. <br/>
          ¡SOLO QUEDA TENER EL MEJOR EQUIPO Y GANAR LA LIGA A LOS DEMÁS USUARIOS!
        </p>
      </td>
    </tr>
  </table>
</div>
```

Y a continuación mostraremos parte del CSS.

```
#Links{
  float: right;
  color: red;
}

body{
  background: lightblue;
  font-family: Arial, Helvetica, sans-serif;
  border: transparent 1px solid;
}

#tabla1 tr:hover {
  background: red;
}

#tabla1 td:hover{
  background: rgb(182, 2, 2);
  color: #fff;
}

#Rutas{
  background: red;
  margin-top: 60px;
  margin-right: 160px;
  margin-left: 160px;
}
```

Aquí se muestra el código CSS que hace que podamos modificar el diseño de nuestra página web.

Por ejemplo en la parte de #Links, llamamos a esa referencia del HTML para poder modificar su posición (con el float) y su color de letra.

Ahora os voy a comentar un poco mi parte funcional, dónde están las recomendaciones y las demás funciones que tengo.

Primero de todo os voy a hacer un desglose de lo que hay:

1. Lo primero que tengo es una parte de añadir jugador.
Esto servirá para uso personal, agrandando la base de datos, dándole valores como su nombre, edad, precio y posición, y esto se quedará guardado en la base de datos para que puedas consultarla.
2. Lo segundo es una parte de añadir estadística.
Si piensas que puedes añadir una estadística que no está puesta en la página la puedes añadir a la base de datos con el nombre que quieras, por ejemplo: veloz, provocador de penaltis, etc..
3. Lo tercero es una parte de poder relacionar ese jugador que añadas con esa estadística que has añadido, por ejemplo si añades el jugador “Joaquín” y añades la estadística “Veloz” esta función te permite relacionarlos.
4. Y por último hay un select dinámico donde puedes seleccionar las estadísticas medidas en la base de datos.
Un ejemplo es, elegir la estadística “Portería a 0” y que en la pantalla te salga aquellos jugadores dónde su estadística sea esa.

A continuación os pondré y os explicaré parte del código JavaScript, que en mi caso es la parte que más me ha costado hacer por mis pocos conocimientos acerca de este lenguaje.

También tengo que decir que varios videos/páginas me han ayudado a entenderlo de forma principiante.


```

<h3> Añadir Jugador </h3>
<form method="post" action="/jugadores/add">
  <label>Nombre del Jugador</label> <br>
  <input type="text" name="nombre">
  <br>
  <label>Edad</label> <br>
  <input type="text" name="edad">
  <br>
  <input type="text" name="precio">
  <br>
  <input type="text" name="posicion">
  <br>
  <input type="submit" value="Añadir Jugador">
</form>

app.post("/jugadores/add", function(req, res){
  var nombre = req.body.nombre;
  var edad = req.body.edad;
  var precio = req.body.precio;
  var posicion = req.body.posicion;

  session2
    .run("CREATE (n:Jugadores {nombre:{nombreParam}, edad:{edadParam}, precio:{precioParam}, posicion:{posicionParam}})", {nombreParam:nombre, edadParam:edad,
    .then(function(result){
      res.render("añadir", {
        nombre: nombre,
        edad: edad,
        precio: precio,
        posicion: posicion
      });
    })
    .catch(function(err){
      console.log(err);
    })
  });
});

```

Este código lo tengo en la parte de jugadores.ejs donde hago un formulario para meter los select para meter luego los datos donde añadir un jugador.

de

creo

luego me voy al app.js a meter la Query necesaria para hacer funcional esta parte.

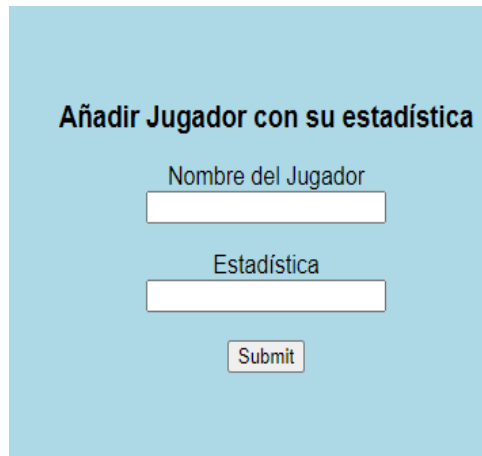
En esta imagen muestro la parte del app.js donde le doy funcionalidad con la base de datos.

En esta imagen muestro el resultado final la operación.

Las parte de añadir estadísticas es lo mismo pero en vez de situarse en jugadores.ejs me sitúo en añadir.ejs donde el formulario para añadir una estadística y

```
app.post( "/jugadores/estadisticas/add" , function(req, res){  
  
  var nombre1 = req.body.nombre1;  
  var nombre2 = req.body.nombre2;  
  console.log(nombre1)  
  console.log(nombre2)  
  session2  
  .run("MATCH (a:EstadisticasJugadores {nombre:{nombreParam2}} MATCH (b:Jugadores {nombre:{nombreParam1}}) create (b)-[:cualidad]->(a) ", {nombreParam1:nombre1,  
  .then(function(result){  
    res.render("añadir", {  
    });  
  })  
  .catch(function(err){  
    console.log(err);  
  });  
});
```

Aquí muestro la parte del código donde relaciono tanto la parte de añadir jugador con la parte de añadir estadística donde meto el Query necesario para la relación y mando hacer un render que me enseñe en la pantalla la interacción. Dicha interacción también la he modificado haciendo el form en la parte de añadir.ejs, por eso el render está dirigido a esa extensión.



Añadir Jugador con su estadística

Nombre del Jugador

Estadística

Submit

La siguiente parte corresponde con el algoritmo de recomendación.

```
app.post('/enseñarEstadistica' , function(req, res){
  var estadistica = req.body.estadistica;
  console.log(estadistica)
  session2
  .run("MATCH (a:EstadisticasJugadores {nombre:{estadisticaParam}})-[:cualidad]-(n:Jugadores) RETURN n", {estadisticaParam:estadistica})
  .then( function(result){
    estadisticasArray=[];
    result.records.forEach(function(record){
      estadisticasArray.push({
        id: record._fields[0].identity.low,
        name: record._fields[0].properties.nombre
      })
    })
    console.log(estadisticasArray[0].name)
    res.render('jugadores',{
      estadisticas:estadisticasArray
    })
  })
  .catch(function(err){
    console.log(err);
  });
});
})
```

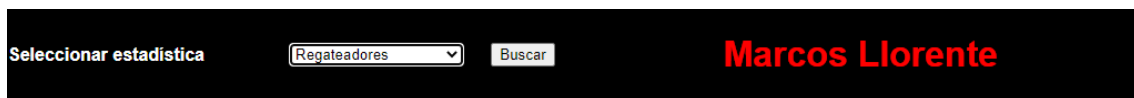
```
<form method= "post" action="/enseñarEstadistica">
  <label><p>Seleccionar estadística</p></label>
  <select name="estadistica" class="custom-select my-1 mr-sm-2" id="inlineFormCustomSelectPref" aria-labelledby="navbarDropdown" >
    <option value="">Choose...</option>
    <option value="Paradas">Paradas</option>
    <option value="Entradas">Entradas</option>
    <option value="Despejes">Despejes</option>
    <option value="Balones Recuperados">Balones Recuperados</option>
    <option value="Porteria a 0">Porteria a 0</option>
    <option value="Asistencias">Asistencias</option>
    <option value="Goleadores">Goleadores</option>
    <option value="Regateadores">Regateadores</option>
  </select>
  <br><br>
  <button type="submit" class="btn btn-primary my-1" id="elemento">Buscar</button>
</form>
```

```
<script>
  var myVar = <%- JSON.stringify(estadisticas) %>;

  console.log(myVar[0].name + "este es el nombre")
  for(let i=0;i<myVar.length;i++){
    printnombre=myVar[i].name;
    console.log(printnombre)
    document.getElementsByClassName("random")[i].innerHTML=printnombre;
  }
  console.log(myVar.length)
</script>
```

Esto es el código donde la primera imagen corresponde con la parte de app.js y la segunda corresponde al script introducido en el jugadores.ejs. En el primer código se crea un array para meterlo en el script y un render que nos tiene que devolver en pantalla el nombre del jugador. El script recorre el array a través de un for en el cual encuentra la estadística deseada y lo recorre hasta encontrar los jugadores que tengan esa estadística.

En esta imagen os muestro el select dinámico que cree con las estadísticas que se muestran en la imagen las cuales seleccionare una y se hará lo que se puso anteriormente, es decir, devolverá en pantalla los nombres de los jugadores con esas estadísticas.



También podrás ver las “keys” de los jugadores en la base de datos Neo4j donde puedes ver si el jugador es caro/barato, su edad y su posición. Esto lo explico en el vídeo que he creado para explicar mi página web.

Aquí os voy a mostrar la parte funcional pero sin el css, lo que viene siendo el primer prototipo.



[Inicio](#)
[¿Cómo Jugar?](#)
[Jugadores](#)

Añadir Jugador

Nombre del Jugador

Edad

Posición

Precio

Añadir Estadística

Estadística deseada del Jugador

Añadir Jugador con su estadística


Nombre del Jugador

Estadística

Seleccionar estadística

Marcos Llorente

Y ahora una imagen con la misma parte pero con el css introducido.



[Inicio](#)
[¿Cómo Jugar?](#)
[Jugadores](#)

Añadir Jugador

Nombre del Jugador

Edad

Posición

Precio

Añadir Estadística

Estadística deseada del Jugador

Añadir Jugador con su estadística

Nombre del Jugador

Estadística

Seleccionar estadística

Marcos Llorente

Algoritmo de Recomendación

El algoritmo de recomendación utilizado para esta práctica se basa en el filtrado de las estadísticas de los jugadores. Es un filtrado colaborativo donde las recomendaciones se hacen dependiendo de las estadísticas preferidas por el usuario o por sus gustos.

Creo que este algoritmo es idóneo para este proyecto ya que esta página cuenta con los selectores de las estadísticas que es lo fundamental en los jugadores de fútbol que utilizamos en este caso, es decir, para poder usar nuestros conocimientos luego en las apps externas donde competimos con amigos u otros usuarios.

Para hacer el sistema de recomendaciones simplemente hay que elegir las estadísticas deseadas y luego mirar en la base de datos si las “keys” son de nuestro grupo, es decir, si el jugador es caro pues igual no me interesa comprarlo o me interesa buscar otro jugador con la misma estadística pero barato. Esto también es adaptable a la edad del jugador la cual también se muestra en la base de datos.

Análisis DAFO



También debo añadir que como desventajas tenía otros proyectos grandes este cuatrimestre y he estado escaso de tiempo para poder hacer la página web.

Líneas de futuro

- ⑩ Hacer una página web viable para una base de datos más grande.
- ⑩ Convertir la página accesible para móviles y tablets.
- ⑩ Mejorar el algoritmo de recomendación para poder elegir más estadísticas o más variables como el precio, edad o posición.

Lecciones aprendidas

Una de las cosas más importantes que he aprendido gracias a este proyecto ha sido que la ambición a veces puede jugar una mala pasada. Para esta página web he querido hacer ciertas cosas que por falta de tiempo no he podido llegar a realizar.

Otra lección aprendida es que en internet siempre hay información/videos que pueden ayudarte a la consecución del proyecto.

Ha sido un proyecto importante en estos años de carrera que llevo y me siento realizado por haberlo realizado a tiempo.

Bibliografía

<https://bertofern.wordpress.com/2019/01/08/solucion-node-js-npm-no-reconocido-como-comando-interno-o-externo/>

https://www.youtube.com/watch?v=snjnJCZhXUM&t=1964s&ab_channel=TraversyMedia

<https://www.youtube.com/watch?v=34-SUNmhIOU>

https://www.youtube.com/watch?v=k3h_y9w_7l4&t=516s

<https://www.youtube.com/watch?v=p3y5o-n7AcA&t=316s>