

PARTE 3: Ejercicio de programación JAVA

Este ejercicio comprende **3** partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega de todo el Proyecto **Maven** en la tarea correspondiente de la webasignatura.
3. Desarrollo de los casos de prueba para las funcionalidades requeridas

ESCENARIO

El sistema de inventario representa un almacén que contiene una variedad de productos. Cada producto está asociado con un código único y se organiza jerárquicamente en categorías y subcategorías. Los nodos internos del árbol representan estas categorías, mientras que los nodos hoja representan los productos individuales sin subcategorías.

Dado un árbol de búsqueda binaria (ABB) que representa un sistema de gestión de inventario, se solicita implementar un algoritmo para recorrer el árbol y separar sus nodos en dos listas: nodos hoja (productos sin más subcategorías) y nodos internos (categorías que contienen subcategorías/productos). Se debe mantener el orden relativo de los nodos dentro de cada lista.

Las firmas de todos los métodos son abiertas, pero se debe incluir una demostración de cómo invocar al método solicitado.

PARTE 1: Funcionalidad a desarrollar (vale 45%):

Descargar de la webasignatura el archivo “Parcial1-2024-2.zip” que contiene el Proyecto **Maven** a ser completado.

Se desea:

1. Implementa en la clase apropiada, un método que, retorno en 2 listas, los productos sin más categorías (nodos hoja) y los productos que contienen subcategorías (nodos internos).

Tipo Producto

Identificador: numérico

Nombre: alfanumérico

PARTE 2: PROGRAMA (vale 30%)

La clase principal se denomina “**Parcial1**”, y tiene su correspondiente método “**main**”. En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados.

1. Leer y cargar los datos de los productos
2. Invocar el método programado
3. Escribe un archivo de texto “**salida.txt**”, con los datos de los productos sin categorías, uno por cada línea, en orden ascendente por nombre y apellido, separados por comas.

PARTE 3: TEST CASES (vale 25%).

Implementa el o los **Casos de Prueba** necesarios para verificar el correcto funcionamiento del método implementado (**sólo a nivel del TArbolDeProductos**).

NOTAS IMPORTANTES:

- Se proveen las interfaces y clases necesarias. Deben implementarse los métodos necesarios (respetando las firmas indicadas). NO SE DEBEN ALTERAR LAS INTERFACES provistas, ni agregar otros métodos que los requeridos en las interfases.
- **NO SE DEBEN CREAR NUEVAS CLASES Y NO SE DEBE INCLUIR NINGUN METODO NO SOLICITADO O INNECESARIO.**
- Debe prevenirse, usando solo las interfaces y clases provistas, que el árbol resultante del punto 2 tenga una estructura extremadamente mala (la altura del árbol resultante no debe ser comparable al tamaño de este).

ENTREGA: Debes entregar TODO el proyecto Maven y los archivos de salida solicitados, en un archivo comprimido “**Parcial1.zip**” en la tarea publicada en la webasignatura, hasta la hora indicada.