

Objetivos**Unidad 3: Estructuras Discretas no recursivas y análisis de algoritmos.**

OE3.1 Explicar el significado de “mejor”, “promedio” y “peor” caso en lo que a comportamiento de algoritmos se refiere.

OE3.2 En el contexto de algoritmos específicos, identificar las características de los datos y/o otras condiciones o supuestos que conduzcan a distintos comportamientos.

OE3.3 Determinar, de manera informal, la complejidad temporal y espacial de algoritmos simples.

OE3.4 Establecer la definición formal de Big O.

OE3.5 Enumerar y contrastar las clases de complejidad de un algoritmo.

OE3.6 Calcular la complejidad temporal de algoritmos iterativos.

OE3.7 Calcular la complejidad espacial de algoritmos iterativos.

OE3.8. Diseñar e implementar un API para un proyecto de pequeña escala, utilizando un lenguaje de programación orientado a objetos, librerías e incluyendo parametrización y generics.

OE3.9. Diseñar e implementar una solución a un problema utilizando un lenguaje de programación teniendo en cuenta un criterio de eficiencia computacional y estándares de codificación seguros.

OE3.10 Definir, implementar y utilizar tablas hash, incluyendo técnicas de resolución de colisiones.

OE3.12 Definir, implementar y utilizar estructuras discretas FIFO y LIFO.

OE3.13 Desarrollar las pruebas unitarias para cada una de las estructuras discretas implementadas.

Unidad 4: Algoritmos y estructuras discretas recursivas

OE4.5 Definir, implementar y utilizar montículos binarios.

OE4.6 Describir la propiedad de montículos y el uso de estos como implementación de una cola de prioridad.

Tarea Integradora I - Sistema de Gestión de Equipos en la Champions League

La UEFA Champions League es el torneo de clubes más prestigioso de Europa. Para su correcta gestión, es necesario diseñar un sistema que permita administrar los equipos participantes, los partidos jugados y la gestión de los rankings de los equipos en función de sus resultados.

Para ello, se requiere implementar un sistema que haga uso de diferentes estructuras de datos, como tablas hash, pilas, colas y colas de prioridad, con el objetivo de optimizar el procesamiento de la información y la toma de decisiones.

Estructuras de Datos**1. Tabla Hash:**

- Se utilizará para almacenar información sobre los equipos y los partidos de manera eficiente, permitiendo una búsqueda rápida.
- **Equipos:** nombre, país, cantidad de títulos, coeficiente UEFA.

- **Partidos:** equipo local, equipo visitante, goles local, goles visitante, fecha.

2. Pila

Se usará para permitir deshacer la última acción realizada en el sistema (agregar equipo, registrar partido, actualizar clasificación, etc.).

Ejemplo: Si un usuario agrega erróneamente un equipo, se puede deshacer la acción eliminando el último elemento agregado a la pila.

3. Cola

Se empleará para gestionar el orden en el que se juegan los partidos en una jornada. Los partidos se jugarán en el orden en que fueron registrados, siguiendo un esquema FIFO (First In, First Out).

Ejemplo: Si se registran los partidos PSG vs. Barcelona y luego Manchester City vs. Bayern, el sistema los procesará en ese mismo orden.

4. Cola de prioridad

Se utilizará para gestionar los equipos en función de su rendimiento en la competición. La prioridad estará determinada por el coeficiente UEFA y los puntos obtenidos.

Grupos de funcionalidades principales

1. Registro de Equipos:

- a. Permitir el registro de equipos con sus respectivos atributos en el sistema.

2. Registro de Partidos:

- a. Permitir registrar los resultados de los partidos jugados entre los equipos.

3. Deshacer Acción:

- a. Utilizar la pila para revertir la última acción ejecutada en el sistema.

4. Gestión de Partidos:

- a. Organizar los partidos en una cola para establecer el orden de juego en cada jornada.

5. Clasificación y Rankings:

- a. Gestionar la clasificación de los equipos en la Champions League utilizando una cola de prioridad basada en los puntos obtenidos y el coeficiente UEFA.

Entregables

1. Desarrollo completo del Método de la Ingeniería. (Ejemplo)

- a. La fase 1 debe incluir la especificación de requerimientos.

2. Análisis.

- a. Análisis de complejidad temporal de al menos dos de los algoritmos implementados (deben ser algoritmos iterativos de complejidad media).
- b. Análisis de complejidad espacial de al menos dos de los algoritmos implementados (deben ser algoritmos que involucren al menos 5 variables de distintos tipos).

3. Diseño.

- a. Diseño del TAD de las estructuras de datos utilizadas.

- b. Diseño completo del diagrama de clases usando genéricos, incluyendo, las estructuras de datos, el paquete del modelo, de la interfaz de usuario y pruebas.
 - c. Diseño de casos de prueba, incluyendo los escenarios, para las estructuras y el sistema.
- 4. Implementación en Java.
 - a. Implementación completa de las estructuras de datos y sus pruebas.
 - b. Implementación completa y correcta del modelo, la UI y las pruebas (contexto del problema).

Este proyecto puede realizarse en grupos de **mínimo 2** y **máximo 3** personas.

Creen sus equipos utilizando el siguiente enlace: [\[link\]](#)

Su repositorio deberá tener al menos 10 commits (por estudiante) con una diferencia de al menos 2 horas entre cada uno de ellos. Dichos commits deberán ser significativos en términos de valor (e.g. no puede ser borrar una variable o cambiar un nombre de un método). En el repositorio o proyecto, debe haber un directorio llamado docs/ donde se deben colocar cada uno de los documentos de diseño. En dicho directorio también deberá estar el reporte con el método de la ingeniería. Incluya cualquier aclaración necesaria para manipular o entender su proyecto en el archivo readme.md como documentación adicional (por ejemplo, el IDE utilizado y los nombres de los integrantes). El aporte de cada integrante se evaluará teniendo en cuenta su aporte y la única manera de corroborarlo será a través de los commits.

Nota: La rúbrica con la que se evaluará esta tarea se encuentra en la pestaña TI1 de Notas y Seguimientos de su grupo. Se recomienda revisar la rúbrica con la que será evaluada su entrega.

Fecha Máxima de Entrega: 26 de Abril del 2025