

---

# PROYECTO FINAL 1º TRIMESTRE BASES DE DATOS NOSQL

- . - . -

## ANÁLISIS DE DATOS NBA PLAYER OF THE WEEK 2020/21

---

### ÍNDICE

<b>1. INTRODUCCIÓN.</b>	<b>2</b>
<b>2. CONTENIDO.</b>	<b>3</b>
2.1 Colección.	3
2.2 Tratamiento de la información.	4
2.2.1 Métodos find()	4
2.2.2 Métodos aggregate()	5
2.2.3 Método distinct()	6
<b>3. REFERENCIAS.</b>	<b>6</b>

# 1. INTRODUCCIÓN.

Uno de los campos del mundo real donde las bases de datos están más extendidas es sin duda en el campo del deporte. Los datos, su recolección y su uso son completamente vitales para su desarrollo, desde los primeros pasos, donde poder que tus ojeadores puedan acceder a los datos de un jugador en categorías juveniles puede significar diferencias sustanciales a la hora de construir un equipo con las mejores piezas posibles, hasta la manera en la que los datos obtenidos por su equipo de analistas durante un partido pueden ayudar a decidir al entrenador de un partido qué es necesario entrenar para mejorar el estado de forma de un equipo.

Es importante destacar que esta importancia llega incluso más allá de los propios jugadores y cuerpo técnico de un equipo, y han permeado hasta tener un rol clave en las noticias acerca de un equipo, los premios que se conceden en cada deporte e incluso la visión acerca de un jugador, un equipo o incluso una liga que el espectador tiene está claramente influenciada por aquellos datos a los que los medios que consumen otorgan más valor y aquellos a los que no se les otorga el suficiente.

Pero si hay un espacio donde la recolección y el análisis de los datos por parte de los medios deportivos es en las retransmisiones de los eventos deportivos. Muchos de estos eventos, como los partidos de fútbol, baloncesto o béisbol se extienden durante varias horas, y es difícil para los comentaristas mantener el interés del espectador. Es por esto que en muchos casos, se apoyan en datos obtenidos a tiempo real, tanto del partido en sí (como por ejemplo cuando en los partidos de fútbol se habla de la posesión de ambos equipos y su rol en el partido) tanto a niveles históricos (como cuando se habla de que cierto jugador ha alcanzado una cantidad de goles, asistencias o partidos que le hace estar en un puesto concreto a nivel histórico).

En este trabajo se unirán varias de estas facetas tratando los datos de la liga americana de baloncesto, la NBA (probablemente la liga de baloncesto más importante del mundo), en concreto aquellos de los galardonados como Player Of The Week, galardón otorgado por la NBA una vez por semana durante la liga regular al mejor jugador de cada conferencia durante esa semana. En específico, se usarán los datos de la temporada 2020/2021, ya que es la última que concluyó previamente a la realización de este trabajo.

## 2. CONTENIDO.

### 2.1 Colección.

Para este proyecto, utilizaremos diversos datos acerca de jugadores de la NBA galardonados con el premio "Player of the Week" a lo largo de la temporada pasada.

Estos datos están estructurados de la siguiente forma:

```
"_id":1, --> (tipo número, es el identificador del objeto)
nombre: "Brandon Ingram", --> (tipo string)
posicion:["AL"], --> (array de strings)
equipo: "New Orleans Pelicans", --> (tipo string)
dorsal:14, --> (tipo número)
fechapotw:[new Date ("2020-12-28")], --> (tipo Fecha)
trayectoria:["Los Angeles Lakers", "New Orleans Pelicans"], -->
  (array de strings)
allstar:false, --> (tipo booleano)
altura:{pies: 6, pulgadas: 7}, (documento, formado por valores de
  tipos numero)
draft: {universidad:"Duke", año:2016, eleccion:2}, --> (documento
  formado por valores tanto de tipo string como de tipo número)
nacionalidad: ["EEUU"] --> (array de strings)
```

Siendo cada dato lo siguiente:

- nombre: nombre del jugador.
- posicion: posición que ocupa el jugador en el campo. Puede ser más de una, por lo que es un array. Se utilizan las siguientes abreviaturas: BA (base), ES (escolta), AL (alero), AP (ala pívot) y P (pívot).
- equipo: equipo en el que jugó el jugador durante la temporada 2020/2021.
- dorsal: Dorsal con el que compite el jugador.
- fechapotw: fechas en las que el jugador consiguió el premio a mejor jugador de la semana. Pueden ser múltiples.
- allstar: booleano que indica si el jugador participó en el All-Star Game de la temporada 2020/2021.
- altura: altura del jugador, dividida en pies y pulgadas, pues es la medida utilizada en los Estados Unidos.
- draft: año en el que el jugador participó en el draft de la NBA, posición en la que fue seleccionado, y en el caso de los jugadores que accedieron a través de la liga universitaria, universidad en la que estudiaban/competían.
- nacionalidad: países de los cuales el jugador tiene nacionalidad.

## 2.2 Tratamiento de la información.

Para el tratamiento de la información, llevaremos a cabo dos tipos de métodos, métodos de tipo “find” y métodos de tipo “aggregate”.

### 2.2.1 Métodos find()

El método “find()” nos permite, mediante los operadores de consulta, ubicar datos dentro de la base de datos.

En este proyecto se han usado:

- Los operadores de comparación:
  - \$eq: Devuelve aquellos documentos en los que el valor del campo al que se aplica sea **exactamente** el valor dado.
  - \$gt: Devuelve aquellos documentos en los que el valor del campo al que se aplica sea **mayor estrictamente** al valor dado.
  - \$gte: Devuelve aquellos documentos en los que el valor del campo al que se aplica sea **mayor o igual** al valor dado.
  - \$lt: Devuelve aquellos documentos en los que el valor del campo al que se aplica sea **menor estrictamente** al valor dado.
  - \$lte: Devuelve aquellos documentos en los que el valor del campo al que se aplica sea **menor o igual** al valor dado.
  - \$ne: Devuelve todos los documentos de la colección **salvo** aquellos donde el valor sea **exactamente** el valor dado.
- Los operadores de consulta mediante arrays:
  - \$in: Devuelve aquellos documentos en los que el valor del campo **se encuentra** entre aquellos proporcionados mediante un array.
  - \$nin: Devuelve aquellos documentos en los que el valor del campo **no se encuentra** entre aquellos proporcionados mediante un array. Esto incluye los casos en los que el campo no existe.
- Los operadores lógicos:
  - \$and: Dadas dos (o más) expresiones de búsqueda, devuelve aquellos documentos en los que se cumplan **ambas** (todas las) **expresiones**.
  - \$nor: Dadas dos o más expresiones de búsqueda, devuelve aquellos documentos en los que **no** se cumpla **ninguna expresión**.
  - \$or: Dadas dos expresiones de búsqueda, devuelve aquellos documentos en los que se cumplan **cualquiera de las expresiones**.
  - \$not: Dada una expresión de búsqueda, devuelve aquellos documentos en los que **no se cumple** esta expresión.
- El operador \$exists: Devuelve aquellos documentos en los que el campo al que se aplica exista en el documento, es decir, tenga algún valor.
- El operador \$regex: Permite buscar patrones en datos de tipo string.

- Los operadores en arrays:
  - \$elemMatch: Dadas dos o más expresiones acerca de los elementos de un array, devuelve todos aquellos documentos en los que **un solo elemento** del array **cumpla todas** las expresiones.
  - \$all: Devuelve aquellos documentos en los que todos los componentes del array proporcionado formen parte del array que es valor del campo proporcionado.
- El operador \$size: Devuelve aquellos documentos en los cuales el campo al que se aplica es un array con el número de elementos que se introduce.

Aparte, se ha utilizado el concepto de variables, que permite crear una entidad de cualquier tipo con un nombre concreto ajena a la base de datos, para luego poder utilizarla de forma más cómoda.

### 2.2.2 Métodos aggregate()

Las operaciones de agregación procesan registros de datos y devuelven resultados obtenidos mediante el procesamiento de estos datos. Las operaciones de agregación pueden agrupar valores de varios documentos y pueden realizar una variedad de operaciones en los datos agrupados para devolver un solo resultado.

Esto se hace mediante diversos métodos y etapas. En este proyecto se han utilizado los siguientes:

- \$group: Agrupa los datos a partir de una expresión, la cual se almacena como “\_id”, que es único para cada grupo, y crea un documento para cada uno de estos grupos. A esos grupos se les pueden añadir más campos cuyos valores provengan de operaciones. Las usadas en este proyecto son las siguientes:
  - \$sum: Suma valores de tipo número. Si se le proporciona un valor numérico en vez de un campo de tipo número del documento original, multiplica esa cantidad por el número de documentos del grupo.
  - \$addToSet: Crea un array con todos los valores de un campo concreto de los documentos agrupados.
  - \$count: Devuelve un valor de tipo número equivalente al número de documentos que hay en el grupo. Es equivalente a { \$sum : 1 }.
  - \$toInt: Convierte cualquier valor de tipo número a entero.
  - \$avg: Calcula la media de todos los datos ofrecidos.
- \$sort: Ordena todos los documentos que obtiene según uno de sus campos.

### 2.2.3 Método distinct()

Con este método, se obtiene un array con todos los valores distintos que tiene un campo concreto a lo largo de la colección.

## **3. REFERENCIAS.**

Los datos utilizados en la base de datos fueron obtenidos en:

<https://basketball.realgm.com/nba/awards/by-type/Player-Of-The-Week/30/2021>

Los datos específicos de cada jugador (como dorsal o universidad de procedencia) fueron obtenidos en las páginas de cada jugador en Basketball RealGM.

Los datos acerca del All-Star Game de la temporada 2020/2021 fueron obtenidos en:

[https://es.wikipedia.org/wiki/All-Star\\_Game\\_de\\_la\\_NBA\\_2021](https://es.wikipedia.org/wiki/All-Star_Game_de_la_NBA_2021)

Los datos acerca de qué equipo está en cada conferencia fueron obtenidos en:

<https://www.nba.com/standings>

Toda la información acerca de los comandos de mongoDB fue obtenida en la propia documentación de mongoDB, accesible desde:

<https://docs.mongodb.com/>