



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

MÓDULO 3: ORDENAMIENTO

Guía de trabajo del reto de muestras numéricas

Con este reto se espera que usted fortalezca sus conocimientos a través de:

- La utilización de los algoritmos presentados en el módulo para resolver problemas de ordenamiento en un contexto específico.
- El análisis de las implementaciones de los algoritmos de ordenamiento adaptados a un problema en un contexto específico.

Esta guía le ayudará a interactuar con el caso y sus diferentes elementos siguiendo un paso a paso. Podrá explorar el código fuente en el lenguaje de programación orientado a objetos Java para entender las diferentes propuestas de implementación de los elementos teóricos vistos durante el módulo.

Finalmente, serán planteadas algunas preguntas abiertas para que identifique y reflexione sobre distintos elementos algorítmicos presentes en los ordenamientos.

ETAPA 1: PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para realizar esta hoja de trabajo, siga las instrucciones dadas a continuación:

1. Descargue el archivo comprimido m3_caso_muestra que contiene el proyecto.
2. Descomprima el archivo e impórtelo en el entorno Eclipse.

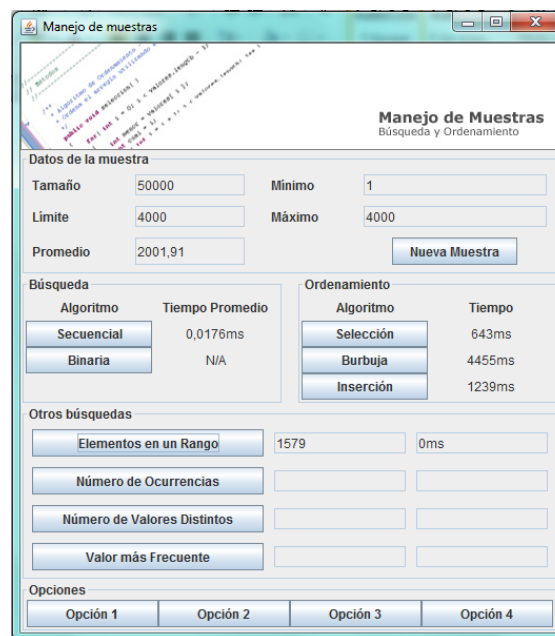
ETAPA 2: EXPLORACIÓN

1. Revise los siguientes documentos que encontrará en la carpeta docs > specs:
 - a. **Descripción.pdf**: Este documento contiene la descripción del proyecto.
 - b. **RequerimientosFuncionales.pdf**: Este documento contiene el detalle de cada requerimiento funcional.



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

- c. **ModeloConceptual.jpg**: Este documento presenta el modelo de clases propuesto para el caso.
2. Ejecute el proyecto m3_caso_muestra como una aplicación en Java. El programa le permite crear conjuntos de números dado un rango y un número de elementos, así como realizar diferentes operaciones de búsqueda y ordenamiento sobre ellos. La interfaz del programa debe ser la que se muestra a continuación:



3. Cree varias muestras aleatorias de diferentes tamaños y rangos. Analice los tiempos de ordenamiento de los algoritmos por Selección, Burbuja e Inserción.
4. Revise la implementación del código de la interfaz y del mundo para entender la estructura del proyecto. Por el momento ignore todas las clases pertenecientes a la interfaz.
- En la clase **Muestra** (clase principal del mundo), revise la implementación de los métodos `ordenarSeleccion`, `ordenarBurbuja` y `ordenarInsercion` que operan sobre un arreglo de números. Intente modificar los algoritmos para determinar cómo las variaciones que haga pueden afectar su comportamiento y correcto funcionamiento.



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

ETAPA 3: REFLEXIÓN

Con base en sus observaciones, responda las siguientes preguntas:

- ¿Cuál algoritmo de ordenamiento es más eficiente y cuál es más ineficiente en tiempo?
¿Cuáles serían las razones para estas variaciones de eficiencia?
- ¿Coinciden sus observaciones con los cálculos de complejidad en tiempo para los diferentes algoritmos explicados anteriormente en el módulo? Si sus observaciones no coinciden ¿Cuáles cree que son las razones para éstas discrepancias? Intente ejecutar los algoritmos varias veces y con diferentes muestras para confirmar sus hipótesis.
- ¿La implementación de los algoritmos de ordenamiento del caso de estudio Ordenador de Muestras Numéricas es más o menos eficiente que la presentada en las lecciones del módulo? ¿En qué se diferencian?