**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 test para las funcionalidades requeridas

**ESCENARIO**

La información de todos los alumnos de una facultad se han almacenado en una lista. Cada alumno está representado mediante un identificador único (numérico, sin código de verificación), su nombre, apellido y carrera en la que está matriculado (estos tres son alfanuméricos). Además, se tiene una lista con la información de los cursos de la facultad.

Una de las tareas típicas que se suele realizar es consultar los alumnos de una cierta asignatura. Para evitar realizar una búsqueda extensa sobre todos los alumnos de la facultad, se decide generar otras estructuras que permitan hacerlo en forma más eficiente.

Se desea entonces, dado la lista de todos los alumnos de la facultad (con los datos indicados más arriba, habiendo sido insertados de acuerdo a su número de CI), y dado el nombre de una cierta carrera, crear un nuevo árbol que indexe la información de los alumnos correspondientes, por Nombre y Apellido (se desea poder imprimir la lista de alumnos de la carrera ordenada por Nombre y Apellido en la forma más eficiente).

Además, se desea contar con un método para armar 2 subgrupos del curso, 1 con los alumnos que se encuentran en niveles pares del árbol y otro con los alumnos en los niveles impares.

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

Descargar de la webasignatura el archivo “Parcial1-2023-2.zip” que contiene el Proyecto **Maven** a ser completado.

**Se desea:**

1. Implementa en la clase apropiada, unmétodo que, recibiendo por parámetro un nombre de materia, devuelva una instancia de un árbol binario de búsqueda con los alumnos que están inscriptos en ese curso y ningún otro.

**Tipo Alumno**

Identificador : numérico

Nombre: alfanumérico

Apellido: alfanumérico

Cursos: numérico[]

**Tipo Curso**

Identificador: numérico

Nombre: alfanumérico

**Tipo Facultad**

Alumnos: TLista<Alumno>

Cursos: TLista<Curso>

**de Tipo Facultad armarIndiceCurso(de tipo string CursoNombre): devuelve TArbolBB<Alumno>**

**de Tipo ArbolBB<Alumno> armarSubgrupos(ArbolBB<Alumno> grupoPar, ArbolBB<Alumno> grupoImpar): devuelve void**

## 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 alumnos y cursos a partir de los archivos alumnos.txt y cursos.txt (el parsing de las líneas ya está dado)
2. Invocar el método **armarIndiceCurso** para el curso indicado en el pizarrón.
3. Invocar el método **armarSubgrupos** para los resultados del paso anterior.
4. Imprime por pantalla el valor total del stock del rubro indicado.
5. Imprime por pantalla los datos del paso 2
6. Escribe un archivo de texto **“salida.txt”**, con los datos de los alumnos del curso, 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 ArbolBB<Alumno>).

## 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, tanto como sea posible utilizando solamente 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 del mismo).

## ENTREGA: Debes entregar TODO el proyecto Netbeans y los archivos de salida solicitados, en un archivo comprimido “Parcial1.zip” en la tarea “PARCIAL1-PARTE3” publicada en la webasignatura, hasta la hora indicada.