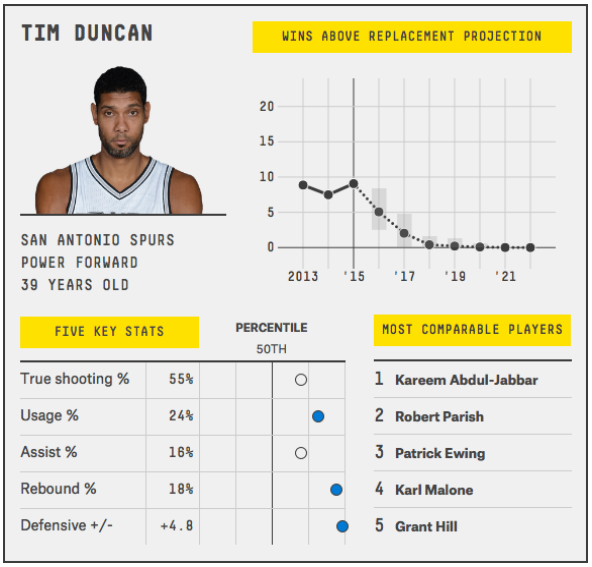
## Objetivos

**Unidad 3: Algoritmos y Estructuras Recursivas**

OE3.1. Calcular la complejidad temporal de algoritmos recursivos.
OE3.3. Resolver ecuaciones de recurrencia que sean resultado del análisis de complejidad temporal de algoritmos recursivos.
OE3.5. Aplicar la técnica de diseño de algoritmos Dividir y Conquistar en la solución de problemas.
OE3.6. Utilizar estructuras recursivas de datos para representar la información del modelo de datos cuando sea conveniente.
OE3.7. Evaluar la utilidad del concepto de orden en un árbol binario de búsqueda para la solución de problemas.
OE3.8. Evaluar la utilidad del concepto de balanceo en un árbol binario de búsqueda para la solución de problemas.
OE3.9. Desarrollar estructuras de datos recursivas ABB.
**Opcional:** OE3.10. Desarrollar estructuras de datos recursivas R&N.
OE3.11. Desarrollar estructuras de datos recursivas AVL.
OE3.14. Desarrollar las pruebas unitarias de cada una de las estructuras de datos recursivas implementadas.

## Enunciado

El baloncesto es uno de los deportes más importantes y populares a nivel orbital, ya sea por la simplicidad de su juego, por su práctica masiva a lo largo y ancho del planeta, por el espectáculo que genera a su alrededor o por la revolución estadística que ha traído consigo. Aunque la esencia de este bello deporte, inventado hace más de un siglo por el profesor canadiense James Naismith, permanece, muchos otros aspectos de este deporte han evolucionado. Con el transcurrir de los años y a medida que el profesionalismo ha ido avanzando, el seguimiento al juego de la pelota naranja se ha servido de diversos factores para aumentar su nivel de detalle y, en consecuencia, el volumen de análisis de datos. Es así como el campo de observación dentro de este juego ha crecido exponencialmente. En la actualidad se llevan estadísticas en múltiples categorías y aspectos del deporte: puntos, rebotes, asistencias, bloqueos, robos, porcentajes de éxito en cada tipo de tiro, etc.




La Federación Internacional de Baloncesto, mejor conocida como FIBA, es el ente regulador del basquetbol a nivel mundial, aquél que define las reglas de este deporte a nivel internacional y el organismo no sólo encargado de organizar y coordinar las más importantes competiciones orbitales sino de reunir a todos los practicantes de este deporte a nivel profesional.

Ante la reciente avalancha de números en donde cualquier cifra proveniente del juego es susceptible de deparar un mensaje mínimamente útil, la FIBA ha decidido aprovechar esta coyuntura y consolidar en una aplicación, los datos de mayor relevancia de cada uno de los profesionales del baloncesto en el planeta, de manera que se puedan efectuar diferentes consultas que permitan realizar análisis sobre 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.

A usted, una compañía desarrolladora de software de mediana escala, pero de mucho futuro, le ha solicitado la implementación de una herramienta para el manejo de información de gran tamaño que permita ingresar datos, ya sea de manera masiva (con archivos .csv por ejemplo) o a través de una interfaz; eliminar o modificar datos; realizar consultas de jugadores utilizando como criterios de búsqueda las categorías estadísticas incluídas (por ejemplo, encontrar aquellos jugadores que han anotado 10 puntos por partido, o aquellos con más de 20 rebotes por partido). Como primera versión del software, la FIBA tan sólo le ha solicitado a usted, como mínimo el incluir los datos por jugador de los siguientes ítems: nombre, edad, equipo y 5 estadísticas (e.g. puntos por partido, rebotes por partido, asistencias por partido, robos por partido, bloqueos por partido). Asimismo, se debe tener en cuenta que esta herramienta puede llegar a almacenar millones de datos y la rapidez para el acceso de estos resulta fundamental para el desempeño de la aplicación.

Como podrá imaginarse, tal cantidad de información deberá guardarse en memoria secundaria ya que por su tamaño es imposible tenerla en memoria principal. Si bien resulta evidente tal necesidad, cabe resaltar la importancia de garantizar un rápido acceso a los datos, es decir, eficiencia en las consultas. Definitivamente la complejidad temporal no puede ser lineal, en lo

que a la búsqueda de jugadores se refiere, ya que sería muy lento, teniendo en cuenta los datos almacenados de cientos de miles de basquetbolistas.

La aplicación que usted desarrolle debe estar en la capacidad de recuperar jugadores de acuerdo a la categoría de búsqueda seleccionada y el valor dado para ella (recuerde que no necesariamente se piden consultas en donde el atributo satisfaga una igualdad). El criterio de búsqueda puede ser cualquiera de los atributos estadísticos, pero para cuatro de ellos la búsqueda debe ser muy rápida. Estas características para las cuales la búsqueda de datos de jugadores es muy eficiente, se denominan índices.

Por esta razón, se utilizarán árboles binarios de búsqueda balanceados para acceder rápidamente a los datos del jugador de manera que la búsqueda tome un tiempo O(log n) para aquellos que tienen índices asociados. Es decir, esta estructura de datos recursiva guardará en ella el valor del atributo y la posición en disco del jugador al que pertenece, de tal manera que la búsqueda sea mucho más eficiente. Para los demás atributos no habrá índice asociado y por tanto la búsqueda será lineal.

Como usted desea justificar la elección de árboles binarios balanceados por sobre árboles binarios no balanceados, usted deberá mostrar el tiempo que se toma en realizar una consulta y adicionalmente permitirá al cliente realizar búsquedas sobre dos criterios estadísticos utilizando ABB como estructura para manejo de índices.

Su programa podrá ejecutar consultas de jugadores de acuerdo a todos los criterios, pero solamente cuatro de ellas (sobre atributos estadísticos) deben resultar eficientes. **Su aplicación deberá contar con una interfaz gráfica**. Igualmente, su programa ha de funcionar con al menos 200.000 datos válidos. Extrapolen de alguna colección de datos (data set) apropiada, por ejemplo https://data.world/jgrosz99/nba-player-data-1978-2016

Usted debe utilizar el método de la ingeniería para resolver este problema y dejar evidencia en su informe de los resultados de cada fase. Recuerde revisar el Resumen del Método de la Ingeniería y el ejemplo del Método de la Ingeniería aplicado a un problema.

### Entregables.

1. Informe PSP0. Cada estudiante debe entregar el informe de su desarrollo.
2. Entrega informe del método de la ingeniería.
3. Especificación de Requerimientos y Diseño. Los requerimientos funcionales debe incluirlos en la fase 1.
4. Diseño del TAD para cada estructura de datos requerida.
5. Diseño del diagrama de clases desacoplado y utilizando generics.
6. Diseño de los casos de prueba, para las estructuras de datos y para el programa.
7. Diseño del diagrama de clases de pruebas unitarias automáticas.
8. Implementación del programa en Java.
9. Implementación de las pruebas unitarias automáticas.
10. Usted debe entregar el enlace del repositorio en GitHub o GitLab con los elementos anteriores. El nombre del repositorio debe estar en inglés, en minúsculas y si tiene varias palabras, éstas van separadas por un guión. Su repositorio debe corresponder con un proyecto de eclipse. Debe tener al menos 10 commits con diferencia de 1 hora entre cada uno de ellos. En el repositorio o proyecto de eclipse debe haber un directorio llamado docs/ en el cual deberán ir cada uno de los documentos del diseño.

**Bonus:** Cuando uno de los árboles binarios de búsqueda autobalanceado se obtenga implementando un Árbol Rojo y Negro.

La rúbrica con la que se evaluará esta tarea se encuentra en el siguiente enlace Rúbrica TI 2.