PARTE 3: Ejercicio de programación JAVA

Este ejercicio comprende **3** partes:

1. Desarrollo de la funcionalidad especificada más abajo (50 %)
2. Desarrollo de los casos de prueba ("test cases") para verificar la corrección de la funcionalidad implementada. (30 %)
3. Desarrollo del programa ejecutable con datos de ejemplo provistos y ejecución correcta del mismo (20%).

ESCENARIO

Supongamos que en una estructura de Árbol Binario de Búsqueda se han almacenado los datos de todos los alumnos de una Universidad. Cada alumno está representado mediante un identificador único (su CI, numérico, sin código de verificación), su apellido y carrera en la que está matriculado.
Una de las tareas típicas que se suele realizar es consultar por los alumnos de una cierta carrera. Para evitar realizar una búsqueda extensa sobre todos los alumnos de la Universidad, se decide generar otras estructuras que permitan hacerlo en forma más eficiente.
Se desea entonces, dado un árbol binario de búsqueda que contenga todos los alumnos de la universidad (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 Apellido (se desea poder imprimir la lista de alumnos de la carrera ordenada por Apellido en la forma más eficiente). Asumiremos que no hay apellidos repetidos.
SE PROVEEN INTERFASES Y CLASES CON IMPLEMENTACIONES ESPECIFICAS DEL ÁRBOL Y ELEMENTO DE ÁRBOL, A EFECTOS DE SIMPLIFICAR LA COMPRENSIÓN Y DESARROLLO DEL EJERCICIO:

Tipo Alumno
CI: numérico
Apellido: alfanumérico
Carrera: alfanumérico

TABBU -Tipo ArbolUniversidad- **// TABBU es un Arbol Binario de Búsqueda específico que contiene nodos //de tipo NodoAlumnoAB**
De Tipo NodoAlumnoAB Raíz

Tipo NodoAlumnoAB
De Tipo comparable Etiqueta
De Tipo Alumno Dato
De Tipo NodoAlumnoAB HijoIzq, HijoDer

SE PIDE:

Desarrollar un método del Arbol Binario de Busqueda de Universidad - **TArbolBBU** - que, dado un parámetro indicando un **nombre de carrera**, devuelva otro **TArbolBBU** que contenga las **referencias** a todos los alumnos matriculados en dicha carrera, utilizando su Apellido como clave.

PARTE 1: Funcionalidad a desarrollar:

Descargar y abrir el proyecto Netbeans desde la webasignatura (Archivo **PARCIAL2-PARTE3**).

Implementar el método del **TArbolBBU** y el correspondiente de **TNodoAlumnoAB** que, recibiendo por parámetro el nombre de una carrera, devuelva otro **TArbolBBU** cargado con referencias a los alumnos correspondiente, utilizando como clave el **Apellido**.

PARTE 2: TEST CASES.

Implementa los **Casos de Prueba** necesarios para verificar el correcto funcionamiento del método (a nivel del árbol).

PARTE 3: PROGRAMA.

En el método "**main**" de la clase "**Main**", implementar lo necesario para:

1- Crear una instancia del TABBU y cargarla con los datos existentes en el archivo "**datos.csv**" (en cada línea del mismo hay un código de alumno, apellido de alumno y carrera, separados por comas)

2- Ejecutar el método "**armarIndiceCarrera**" del árbol de Universidad, recibiendo como parámetro un nombre de carrera a especificar (se indicará en el pizarrón).

3- Escribir un archivo "**salida.txt**" que contenga el listado de los alumnos de la carrera especificada, en orden ascendente por apellido, emitido a partir del árbol resultante de la ejecución del paso 2.

ENTREGA: Debes entregar TODO el proyecto Netbeans, más el archivo de salida "salida.txt", en un archivo comprimido "Parcial2.zip" en la tarea "PARCIAL2-PARTE3" publicada en la webasignatura, hasta la hora indicada.

NOTA IMPORTANTE: Las operaciones **a realizar estarán indicadas en las interfaces provistas**. Se deberá respetar las firmas establecidas en las mismas.

El trabajo será evaluado y calificado mediante casos de test estándar desarrollados por la Cátedra, que invocarán los métodos requeridos con las firmas establecidas**.**