

PARTE 3: Ejercicio de programación JAVA

Duración 60 minutos

Este ejercicio comprende **3** partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Implementación y ejecución del programa principal
3. Desarrollo de los casos de prueba ("test case", al menos uno) para verificar la corrección de la funcionalidad implementada.

ESCENARIO

En una estructura de lista estándar 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, carrera y un campo adicional "matriculado" que indica si está actualmente registrado en al menos una asignatura de esa carrera. Es habitual emitir un listado de todos los alumnos.

Por otra parte, dado que es frecuente la consulta de los datos de alumnos que **de una cierta carrera que efectivamente están matriculados en alguna asignatura**, por **apellido**, se ha decidido crear también un árbol binario de búsqueda que permita un acceso eficiente para satisfacer esta consulta.

PARTE 1: Funcionalidades a desarrollar (vale 40%):

Descargar y abrir el proyecto Netbeans desde la webasignatura (Archivo "EXAMEN JULIO 2017 CODIGO BASE"). La clase principal se denomina "**Examen**", y tiene su correspondiente método "**main**".

Desarrollar e implementar:

- En **Lista<T>**: **insertar(TNodo<T> unNodo)**
- En **TArbolBB<T>**:
 - **insertar (TElementoAB<T> unElemento)**
 - **arrayList<T> listar()**
- **TElementoAB<T>** los correspondientes métodos recursivos:
 - **insertar (TElementoAB<T> unElemento)**
 - **listar (arrayList<T> unArrayList)**

clase TAlumno

Cedula **entero**

Apellido **string**

Carrera **string**

Matriculado **boolean**

PARTE 2: PROGRAMA (vale 30%).

En el método "**main**" de la clase "**Examen**", implementar lo necesario para aplicar los TDA y métodos desarrollados, ejemplificando de la siguiente forma (asumiendo que los cursos pueden ser representados como instancias de **TConjunto**):

1. Instanciar una lista (TDA Lista provisto) y cargar en la misma todos los alumnos (en cada nodo, el campo "datos" es genérico con T = TAlumno) a partir del archivo "**alumnos.csv**", que contiene un alumno por línea, con la siguiente estructura (campos separados por comas):

CEDULA (numérico), **APELLIDO**(alfanumérico), **CARRERA**(alfanumérico), **MATRICULADO**(alfanumérico, SI/NO)

2. Instanciar un árbol binario de búsqueda (con T = TAlumno) y cargarlo con los alumnos **de la carrera especificada (se indicará en el pizarrón)** que se encuentran **matriculados**, usando como clave de búsqueda / inserción el campo **"APELLIDO"** (la etiqueta del TElementoAB es el Apellido, los datos<T> son el TAlumno correspondiente).
3. Invocar al método **"listar"**, que devuelve en un **arrayList** todos los alumnos de la carrera especificada, que están matriculados (**contenidos en el árbol**), ordenados por su **apellido**.
4. Imprimir por consola los datos de los alumnos contenidos en el arrayList del punto 4, un alumno por línea, con el siguiente formato:

APELLIDO, CARRERA, MATRICULADO

PARTE 3: TEST CASES (vale 30%)

Implementa los **Casos de Prueba** necesarios para verificar el correcto funcionamiento de los métodos desarrollados.

ENTREGA: Debes entregar **TODO** el proyecto Netbeans, en un archivo comprimido **"EXAMEN2017.zip"** en la tarea **"EXAMEN-JULIO-2017-PARTE3"** publicada en la webasignatura, hasta la hora indicada.