



Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA



MEMORIA DEL PROYECTO NBAnalyzer

Autor:
Omar Arias Fernández (44435655T)

CURSO 2020/2021

ÍNDICE

INTRODUCCIÓN	3
DESCRIPCIÓN DEL PROBLEMA.....	3
HERRAMIENTAS	4
APLICACIÓN DESARROLLADA	4
<i>FRONT-END</i>	4
<i>BACK-END</i>	9
<i>BASE DE DATOS</i>	10
ANÁLISIS DE RESULTADOS.....	11
<i>EJECUCIÓN 1: BÚSQUEDA POR FILTROS DE PARÁMETROS ESPECÍFICOS</i>	11
<i>EJECUCIÓN 2: BÚSQUEDA POR RECOMENDACIÓN EN BASE AL ESTILO DE JUEGO</i>	12
ALGORITMO USADO PARA LA RECOMENDACIÓN DE LOS JUGADORES	14
ALGORITMO DEL JUGADOR CON ESTILO '3&D'	14
ALGORITMO DEL JUGADOR CON ESTILO 'PLAYMAKER'	14
ALGORITMO DEL JUGADOR CON ESTILO 'DEFENSIVE REBOUNDER'	15
ALGORITMO DEL JUGADOR CON ESTILO 'OFFENSIVE REBOUNDER'	15
ALGORITMO DEL JUGADOR CON ESTILO 'DEFENSIVE PLAYER'	15
ALGORITMO DEL JUGADOR CON ESTILO 'ALL-AROUND PLAYER'	16
DAFO (Debilidades, amenazas, fortalezas y oportunidades)	17
LÍNEAS DE FUTURO	18
LECCIONES APRENDIDAS	18
BIBLIOGRAFÍA Y REFERENCIAS.....	19

INTRODUCCIÓN

En el siguiente documento se procederá a la realización de la memoria del proyecto correspondiente al desarrollo de la aplicación **NBAAnalyzer**, una aplicación web con formato de aplicación de escritorio que sirva como herramienta de ayuda para las franquicias pertenecientes a la *National Basketball Association* (NBA).

La finalidad de esta aplicación es el análisis de las estadísticas individuales de cada jugador para ofrecer al equipo que haga uso de esta herramienta el jugador que mejor se adapta a las necesidades que tiene en función de una serie de parámetros que debe introducir dentro de la aplicación.

Dentro de la aplicación se encontrará un apartado más técnico donde el usuario podrá solicitar una recomendación de jugadores basada en unos parámetros numéricos como pueden ser por ejemplo posición, edad, puntos por partido o porcentaje de acierto en tiros de campo. Este apartado está destinado a ese usuario que tiene claro el tipo de jugador que necesita de una forma más específica. Pero para esas franquicias que no tienen tan claro esos datos estadísticos del usuario la aplicación dispone de un apartado para la recomendación de jugadores por parte del sistema en función de su estilo de juego.

A continuación, se explicará más a fondo el funcionamiento de la aplicación, así como las tecnologías que usa, los algoritmos de recomendación de jugadores, el problema que resuelve, un DAFO y las líneas de futuro por las que se puede seguir ampliando este proyecto.

DESCRIPCIÓN DEL PROBLEMA

En los últimos 30 años la NBA ha pasado a convertirse en un producto a nivel mundial sin precedentes en el baloncesto. La tarea realizada durante la presidencia de David Stern como presidente de la NBA hizo que esta liga fuese más allá de las fronteras estadounidenses y tuviese visibilidad más allá de las fronteras estadounidenses. Todo esto lleva a que los equipos ansíen ganar el anillo que otorga la NBA al campeón de la liga, haciendo que estos equipos busquen tener los mejores jugadores para proclamarse campeones.

Y aquí llega el problema que se intenta solucionar con esta aplicación, ya que antes del inicio de cada temporada se juega un partido muy importante que es el periodo de mercado de traspasos, donde los equipos juegan las cartas que tienen para posteriormente tener el mejor equipo durante la temporada. Con esta aplicación se busca ayudar a los equipos de la NBA a identificar posibles jugadores que hagan mejor al equipo para poder reclutarlos, ya que es una tarea bastante difícil y toda la ayuda que pueda tener un equipo es muy necesaria para conseguir a ese jugador que puede marcar diferencias en tu equipo y llevarte a lo más alto.

La aplicación inicialmente se diseña para que, con los datos que dispone de los jugadores, ofrezca a los mejores jugadores que encajarían en ese perfil que necesita el equipo.

HERRAMIENTAS

Como mencionamos en la introducción, estamos hablando de una aplicación web con aspecto de aplicación de escritorio.

Para desarrollar el código a nivel de Front-end, hemos hecho uso de la tecnología **Vue** acompañada de otra tecnología llamada **Electron JS**, la cual nos permite que la aplicación tenga ese aspecto de aplicación de escritorio que mencionamos.

Para desarrollar el código a nivel de back-end y servidor, se ha realizado con la tecnología **Node JS**, con el lenguaje JavaScript. Desde este apartado es donde se realizan las peticiones correspondientes a la base de datos donde se encuentran los datos de los jugadores.

Para la base de datos, se ha decidido usar una innovadora tecnología de base de datos como es **Neo4j**, un sistema de bases de datos basada en grafos de conocimiento, donde los datos se almacenan como nodos y se unen mediante una serie de relaciones que se establecen entre nodos.



APLICACIÓN DESARROLLADA

La aplicación tiene dos funciones principales: Recomendación de jugadores basada en estadísticas técnicas del mismo introducidas por el usuario o recomendación de jugadores basada en el estilo de juego del mismo propuestas por el sistema.

En el primer caso, es el usuario quien tiene el control sobre qué parámetros del usuario quiere hacer hincapié, disponiendo en la interfaz de campos de texto donde introducir los valores mínimos en cada una de las estadísticas que desee.

En el segundo caso, el usuario solo tendrá capacidad de decisión sobre el estilo de juego, número de jugadores a mostrar y posición quiere obtener, dejando las recomendaciones en manos del sistema. A continuación, se desglosará la aplicación en 3 apartados: Front-end, Back-end y Base de datos.

FRONT-END

En la interfaz de usuario encontramos un total de 3 vistas, a las cuales se accederá en función de la búsqueda que desee realizar. Una vez se arranca la aplicación, se accede a la vista principal donde se encontrará los dos tipos de recomendación expuestos anteriormente, con una serie de campos de texto para introducir los datos de búsqueda y un par de botones para hacer las búsquedas.



En el primer caso de recomendación el usuario dispone de una serie de campos 'input' para que el sistema sepa qué jugadores buscar. Estos son la posición, edad, porcentaje de tiros de campo, porcentaje de tiros de 3, porcentaje de tiros libres, puntos por partido, rebotes por partido y asistencias por partido. El usuario decidirá cuáles de ellos rellenar o no. En caso de que no le interese algún parámetro de los expuestos podrá realizar la búsqueda igualmente, pero el parámetro de la posición debe ser introducido de forma obligatoria.

Para el segundo caso de recomendación, se dispone de 3 *ComboBox*, donde el usuario deberá escoger un estilo de los que dispone la aplicación, los cuales son:

- **3&D** – Jugador que se caracteriza por tener un acierto elevado en tiros exteriores y por ser un gran defensor perimetral y en algunos casos interior.
- **Playmaker** – Se denomina al jugador que dirige a un equipo, normalmente siendo el base (PG) o escolta (SG) del equipo. Se caracteriza por atesorar el mayor tiempo de la posesión el balón en sus manos y tener un número elevado de asistencias y puntos.
- **Defensive rebounder** – Se denomina así al jugador que promedia un número alto de rebotes defensivos por partido. Normalmente suelen ser jugadores que actúan en la posición de alero (SF), ala-pívot (PF) o pívot (C).
- **Offensive rebounder** – Se denomina así al jugador que promedia un número alto de rebotes ofensivos por partido. Normalmente suelen ser jugadores que actúan en la posición de alero (SF), ala-pívot (PF) o pívot (C).
- **Defensive player** - Se denomina así al jugador que tiene un gran nivel de defensa perimetral con un número alto de robos, rebotes defensivos y tapones.
- **All-around player** - Se denomina así al jugador que tiene la capacidad de jugar en varias posiciones del campo. Por ejemplo, un jugador que actúa como base pero que dispone de gran altura y tiene estadísticas propias de un pívot puede considerarse un *All-around player* ya que tiene las características para jugar en esa posición. En el caso de un pívot, puede considerarse un *All-around player* cuando posee un número de asistencias y puntos por partido cercano o mayor a lo que suele promediar un base o escolta.

El siguiente parámetro para introducir será el número de jugadores a mostrar, en el cual se dispone de las siguientes opciones:

- Top 3
- Top 10
- Top 15
- Top 25

Hay que destacar en este apartado que en el caso de que se seleccione un número de jugadores a mostrar mayor al número de jugadores que existan, se mostrará la totalidad de los jugadores disponibles. Finalmente se introducirá la posición a buscar. En este caso será obligatorio introducir los 3 campos de búsqueda.

También se contará dentro de la interfaz con un botón en la parte superior derecha que sirve para ver las últimas noticias relacionadas con la NBA, haciendo que salte una ventana emergente que llevará hacia la página oficial de noticias de la NBA.

NBAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

Bienvenido a NBAAnalyzer, la aplicación donde podrás encontrar el jugador que necesitas. Nuestro sistema de recomendación basado en los parámetros indicados te ofrecerá una serie de opciones que se adecúen a los mismos.

PARÁMETROS DE BÚSQUEDA:

POSICIÓN

☐ PG

☐ SG

☐ SF

☐ PF

☐ C

EDAD

Ej. 27

% TIROS DE CAMPO

Ej. 0.39

% TIROS DE 3 PUNTOS

Ej. 0.34

% TIROS LIBRES

Ej. 0.87

PUNTOS POR PARTIDO

Ej. 16

REBOTES POR PARTIDO

Ej. 4

ASISTENCIAS POR PARTIDO

Ej. 8

BUSCAR JUGADORES

O si no tienes claro lo que buscas, podemos ofrecerte algunos jugadores en función de su estilo de juego:

Escoge uno de las siguientes estilos

Escoge el número de jugadores

Escoge la posición del jugador

RECOMENDAR JUGADORES

¡Repórtanos con errores o posibles mejoras del sistema!

2020 - Curso 20/21 Sistemas de información de gestión y business intelligence

AYUDA

Figura 1: Vista de la página principal de NBAAnalyzer

Dado que las vistas de ambos casos de búsqueda son iguales, se procederá a explicar ambas juntas y se expondrá mediante figuras un caso de ejecución de ambas.

Al fondo de la ventana principal dispondremos de un botón de ayuda donde se explicará también el funcionamiento de la aplicación para los usuarios principiantes que nunca la hayan usado.

En el momento que accionemos el botón de búsqueda (Tanto el de buscar jugadores como el de recomendar jugadores) el programa hará comprobaciones a ver si se dispone de los parámetros mínimo solicitados para la recomendación de jugadores. En caso de no tenerlas se mostrará una ventana emergente de error. A la hora de realizar la búsqueda, se deben introducir los parámetros **mínimos** que debe poseer el jugador, es decir, si metemos 0.35 en la casilla de porcentaje de tiros de campo, buscaremos los jugadores que tengan ese mismo porcentaje o mayor. Esta regla se aplica a todas las casillas excepto a la edad, que será menor o igual a la introducida.

NBAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

Bienvenido a NBAAnalyzer, la aplicación donde podrás encontrar el jugador que necesitas. Nuestro sistema de recomendación basado en los parámetros indicados te ofrecerá una serie de opciones que se adecúen a los mismos.

PARÁMETROS DE BÚSQUEDA:

POSICIÓN

☐ PG

☐ SG

☐ SF

☐ PF

☐ C

EDAD

Ej. 27

% TIROS DE CAMPO

Ej. 0.39

% TIROS DE 3 PUNTOS

Ej. 0.34

% TIROS LIBRES

Ej. 0.87

PUNTOS POR PARTIDO

Ej. 16

REBOTES POR PARTIDO

Ej. 4

ASISTENCIAS POR PARTIDO

Ej. 8

BUSCAR JUGADORES

O si no tienes claro lo que buscas, podemos ofrecerte algunos jugadores en función de su estilo de juego:

Escoge uno de las siguientes estilos

Escoge el número de jugadores

Escoge la posición del jugador

RECOMENDAR JUGADORES

¡Repórtanos con errores o posibles mejoras del sistema!

2020 - Curso 20/21 Sistemas de información de gestión y business intelligence

AYUDA

¡BÚSQUEDA INCOMPLETA!

Debes introducir al menos la posición del jugador que quieres buscar

Aceptar

Figura 2: Caso de búsqueda errónea de jugadores



NBAAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

Bienvenido a NBAAnalyzer, la aplicación donde podrás encontrar el jugador que necesitas. Nuestro sistema de recomendación basado en los parámetros indicados te ofrecerá una serie de opciones que se adecuen a los mismos.

PARÁMETROS DE BÚSQUEDA:

POSICIÓN: PG, SG, SF, PF, C

EDAD: Ej. 27

PUNTOS POR PARTIDO: Ej. 16

% TIROS DE CAMPO: Ej. 0.39

% TIROS DE 3 PUNTOS: Ej. 0.34

% TIROS LIBRES: Ej. 0.87

REBOTES POR PARTIDO: Ej. 4

ASISTENCIAS POR PARTIDO: Ej. 8

STEALS POR PARTIDO: Ej. 0.1

BUSCAR JUGADORES

¡BÚSQUEDA INCOMPLETA!
Debes introducir el estilo del jugador, número de jugadores y posición a buscar.

Aceptar

RECOMENDAR JUGADORES

¡Repórtanos con errores o posibles mejoras del sistema!

2020 - Curso 20/21 Sistemas de información de gestión y business intelligence

Figura 3: Caso de recomendación errónea de jugadores

NBAAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

Bienvenido a NBAAnalyzer, la aplicación donde podrás encontrar el jugador que necesitas. Nuestro sistema de recomendación basado en los parámetros indicados te ofrecerá una serie de opciones que se adecuen a los mismos.

PARÁMETROS DE BÚSQUEDA:

POSICIÓN: PG, SG, SF, PF, C

EDAD: Ej. 27

PUNTOS POR PARTIDO: Ej. 16

% TIROS DE CAMPO: Ej. 0.39

% TIROS DE 3 PUNTOS: Ej. 0.34

% TIROS LIBRES: Ej. 0.87

REBOTES POR PARTIDO: Ej. 4

ASISTENCIAS POR PARTIDO: Ej. 8

STEALS POR PARTIDO: Ej. 0.1

BUSCAR JUGADORES

RECOMENDAR JUGADORES

¡Repórtanos con errores o posibles mejoras del sistema!

Para comenzar a usar esta aplicación, tan solo tendrás que introducir los parámetros sobre el jugador que estás buscando para que el programa comience a contrastar con la base de datos para darte las mejores opciones que haya en el mercado de jugadores de la NBA. El único requisito obligatorio para comenzar una búsqueda es que se introduzca una posición.

En la parte inferior disponemos de un sistema de búsqueda de jugadores en función de su estilo de juego, para los que no tienen un perfil claro de jugador a buscar. Deberá introducir de forma obligatoria todos los parámetros para comenzar la búsqueda.

CERRAR VENTANA

Figura 4: Ventana emergente con instrucciones de uso

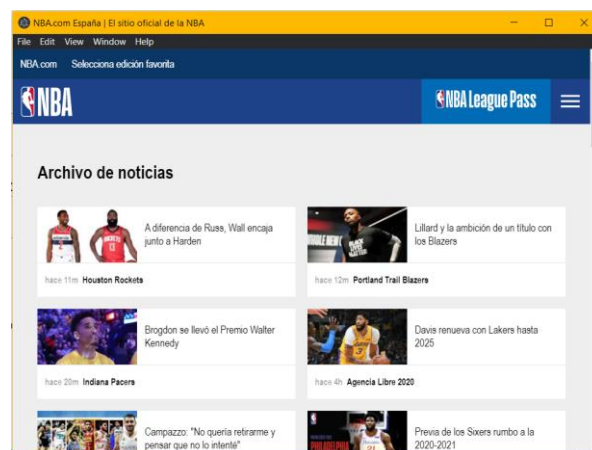


Figura 5: Ventana emergente de las noticias de la NBA



En el caso de tener los parámetros mínimos, el sistema enviará esas estadísticas a la siguiente vista, para que desde esta se proceda a solicitar dichos datos a la base de datos. Una vez obtiene los jugadores de la base de datos, se muestran en el siguiente formato:

NBAAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

VOLVER A LA PÁGINA PRINCIPAL

Nombre	Edad	POS	Equipo	SALARY (\$)	PPG	RPG	ORB	DRB	SPG	TPG	APG	FG%	FT%	3P%	FPG
Joel Embiid	25	C	PHI	\$27,504,630	23.4	11.8	2.8	9	0.9	3.1	3.1	0.474	0.814	0.348	3.4
Karl-Anthony Towns	24	C	MIN	\$27,285,000	26.5	10.8	2.7	8.1	0.9	3.1	4.4	0.508	0.796	0.412	3.3
Rudy Gobert	27	C	UTA	\$25,258,427	15.1	13.7	3.4	10.3	0.8	1.9	1.5	0.698	0.621		3.2
Andre Drummond	26	C	CLE	\$27,093,018	17.7	15.2	4.4	10.8	1.9	3.6	2.7	0.533	0.575	0.143	3.5
Clint Capela	25	C	HOU	\$16,896,552	13.9	13.8	4.3	9.5	0.8	1.6	1.2	0.629	0.529		2.6
Deandre Ayton	21	C	PHO	\$9,562,920	19	12	4.1	7.9	0.7	2.4	1.9	0.548	0.769	0	3.2
Domantas Sabonis	23	C	IND	\$3,529,554	18.5	12.4	3	9.4	0.8	2.7	5	0.54	0.723	0.254	3.1
Hassan Whiteside	30	C	POR	\$27,093,018	16.3	14.2	4	10.2	0.4	1.9	1.2	0.618	0.68	0.571	2.9

Rows per page: 15 1-8 of 8

¡Repórtanos con errores o posibles mejoras del sistema!

2020 – Curso 20/21 Sistemas de información de gestión y business intelligence

AYUDA

Figura 6: Caso de búsqueda mediante datos estadísticos específicos

NBAAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

Estos son los mejores Playmaker de la liga en la posición de SG

VOLVER A LA PÁGINA PRINCIPAL

Nombre	Edad	POS	Equipo	SALARY (\$)	PPG	RPG	ORB	DRB	SPG	TPG	APG	FG%	FT%	3P%	FPG
James Harden	30	SG	HOU	\$38,199,000	34.4	6.4	1	5.3	1.7	4.5	7.4	0.435	0.861	0.352	3.4
Jrue Holiday	29	SG	NOP	\$26,131,111	19.6	4.9	1.3	3.6	1.7	2.9	6.9	0.452	0.7	0.357	2.3
Spencer Dinwiddie	26	SG	BRK	\$10,605,600	20.6	3.5	0.5	3	0.6	2.7	6.8	0.415	0.778	0.308	2.8
Devin Booker	23	SG	PHO	\$27,285,000	26.1	4.2	0.4	3.7	0.7	3.9	6.6	0.487	0.916	0.36	3
Bradley Beal	26	SG	WAS	\$27,093,018	30.5	4.2	0.9	3.3	1.2	3.4	6.1	0.455	0.842	0.353	2.2
Marcus Smart	25	SG	BOS	\$12,553,571	13.5	3.8	0.8	3.1	1.6	1.7	4.8	0.38	0.835	0.348	2.7

Rows per page: 15 1-6 of 6

¡Repórtanos con errores o posibles mejoras del sistema!

2020 – Curso 20/21 Sistemas de información de gestión y business intelligence

AYUDA

Figura 7: Caso de búsqueda mediante recomendación del sistema

Como se puede observar, las diferencias entre ambas vistas son mínimas, simplemente cambia la consulta que se realiza a la base de datos en cada una de ellas, de las cuales se hablará en el apartado del Back-end.

BACK-END

En el back-end se hace la comunicación de la interfaz de usuario con la base de datos. Como ya mencionamos antes, se hace uso de la tecnología **Node JS** la cual nos permite tener un script donde capturar las peticiones mediante el protocolo HTML y transformarlas en consultas a la base de datos.

Dentro de esta tecnología se hace uso de una serie de extensiones que son necesarias para el correcto funcionamiento de la aplicación, las cuales son:

- Express - Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.
- Body-parser – Middleware que sirve para analizar el cuerpo de las solicitudes entrantes a Node JS permitiendo detectar parámetros introducidos en la petición.
- Neo4j Driver – Driver que permite conectar el servidor Node JS con una base de datos de Neo4j de forma local o remota, abriendo sesiones para realizar consultas contra la base de datos.
- Axios – Cliente HTTP basado en promesas para conectar Node JS y el navegador web.

El servidor cuando comienza su ejecución escucha continuamente peticiones para actuar en caso de detectar una. Cuando la recibe, ejecuta un fragmento de código en función de la petición recibida (get, post, put...), como se muestra en el siguiente ejemplo:

```
//FUNCION PARA DEVOLVER LA MEDIA DE REBOTES
app.get("/avgRebotes", function(req, res) {
  //AQUI HACEMOS LA CONSULTA A LA BASE DE DATOS
  query = "MATCH (p:Player) where p.posicion='PF' or p.posicion='C' return avg(p.rebotes_pp)"
  var lista=[]
  const consultaRebotes = driver.session();
  consultaRebotes.run(query).subscribe({
    onNext: function(result) {
      lista.push(result.get(0));
    },
    onCompleted: function() {
      res.send(lista);
      consultaRebotes.close();
    },
    onError: function(error) {
      console.log(error);
    }
  });
});
```

Figura 8: Ejemplo de consulta para obtener la media de rebotes de los jugadores

Vemos que en esta petición ejecuta la query contra la base de datos, y una vez es resuelta almacena los datos en una lista y los envía de vuelta al *Front-end*.

En este punto es donde comienza a hacerse uso del driver de Neo4j para JavaScript, haciendo de puente entre el servidor del *Back-end* y la base de datos para obtener los datos.

BASE DE DATOS

Este punto es realmente importante dentro del proyecto, ya que el entorno de base de datos que se usa es Neo4j, un entorno de desarrollo de bases de datos basado en grafos de conocimiento. Aquí no se usan las típicas tablas que se usan en el entorno de SQL o similares. Neo4j hace uso de grafos para almacenar los datos, mediante los nodos que representan los datos, con una serie de propiedades específicas de cada nodo. Estos establecen una relación con los demás nodos dentro del grafo mediante las denominadas '*Relaciones*', caracterizadas por un nombre y pudiendo alojar una serie de características de esta.

Este estilo de base de datos rompe con lo establecido por SQL con tablas, donde cada tabla debe tener una clave primaria y relacionándose con las demás tablas mediante claves foráneas. En Neo4j no se necesita tener claves primarias ni foráneas, ya que los nodos se conectarán todos entre sí, permitiendo establecer con mayor facilidad patrones de comportamiento que pueden ser útiles a la hora de obtener conclusiones.

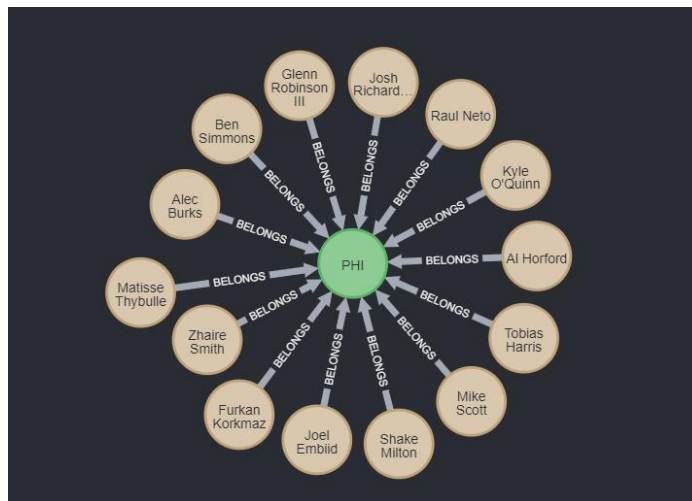


Figura 9: Ejemplo de nodos relacionados de los jugadores del equipo 'Philadelphia 76ers'

En nuestro caso partiremos de los nodos que se agrupan en dos tipos: Jugadores y equipos. Cada jugador estará unido a un nodo equipo mediante una relación de pertenencia. Dado que el sistema se centra en las estadísticas de los jugadores, los nodos equipo tendrán una importancia secundaria en nuestra base de datos.

Por cada nodo jugador se establece una serie de propiedades que almacenan las estadísticas del jugador en concreto, las cuales son:

- Nombre
- Edad
- Posición
- Valoración
- Salario
- Puntos por partido



- Rebotes por partido (Suma de rebotes ofensivos y defensivos)
- Rebotes ofensivos por partido
- Rebotes defensivos por partido
- Porcentaje de acierto en tiros de campo
- Porcentaje de acierto en tiros de 2 puntos
- Porcentaje de acierto en tiros libres
- Porcentaje de acierto en tiros perimetrales (Triples)
- Minutos disputados por partido
- Partidos jugados
- Partidos como titular
- Robos por partido
- Pérdidas de balón por partido
- Tapones por partido
- Faltas realizadas por partido

Todos los parámetros mencionados anteriormente son los que se usarán para filtrar los dos tipos de búsqueda, ya que en función de ellos podemos vislumbrar el estilo de juego que tiene cada uno.

ANÁLISIS DE RESULTADOS

Para mostrar la ejecución del programa vamos a hacer dos ejecuciones: la primera correspondiente a la búsqueda más técnica y la segunda a la búsqueda por estilo del jugador.

EJECUCIÓN 1: BÚSQUEDA POR FILTROS DE PARÁMETROS ESPECÍFICOS

Partiremos desde la pantalla principal de la aplicación una vez que se arranca. Supondremos un caso de un equipo en concreto, los Boston Celtics. Poseen un equipo donde tienen buenos jugadores perimetrales con gran acierto en triples, pero tienen escasa presencia interior, ya que solo disponen de un pívot con nivel para jugar de titular y apenas llega a los 2.10 metros. Todo equipo que se precie en esta liga debe tener al menos un pívot potente en el que confiar la tarea de recoger rebotes, anotar y taponar a los rivales.

Bien, podrían hacer uso de la aplicación para buscar a un pívot joven que promedie un buen número de puntos y rebotes por partido, y para ello se introducirá los parámetros mínimos del jugador que quieren incorporar.

POSICIÓN	EDAD Ej. 27	% TIROS DE CAMPO Ej. 0.39	% TIROS DE 3 PUNTOS Ej. 0.34	% TIROS LIBRES Ej. 0.87
<input type="radio"/> PG	<input type="text" value="25"/>	<input type="text" value="Ej. 0.39"/>	<input type="text" value="Ej. 0.34"/>	<input type="text" value="Ej. 0.87"/>
<input type="radio"/> SG				
<input type="radio"/> SF	PUNTOS POR PARTIDO Ej. 16	REBOTES POR PARTIDO Ej. 4	ASISTENCIAS POR PARTIDO Ej. 8	<input type="button" value="BUSCAR JUGADORES"/>
<input type="radio"/> PF	<input type="text" value="13"/>	<input type="text" value="8"/>	<input type="text" value="Ej. 8"/>	
<input checked="" type="radio"/> C				

Figura 10: Pantalla principal con los parámetros de búsqueda introducidos

Para esta búsqueda el *Staff técnico* de los Boston Celtics ha averiguado que la media de puntos de un pívot en la Liga es de 9.19 puntos por partido y la media de rebotes es de 6.31 por partido. Una vez planteado el escenario sabemos que Boston necesita un jugador por encima de la media que tenga unos mínimos de garantía, entonces para ello procedemos a buscar un jugador que tenga unas estadísticas por encima de esa media y que además sea joven para que tenga proyección y pueda mejorar en la franquicia.

Para esta búsqueda aplicamos un filtro de que el jugador tenga un promedio en puntos mayor a 13 y de rebotes mayor a 8, además de que tenga una edad menor a 25 y sea *Center* (Pívot).

NBAAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA

VOLVER A LA PÁGINA PRINCIPAL

Nombre	Edad	POS	Equipo	SALARY (\$)	PPG	RPG	ORB	DRB	SPG	TPG	APG	FG%	FT%	3P%	FPG
Joel Embiid	25	C	PHI	\$27,504,630	23.4	11.8	2.8	9	0.9	3.1	3.1	0.474	0.814	0.348	3.4
Karl-Anthony Towns	24	C	MIN	\$27,285,000	26.5	10.8	2.7	8.1	0.9	3.1	4.4	0.508	0.796	0.412	3.3
Rudy Gobert	27	C	UTA	\$25,258,427	15.1	13.7	3.4	10.3	0.8	1.9	1.5	0.698	0.621		3.2
Andre Drummond	26	C	CLE	\$27,093,018	17.7	15.2	4.4	10.8	1.9	3.6	2.7	0.533	0.575	0.143	3.5
Clint Capela	25	C	HOU	\$16,896,552	13.9	13.8	4.3	9.5	0.8	1.6	1.2	0.629	0.529		2.6
Deandre Ayton	21	C	PHO	\$9,562,920	19	12	4.1	7.9	0.7	2.4	1.9	0.548	0.769	0	3.2
Serge Ibaka	30	C	TOR	\$23,271,604	16	8.3	2.1	6.2	0.5	2	1.5	0.518	0.748	0.398	2.8
Domantas Sabonis	23	C	IND	\$3,529,554	18.5	12.4	3	9.4	0.8	2.7	5	0.54	0.723	0.254	3.1
Hassan Whiteside	30	C	POR	\$27,093,018	16.3	14.2	4	10.2	0.4	1.9	1.2	0.618	0.68	0.571	2.9

Rows per page: 15 1-9 of 9

¡Repórtanos con errores o posibles mejoras del sistema!

2020 – Curso 20/21 Sistemas de información de gestión y business intelligence

Figura 11: Resultados de la búsqueda de los pívots para los Boston Celtics

Una vez realizamos la búsqueda, vemos que la lista anterior de jugadores cumple los requisitos que hemos planteado anteriormente. Aquí se ofrece más información de cada uno de ellos, pudiendo valorar pros y contra de cada uno y valorar la viabilidad de un posible traspaso con la franquicia a la que pertenece. El orden en el que aparecen ordenados es mediante la valoración que posee, que es una media de los valores de cada atributo de jugador proporcionada por el videojuego NBA 2K20.

EJECUCIÓN 2: BÚSQUEDA POR RECOMENDACIÓN EN BASE AL ESTILO DE JUEGO

Para este caso vamos a situarnos en la posición de la franquicia de Pennsylvania, los *Philadelphia 76ers*. Todos los aficionados tanto a este equipo como al resto de franquicias de la liga son conscientes de las limitaciones que sufre este equipo para obtener puntos desde la línea de tiro de 3 puntos, ya que tienen un gran poderío a nivel de puntuación en la zona, pero en el perímetro no gozan de gran acierto. También tienen la idea de reforzar la defensa perimetral ya que encajan muchos puntos de tiros de triple y eso los condena a muchas derrotas.

Pero debemos tener en cuenta además que los *76ers* cuentan con un núcleo joven formado por el base australiano Ben Simmons y el pívot camerunés Joel Embiid, titulares indiscutibles,



con lo cual el jugador ideal a fichar deberá ocupar la posición de alero que es la posición menos reforzada que tienen.

Sabiendo todo esto el equipo identifica que el perfil de jugador que necesita es un estilo conocido en la NBA como un 3&D (Triples y defensa). Este tipo de jugador se caracteriza por tener un buen porcentaje de tiros de 3 puntos y gozar de una defensa perimetral bastante aceptable. Por ello podrían usar la aplicación para buscar un jugador de este estilo por la posición que buscan en concreto.

O si no tienes claro lo que buscas, podemos ofrecerte algunos jugadores en función de su estilo de juego:

Escoge uno de los siguientes estilos

3&D

Escoge el número de jugadores


Top 25

Escoge la posición del jugador


SF

RECOMENDAR JUGADORES

Figura 12: Parámetros de búsqueda para el jugador 3&D de los Philadelphia 76ers



NBAnalyzer

ÚLTIMAS NOTICIAS EN LA NBA 

Estos son los mejores 3&D de la liga en la posicion de SF

VOLVER A LA PÁGINA PRINCIPAL

Nombre	Edad	POS	Equipo	SALARY (\$)	PPG	RPG	ORB	DRB	SPG	TPG	APG	FG%	FT%	3P%	FPG
Khris Middleton	28	SF	MIL	\$30,603,448	21.1	6.2	0.7	5.5	0.9	2.1	4.1	0.499	0.908	0.418	2.2
Paul George	29	SF	LAC	\$33,005,556	21	5.7	0.5	5.2	1.3	2.7	3.9	0.432	0.882	0.399	2.4
Gordon Hayward	29	SF	BOS	\$32,700,690	17.3	6.5	1	5.5	0.8	1.8	4.1	0.502	0.847	0.392	1.8
Jaylen Brown	23	SF	BOS	\$6,534,829	20.4	6.4	1.1	5.3	1.1	2.3	2.2	0.49	0.736	0.381	3
OG Anunoby	22	SF	TOR	\$2,281,800	10.7	5.4	1.2	4.2	1.4	1.2	1.6	0.507	0.686	0.381	2.5
Will Barton	29	SF	DEN	\$12,776,786	15.1	6.3	1.3	5	1.1	1.5	3.7	0.45	0.767	0.375	2.1
Kawhi Leonard	28	SF	LAC	\$32,742,000	26.9	7.3	1	6.3	1.8	2.7	5	0.469	0.889	0.366	1.9
Josh Hart	24	SF	NOP	\$1,934,160	10.2	6.5	0.9	5.5	1	1.1	1.6	0.416	0.747	0.342	2.5
Taurean Prince	25	SF	BRK	\$3,481,985	12.1	6	0.8	5.2	0.9	2	1.8	0.376	0.798	0.339	2.5
Jonathan Isaac	22	SF	ORL	\$5,806,440	12	6.9	1.8	5.2	1.6	1.5	1.4	0.463	0.767	0.33	2.5

Rows per page: All 1-10 of 10 < >

Figura 13: Resultados de la búsqueda de los 3&D para los Philadelphia 76ers

Una vez realizamos la búsqueda, vemos que la lista anterior de jugadores cumple los requisitos que hemos planteado anteriormente. Aquí se ofrece más información de cada uno de ellos, pudiendo valorar pros y contra de cada uno y valorar la viabilidad de un posible traspaso con la franquicia a la que pertenece. Estos jugadores cumplen una serie de requisitos especificados en el algoritmo de la búsqueda, explicado en el correspondiente apartado.

Como se puede observar, esta aplicación se puede usar en varios escenarios en los que se puede encontrar una franquicia, aconsejando un fichaje que puede devolverlos a la élite del baloncesto estadounidense.

ALGORITMO USADO PARA LA RECOMENDACIÓN DE LOS JUGADORES

En este apartado nos centraremos más en la funcionalidad de recomendación de jugadores ya que para la búsqueda por filtrado de datos el algoritmo es simple, solo hay que buscar los jugadores que tienen valores mayores a los introducidos en los respectivos campos de texto.

A continuación, se desglosará uno a uno los algoritmos explicando su complejidad y como realiza las búsquedas. Antes de proceder a ello cabe destacar que los algoritmos están limitados a los datos con los que contamos en la base de datos. Probablemente de contar con una base de datos con estadística avanzada de cada uno de los jugadores el algoritmo sería más exacto, pero la estadística avanzada en la NBA suele ser de fuentes privadas y teniendo que pagar por los datos, lo cual no es apropiado para este proyecto.

ALGORITMO DEL JUGADOR CON ESTILO '3&D'

Como ya mencionamos anteriormente, un jugador **3&D** se caracteriza por un buen acierto en triples y una defensa perimetral consistente. Con las estadísticas que tenemos de cada jugador en la base de datos podemos determinar si un jugador se puede considerar un 3&D cuando en cada uno de los campos de defensa y tiro perimetral tiene unos buenos números.

Por tanto, primero realizamos un escaneo a todos los jugadores de la base de datos buscando los datos de rebotes defensivos y robos por partido (En el apartado defensivo) y de porcentaje de tiro de triple (En el apartado de acierto de triple) y hallamos la media de cada uno de estos campos con los valores de los jugadores que son aleros, ala-pívots y pívots (Ya que son los jugadores que mejores datos tienen defensivamente), determinando así la media que un jugador NBA tiene en estos apartados.

Como decía, para considerar un jugador **3&D** debe ser especialmente bueno en estos campos, por tanto, la búsqueda realizada en los jugadores de la base de datos es encontrar aquellos que superan los valores medios en estos 3 datos, escogiendo así los jugadores que encajan con este estilo de juego.

ALGORITMO DEL JUGADOR CON ESTILO 'PLAYMAKER'

Como ya mencionamos anteriormente, un jugador **Playmaker** se caracteriza por ser el jugador que lleva la dirección de un equipo, llevándolo en volandas asistiendo a sus compañeros y anotando un número notable de puntos. Con las estadísticas que tenemos de cada jugador en la base de datos podemos determinar si un jugador se puede considerar un Playmaker basándonos en las asistencias y puntos que reparte por partido.

Por tanto, primero realizamos un escaneo a todos los jugadores de la base de datos buscando los datos de asistencias y puntos por partido, para que al igual que en el otro algoritmo, hallar la media de estos campos en todos los bases y escoltas de la NBA (Ya que son los que mayor número de asistencias y puntos reparten por partido), determinando así la media que un jugador NBA tiene en estos apartados.



Finalmente, para considerar un jugador **Playmaker** debe ser especialmente bueno en estos campos, por tanto, la búsqueda realizada en los jugadores de la base de datos es encontrar aquellos que superan los valores medios en estos 2 datos (media de puntos y media de asistencias), escogiendo así los jugadores que encajan con este estilo de juego.

ALGORITMO DEL JUGADOR CON ESTILO 'DEFENSIVE REBOUNDER'

Como ya mencionamos anteriormente, un jugador **Defensive rebounder** se caracteriza por ser un gran reboteador defensivo. Con las estadísticas que tenemos de cada jugador en la base de datos podemos determinar si un jugador se puede considerar un **Defensive rebounder** cuando en el campo de rebotes defensivos por partido tiene un valor por encima de la media.

Por tanto, primero realizamos un escaneo a todos los jugadores de la base de datos buscando los datos de rebotes defensivos en pívots y ala-pívots (Ya que son las posiciones en el campo que más rebotes se recogen) y hallamos la media de este dato con los valores de todos los jugadores, determinando así la media que un jugador NBA tiene en este apartado.

Como decía, para considerar un jugador **Defensive rebounder** debe tener una media de rebotes defensivos por partido bastante alta, por tanto, la búsqueda realizada en los jugadores de la base de datos es encontrar aquellos que superan los valores medios en este dato, escogiendo así los jugadores que encajan con este estilo de juego.

ALGORITMO DEL JUGADOR CON ESTILO 'OFFENSIVE REBOUNDER'

Como ya mencionamos anteriormente, un jugador **Offensive rebounder** se caracteriza por ser un gran reboteador ofensivo. Con las estadísticas que tenemos de cada jugador en la base de datos podemos determinar si un jugador se puede considerar un **Offensive rebounder** cuando en el campo de rebotes ofensivos por partido tiene un valor por encima de la media.

Por tanto, primero realizamos un escaneo a todos los jugadores de la base de datos buscando los datos de rebotes Ofensivos en pívots y ala-pívots (Ya que son las posiciones en el campo que más rebotes se recogen) y hallamos la media de este dato con los valores de todos los jugadores, determinando así la media que un jugador NBA tiene en este apartado.

Como decía, para considerar un jugador **Offensive rebounder** debe tener una media de rebotes defensivos por partido bastante alta, por tanto, la búsqueda realizada en los jugadores de la base de datos es encontrar aquellos que superan los valores medios en este dato, escogiendo así los jugadores que encajan con este estilo de juego.

Este algoritmo es similar al aplicado en el caso anterior, pero variando el dato analizado.

ALGORITMO DEL JUGADOR CON ESTILO 'DEFENSIVE PLAYER'

Para este caso usaremos un algoritmo similar al que usamos en el **3&D**, pero variando el análisis de algunos datos a analizar.

Dado que este es un apartado puramente defensivo, consideraremos analizar tres parámetros básicos en la defensa de un jugador como son los rebotes defensivos por partido, robos por partido y tapones por partido.



Por tanto, primero realizamos un escaneo a todos los jugadores de la base de datos buscando los datos de rebotes defensivos por partido, robos por partido y tapones por partido hallamos la media de cada uno de estos campos con los valores de los jugadores que son aleros, ala-pívots y pívots (Ya que son los jugadores que mejores datos tienen defensivamente), determinando así la media que un jugador NBA tiene en estos apartados.

Como decía, para considerar un jugador defensivo debe ser especialmente bueno en estos campos, por tanto, la búsqueda realizada en los jugadores de la base de datos es encontrar aquellos que superan los valores medios en estos 3 datos, escogiendo así los jugadores que encajan con este estilo de juego.

ALGORITMO DEL JUGADOR CON ESTILO 'ALL-AROUND PLAYER'

Este tipo de jugador es bastante peculiar e innovador, ya que ha ido apareciendo durante estos últimos años.

En este algoritmo dependemos de la posición que quiera buscar el usuario para determinar si es un **All-around player**. A continuación, analizaremos los distintos tipos de **All-around player** que existen en el mundo del baloncesto NBA

ALL-AROUND PLAYER PARA UN BASE/ESCOLTA

En el caso de un base o un escolta, suelen tener un número alto de asistencias y puntos, pero suelen carecer en rebotes debido a que son jugadores que no llegan a los 2 metros haciendo que esto les impida recoger todos los rebotes que desean.

Sabiendo esto, ejecutaremos una búsqueda de la media de rebotes por partido que tienen los jugadores que ocupan las posiciones de ala-pívot y pívot, por el motivo de que son los jugadores con mayor promedio de rebotes. Lo que buscamos con este dato es buscar los bases y/o escoltas que tengan un promedio de rebotes mayor a la media, ya que, si supera el valor medio, nos indica que podría desempeñar la misión de jugar en una posición más adelantada de forma bastante decente. Por tanto, la consulta contra la base de datos consta de la búsqueda de los jugadores que superen dicho valor medio de rebotes por partido de un ala pívot o pívot.

ALL-AROUND PLAYER PARA UN ALERO, ALA-PÍVOT O PÍVOT

En este caso, el prototipo de jugador **All-around player** busca ese tipo de jugador que a pesar de que ocupa posiciones más interiores dentro del campo, es capaz de repartir un número alto de asistencias y pases exitosos. Este dato nos haría ver que ese jugador podría desempeñar una función de base o escolta para la cual es necesaria una gran visión del juego y capacidad de pensamiento ya que son los dos jugadores que llevan el control de los ataques de su equipo, por tanto, deben saber gestionar los tiempos y escoger los pases correctos para que su equipo anote canasta. También necesitaremos saber el número medio de puntos debido a que suele ser bastante alto en estas dos posiciones porque suelen ser los jugadores considerados 'clutch' y que se juegan el mayor número de tiros a final de posesión.



Sabiendo esto, se ejecutará una búsqueda sobre los valores medios de asistencias y puntos de un base y de un escolta dentro de los jugadores de la NBA, usándolo posteriormente de valor mínimo para determinar si un jugador es apto para ser considerado 'All-around player'.

DAFO (Debilidades, amenazas, fortalezas y oportunidades)

Para empezar, comenzaremos dando una breve introducción de lo que es un DAFO. Un DAFO se puede considerar como una matriz, la cual está formada por 4 elementos clave que permite hacer un análisis de un proyecto determinado. Los 4 puntos mencionados anteriormente son Debilidades, amenazas, fortalezas y oportunidades. Mediante la definición de estos podemos identificar distintos elementos que pueden provocar que nuestra aplicación mejore, haciéndola más completa, fuerte y eficiente.

DEBILIDADES: La principal limitación que posee esta aplicación son los datos de los jugadores y por ello, el algoritmo. Las estadísticas de las que se dispone para este proyecto son muy básicas, lo cual hace que el algoritmo merme considerablemente, pudiendo perder exactitud a la hora de evaluar a los jugadores. En caso de disponer de datos sobre estadística avanzada de los jugadores y equipos, podría mejorarse la implementación del algoritmo haciendo de NBAAnalyzer un producto muy potente e interesante.

AMENAZAS: Quizás no existan amenazas muy conocidas, ya que a nivel público no existe un software como este. Esto se debe a que los *Staff* de cada equipo controlan los datos mayoritariamente con bases de datos enlazadas a hojas de cálculo, pero no disponen, al menos de forma declarada en los medios, de un software como tal, simplemente existen empresas privadas que venden los datos a los equipos, pero en formato plano de tablas de base de datos estándar y sin una interfaz de usuario con la que interaccionar.

FORTALEZAS: La principal fortaleza que tiene esta aplicación sigue un poco el hilo de lo comentado en los puntos anteriores. En la actualidad no existen muchas aplicaciones de este estilo a nivel público, lo que otorga a NBAAnalyzer un punto extra de exclusividad que hace que pueda llamar la atención de más gente.

OPORTUNIDADES: La principal oportunidad de esto se encuentra en la carencia de amenazas del punto anterior, ya que públicamente no existen aplicaciones de este estilo muy potentes y demandadas, lo cual abre una línea de negocio para aplicaciones de este estilo. Pero como se expresaba anteriormente, el futuro de esta aplicación se basa en los datos de los que se disponga.



LÍNEAS DE FUTURO

En cuanto a las líneas de futuro que posee esta aplicación, podemos destacar varios puntos principales en los que podría centrarse la expansión de esta aplicación, tanto a nivel de ofrecer otro tipo de funcionalidades como el de mejorar las ya existentes:

- MEJORA DE LOS ALGORITMOS DE IDENTIFICACIÓN DE ESTILO:

Probablemente esta sea la línea de futuro más urgente a implementar dentro de la aplicación, dado que los algoritmos implementados en esta versión del software no son todo lo complejos y exactos que se desea debido a la precariedad de los datos. Como se ha mencionado en los apartados anteriores, de contar con una base de datos más extensa y específica se puede conseguir que esta aplicación vaya más lejos de ser un simple proyecto de una asignatura universitaria.

- IMPLEMENTACIÓN DE FUNCIONALIDADES COMPLEMENTARIAS A LA RECOMENDACIÓN DE FICHAJES

Para ser el mejor equipo de toda la liga no solo necesitas tener los mejores jugadores, si no complementarlos y hacerlos jugar de la manera más eficiente y cómoda para ellos. Esto hace que los datos de los que se dispone de cada jugador puedan ser implementados además para el desarrollo de tácticas eficientes en función de los jugadores que se disponga en el equipo. Esto haría de NBAlyzer una aplicación mucho más completa y funcional.

LECCIONES APRENDIDAS

Viendo el proyecto desde el punto de vista actual respecto al inicio del curso, el trabajo realizado hasta ahora ha sido bastante satisfactorio a nivel personal, dado que la elección de esta asignatura y la realización de este proyecto han hecho que conozca una nueva tecnología como son las bases de datos de grafos de conocimiento de Neo4j. Quizás en este proyecto no se le haya podido sacar todo el potencial del que dispone esta tecnología, pero estoy seguro que de cara al futuro y con un correcto uso de la misma se pueden hacer proyectos muy interesantes usando esta tecnología.

También quiero destacar el lenguaje Cypher, el cual permite hacer las consultas a la base de datos de Neo4j. Tras trabajar con él, considero que es un lenguaje muy útil a nivel de trabajo con bases de datos y recomiendo a todo el que pueda aprender al menos los conceptos más básicos que tiene.



BIBLIOGRAFÍA Y REFERENCIAS

Tecnología empleada para el desarrollo del Front-end (Vue JS):

<https://vuejs.org/>

Plugin para dar aspecto de aplicación de escritorio (Electron JS):

<https://www.electronjs.org/>

Tecnología empleada para el desarrollo del Back-end (Node JS):

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

Tecnología empleada para la creación de la base de datos (Neo4j):

<https://neo4j.com/>

Tecnologías implementadas en el servidor JavaScript:

<https://neo4j.com/developer/javascript/>

<https://expressjs.com/es/>

<https://www.npmjs.com/package/body-parser>

<https://www.npmjs.com/package/axios>

Fuente de la base de datos de los jugadores NBA (Kaggle):

<https://www.kaggle.com/stormex/nba-1920-season-stats-ratings-and-salaries>

Análisis DAFO: Qué es y cómo aplicarlo:

<https://anaivars.com/analisis-dafo-que-es/>

Imágenes usadas a lo largo del documento:

<https://www.wolfpress.co/que-es-back-end-y-front-end/>

<https://neo4j.com/>

<https://geekwall.in/p/HybaFAV0Q/building-a-budget-manager-with-vue-js-and-node-js-i>