

Integrantes:

- Daniel Guzmán
- Mayumi Tamura

## Método de la Ingeniería

### Contexto problemático

La Federación Internacional de Baloncesto (FIBA) ha decidido consolidar en una aplicación, los datos de mayor relevancia de cada uno de los profesionales del baloncesto en el mundo, de manera que se puedan realizar diferentes consultas que permitan un análisis de estos datos, se conozcan patrones acerca del desarrollo del deporte, los criterios que toman más fuerza o, en general, hacia dónde se dirige el deporte en la actualidad.

### Paso 1. Identificación del Problema

#### Identificación de necesidades

- La FIBA requiere que sea posible hacer diferentes consultas para analizar mejor los datos consolidados en la aplicación y conocer patrones del desarrollo del deporte.
- La herramienta debe manejar información de gran tamaño, recibir más datos de manera masiva ó a través de la interfaz y asimismo permitir eliminar y modificar.
- Debe ser posible realizar consultas sobre los jugadores a través de cuatro diferentes criterios de búsqueda con base en los rubros estadísticos de los jugadores.
- La aplicación deberá guardar dicha información en la memoria secundaria, además de garantizar un acceso eficiente para todas las acciones permitidas sobre la misma.

#### Definición del Problema

Una compañía desarrolladora de software requiere la implementación de una aplicación para el manejo de información de los jugadores que cumpla con la solicitud de la FIBA, así como una alta eficiencia y funcionalidad de esta.

**Requerimientos**

Nombre	RF1. Ingresar de datos.
Descripción	Permitir el ingreso de nuevos datos por el usuario, ya sea de manera masiva o a través de la interfaz.
Entrada	Archivos csv ó datos del jugador en la interfaz.
Salida	(Los datos han sido guardados)

Nombre	RF2. Eliminar datos.
Descripción	Permitir al usuario eliminar datos de los jugadores, o eliminar uno de la lista.
Entrada	Jugador a eliminar.
Salida	(Los datos han sido eliminados)

Nombre	RF3. Modificar datos.
Descripción	Permitir al usuario modificar datos de los jugadores.
Entrada	Jugador a modificar. Datos a modificar.
Salida	(Los datos han sido modificados)

Nombre	RF4. Realizar consultas de jugadores por puntos.
Descripción	Permitir al usuario realizar consultas por el criterio de búsqueda de puntos por partido de los jugadores.
Entrada	Valor buscado.
Salida	Jugadores con el valor buscado.

Nombre	RF5. Realizar consultas de jugadores por rebotes.
Descripción	Permitir al usuario realizar consultas por el criterio de búsqueda de rebotes por partido de los jugadores.
Entrada	Valor buscado.
Salida	Jugadores con el valor buscado.

Nombre	RF6. Realizar consultas de jugadores por asistencias.
Descripción	Permitir al usuario realizar consultas por el criterio de búsqueda de asistencias por partido de los jugadores.
Entrada	Valor buscado.
Salida	Jugadores con el valor buscado.

Nombre	RF7. Realizar consultas de jugadores por robos.
Descripción	Permitir al usuario realizar consultas por el criterio de búsqueda de robos por partido de los jugadores.
Entrada	Valor buscado.
Salida	Jugadores con el valor buscado.

Nombre	RF8. Realizar consultas de jugadores por bloqueos.
Descripción	Permitir al usuario realizar consultas por el criterio de búsqueda de bloqueos por partido de los jugadores.
Entrada	Valor buscado.
Salida	Jugadores con el valor buscado.

Nombre	RF9. Guardar los datos en memoria secundaria.
Descripción	La aplicación debe almacenar la información en memoria secundaria.
Entrada	
Salida	

### Requerimientos No Funcionales (RNF)

Nombre	RNF1. Garantizar eficiencia en las consultas.
Descripción	Se debe garantizar un rápido acceso a los datos, teniendo en cuenta que son los datos de millones de basquetbolistas guardados, la complejidad temporal no puede ser lineal para cuatro de los cinco criterios de búsqueda.
Entrada	
Salida	

Nombre	RNF2. Utilizar ABB para búsqueda de índices asociados.
Descripción	Se debe utilizar un árbol binario de búsqueda balanceado para acceder a los datos del jugador, para aquellos que tienen índices asociados. La estructura deberá guardar el valor del atributo y la posición en disco del jugador al que pertenece. Se debe utilizar dos AVL y dos rojinegros para los cuatro criterios de búsqueda.
Entrada	
Salida	

Nombre	RNF3. Mostrar el tiempo por consulta.
<b>Descripción</b>	Se debe mostrar el tiempo que le toma a la aplicación realizar una consulta, para justificar la elección de árboles binarios balanceados.
<b>Entrada</b>	(Realizar una consulta)
<b>Salida</b>	(El tiempo de la consulta)

Nombre	RNF4. Realizar búsquedas utilizando ABB.
<b>Descripción</b>	Se debe permitir al cliente realizar búsquedas sobre dos criterios estadísticos utilizando ABB como estructura para manejo de índices.
<b>Entrada</b>	
<b>Salida</b>	

## Paso 2. Recopilación de Información

Fuentes:

[https://es.wikipedia.org/wiki/Federaci%C3%B3n\\_Internacional\\_de\\_Baloncesto](https://es.wikipedia.org/wiki/Federaci%C3%B3n_Internacional_de_Baloncesto)

[https://es.wikipedia.org/wiki/Punto\\_\(baloncesto\)](https://es.wikipedia.org/wiki/Punto_(baloncesto))

<https://www.planetabasketball.com/baloncesto-rebote.htm>

[https://es.wikipedia.org/wiki/Rebote\\_\(baloncesto\)](https://es.wikipedia.org/wiki/Rebote_(baloncesto))

[https://es.wikipedia.org/wiki/Asistencia\\_\(baloncesto\)](https://es.wikipedia.org/wiki/Asistencia_(baloncesto))

[https://es.wikipedia.org/wiki/Robo\\_\(baloncesto\)](https://es.wikipedia.org/wiki/Robo_(baloncesto))

<https://sites.google.com/site/baloncestomoralo/Home/fundamentos-tecnicos-del-baloncesto/bloqueos>

### Definiciones

#### **FIBA**

La Federación Internacional de Baloncesto (FIBA) es el organismo que se dedica a regular las normas del baloncesto mundialmente, así como de celebrar periódicamente competiciones y eventos en sus dos disciplinas. Cuenta en 2018 con la afiliación de 213 federaciones

nacionales, divididas a su vez en 5 federaciones continentales, que son: África, América, Asia, Europa y Oceanía.

La FIBA tiene como misión organizar y coordinar numerosas competiciones de baloncesto a nivel internacional en categorías masculina y femenina, entre las que destacan: Torneo de baloncesto en los Juegos Olímpicos, Copa Mundial de Baloncesto masculino y femenino, Campeonato Mundial de Baloncesto Sub-19, Campeonato Mundial de Baloncesto Sub-17

### **Puntos**

En el baloncesto, un punto se utiliza para realizar un seguimiento de la puntuación en un partido. Los puntos pueden ser acumulados por hacer tiros de campo (dos o tres puntos) o tiros libres (un punto). Si un jugador anota un tiro de campo dentro de la línea de tres puntos, el jugador anota dos puntos. Si el jugador anota un tiro de campo de más allá de la línea de tres puntos, el jugador marca tres puntos. El equipo que ha registrado el mayor número de puntos al final de un partido es declarado ganador de ese partido.

### **Puntuación**

El objetivo del baloncesto es marcar más puntos que el equipo contrincante, encestando el balón en el cesto contrario e impidiendo a los jugadores contrincantes encestar en la propia.

Se llama canasta cuando el balón entra en el cesto del contrincante y se puntúan de la siguiente manera, una canasta lanzada desde el tiro libre vale 1 punto, desde la zona de tiro de 2 puntos vale 2 puntos y desde la zona de tiro de 3 puntos vale 3 puntos.

La puntuación se lleva mediante dos sistemas: el primero es el acta oficial del partido donde el anotador marca todas las canastas y puntos por jugador; el segundo sistema es el tablero electrónico, donde además se señala el cronómetro, el conteo de faltas de cada jugador y de cada equipo y el cuarto que se está jugando

### **Rebotes**

El rebote se indica como toda acción en procura del balón después que ha rebotado en el tablero y/o en el aro. A mayor cantidad de rebotes, más posibilidades de control de la pelota y por ende más oportunidades de encestar.

El rebote es importante no sólo para la defensa sino también en el ataque. En la defensa resta oportunidades de lanzamiento al adversario y en el ataque permite arremeter con nuevos lanzamientos.

Los rebotes están divididos en dos principales categorías: por un lado, los rebotes ofensivos, que son aquellos que recupera un equipo cuando ataca, sin cambiar la posesión del balón, y por otro, los defensivos, en los cuales el equipo que defiende gana la posesión tras un lanzamiento a canasta fallado por el oponente. El rebote defensivo se utiliza para evitar los tiros contrarios y ganar la posesión de la pelota en el propio tablero.

Los rebotes se anotan en las estadísticas a aquellos jugadores que ganan claramente la posesión tras un lanzamiento a canasta.

**Asistencias**

Una asistencia es un pase a un jugador que se encuentra en una posición de ventaja o que le ayuda a conseguir una canasta.

Las asistencias se cuentan entre las estadísticas individuales y colectivas de los equipos de baloncesto. Solamente el pase directamente anterior a la canasta se puede contar como asistencia, por lo que no se puede contabilizar más de una asistencia por canasta.

**Robos en partido**

Los robos se acreditan al jugador defensivo que causa primero la pérdida de balón, incluso si no termina con la posesión de la pelota viva. Para ganar un robo, el jugador defensivo debe ser el iniciador de la acción que causó la pérdida de balón, no sólo el benefactor. Cada vez que un robo se registra por un jugador defensivo, un jugador ofensivo debe ser acreditado con un balón perdido.

**Bloqueos**

El bloqueo es un movimiento ofensivo que se realiza para conseguir mejores situaciones y posiciones dentro del campo de juego. Es el gesto técnico que realiza un jugador atacante mediante el cual y con el uso de su propio cuerpo, permite a un compañero obtener ventaja sobre su defensor. Se puede realizar tanto al jugador con balón como al jugador sin balón.

El reglamento define al bloqueo como una "pantalla ilegal", por lo que es preciso realizarlo correctamente para no perder el balón en una falta de ataque.

Hay varios tipos de bloqueo, aunque la mecánica sea la misma para casi todos, conviene distinguir las distintas formas de efectuar un bloqueo: bloqueo de frente, bloqueo de espaldas y bloqueo lateral.

**Paso 3. Búsqueda de Soluciones Creativas**

Se realiza una lluvia de ideas donde se plantean soluciones sobre cómo podría funcionar la aplicación, la interfaz y que estructuras de datos, árboles, se podrían usar para cada criterio de búsqueda. Además, principalmente, cómo se resolvería el problema de jugadores con mismo valor para ciertos rubros estadísticos al momento de realizar una búsqueda, respecto a la eficiencia temporal.

**Clases:**

- FIBA

**Ideas planteadas:****Estructuras de datos:**

Se plantea la asignación de estructuras de datos, es decir de los árboles binarios de búsqueda, para las consultas por diferentes criterios.

Alternativa 1: Árbol RN y árbol AVL:

- Esta alternativa consiste en utilizar AVL para los criterios de puntos y rebotes por partido y usar un RN para los criterios de bloqueos y robos por partidos, dado que son estadísticas menos frecuentes a diferencia de los primeros criterios.

Alternativa 2: Árboles y arreglos:

- Esta alternativa consiste en usar los árboles AVL, BBA, RN pero los nodos serían arreglos de cadenas de caracteres donde se almacenarán todos los jugadores con valores repetidos.

Alternativa 3: Árbol binario de búsqueda, árbol rojinegro y árbol AVL:

- Esta alternativa consiste en utilizar los tres tipos de árboles de manera simple sin ninguna estructura extra. Un AVL y un ABB para el criterio de búsqueda puntos, un RN y un ABB para búsqueda por asistencias, un AVL para robos y un RN para bloqueos. Los árboles que son utilizados para un mismo criterio se comparan en tiempos de búsqueda, inserción y eliminación.

**Eficiencia temporal en Búsquedas:**

Se plantean posibles soluciones para el problema en tiempo que se genera al buscar un valor para el cual varios jugadores cumplen.

Alternativa 1. Árbol de Queues.

Implementar en cada árbol Queues de jugadores, cuyo valor buscado es igual, en cada nodo, en el mejor caso cada nodo tendrá una Queue de un sólo jugador.

Alternativa 2. Arreglo de Hash Table.

Implementar un arreglo de 4 posiciones que contenga Hash Tables, cada posición representaría un criterio de búsqueda. Al encontrar en la Hash Table el valor buscado se desplegará un ArrayList de los jugadores que tienen este mismo valor, para obtener de esta lista el jugador buscado.

Alternativa 3. Árbol de HashTable.

Implementar un árbol cuyos nodos tengan el valor buscado y dado el caso de tener múltiples jugadores con un mismo valor, desplegar en una Hash Table dichos jugadores, para obtener en menor tiempo el jugador requerido.

**Paso 4. Transición de las Ideas a los Diseños Preliminares**

(alternativas preseleccionadas)

**Estructuras de datos:**



### Alternativa 2: Árboles y arreglos:

- Esta alternativa haría el almacenamiento de datos más sencillo, sin embargo podría afectar la funcionalidad de las consultas.

### Alternativa 3: Árbol binario de búsqueda, árbol rojinegro y árbol AVL:

- Esta alternativa ofrece que las consultas se realicen en un tiempo de  $\log N$ , sin afectar de manera negativa el almacenamiento de los datos.

### **Eficiencia temporal en Búsquedas:**

#### Alternativa 1. Árbol de Queues.

Esta alternativa ofrecería un fácil acceso a los elementos, aunque su búsqueda en complejidad temporal sea lineal.

#### Alternativa 3. Árbol de HashTable.

Esta alternativa ofrecería una óptima complejidad por el uso de la Hash Table.

### **Paso 5. Evaluación y Selección de la Mejor Solución**

#### Criterios

#### **Criterio A. Velocidad, complejidad temporal.**

- [2] Las estructuras de datos a usar deben representar en composición una solución eficiente, con baja complejidad temporal.

#### **Criterio B. Simplicidad en el manejo de los elementos.**

- [1] Las estructuras de datos combinadas deben mantener una simplicidad en su composición para un óptimo entendimiento del desarrollo de la aplicación.

#### **Criterio C. Acceso a los elementos de las estructuras de datos.**

- [2] Las estructuras deben limitar el acceso a los elementos para optimizar las consultas requeridas.

#### Evaluación

	<b>Criterio A</b>	<b>Criterio B</b>	<b>Criterio C</b>	<b>Total</b>
<b>Árboles y arreglos</b>	1	1	1	3
<b>Árbol binario de búsqueda, árbol</b>	2	1	2	5

<b>rojinegro y árbol AVL</b>				
<b>Árbol de Queues</b>	0	1	2	3
<b>Árbol de HashTable</b>	2	0	2	4

### Selección

Se escogen las alternativas:

- Implementación de un árbol AVL y un ABB para el criterio de puntos.
- Implementación de un árbol RN y un ABB para el criterio de asistencias.
- Implementación de un árbol AVL para el criterio de robos por partido.
- Implementación de un árbol RN para el criterio de bloqueos por partido.
- Implementación de una Hash Table para los jugadores con mismo valor de un criterio.
- Composición de las estructuras previas para las consultas.

La elección de implementar estas estructuras de datos para la solución del problema de los jugadores fue llegar a una mejor complejidad temporal, dada la eficiencia de la composición de estas, además de mantener simple la idea, no dificultar el acceso a los elementos ni afectar la comprensión de la solución de desarrollo propuesta.

## Paso 6. Preparación de Informes y Especificaciones

Especificación del Problema (en términos de entrada/salida)

### Definición de TADS

Documentación: TADs, diseño de pruebas, diagramas de clases y de pruebas, se encuentran en la carpeta Docs del repositorio.

## Paso 7. Implementación del Diseño

Implementación en Java.

<https://github.com/TakumiMay/BasketApp.git>