



Práctica 11: Preparación para prueba

Para resolver este examen debe completar la hoja adjunta que se le entrega, con porciones de código. No puede cambiar el código que se le provee ni las firmas de las funciones, sólo debe completar las partes que faltan.

1. Trabajaremos con la siguiente estructura:

```
type estudiante = {Nombre: String, Notas: real list};
```

- a. Defina una función **promedios** que dada una lista de estudiantes (**estudiante list**) devuelva otra lista de estructuras, donde cada estructura contenga el nombre del alumno y su promedio.
- b. Defina una función **aprobados** que dada una lista de estudiantes (**estudiante list**) devuelva la cantidad de alumnos aprobados (para aprobar la materia se necesita un promedio mayor o igual a 6).

2. Considere la siguiente definición de árbol binario ya vista en clase y un nuevo tipo de dato **libro**:

```
datatype 'etiqueta arbolbin =  
  Vacio |  
  Nodo of 'etiqueta * 'etiqueta arbolbin * 'etiqueta arbolbin;
```

```
type libro = {Titulo: string, Autor: string, Disponible: bool};
```

Suponga que una biblioteca de una escuela guarda el registro de todos sus libros como un árbol binario de búsqueda de tipo **libro arbolbin** (existe un solo ejemplar de cada libro). Cada nodo del árbol es una estructura que contiene los datos de un libro (el título del libro escrito todo en minúscula), su autor y un booleano que indica si el libro se encuentra o no disponible. El árbol binario de búsqueda está ordenado de acuerdo al orden alfabético del campo **Titulo**. Implemente las siguientes funcionalidades que permitan utilizar la agenda:

- a. Defina una función **ingresar_libro** que tome el registro de todos los libros de esta biblioteca (implementada como un árbol binario de búsqueda) y los datos de un libro y si dicho libro no se encuentra ya en el registro se agrega al mismo (recuerde mantener la propiedad de árbol binario de búsqueda), indicando disponibilidad. Si el libro ya estaba en el registro no debe modificarse el mismo.
Ayuda: Para esto defina antes una función **crear_libro** que dados los datos de un libro, cree la estructura libro con dichos datos.
- b. Defina una función **prestar_libro** que tome el registro de libros (implementado como un árbol binario de búsqueda) y un título y si dicho libro se encuentra disponible, modifique su estado en el registro (indicando ahora que no está disponible). Si el libro ya no se encontraba disponible o el libro no se encontraba en el registro no debe modificar nada en el mismo.
- c. Defina una función **buscar_libro** que tome el registro de libros (implementado como un árbol binario de búsqueda) y el título de un libro y devuelva **true** si el mismo se encuentra disponible, **false** en cualquier otro caso.
- d. Defina una función **buscar_autor** que tome el registro de libros (implementado como un árbol binario de búsqueda) y el título de un libro y devuelva su autor en el caso en que el libro se encuentra registrado. Si el libro no se encuentra en el registro debe devolver **"Libro no hallado"**.
- e. Defina una función **libros_prestados** que tome el registro de libros (implementado como un árbol binario de búsqueda) y devuelva una lista con todos los títulos de los libros que no se encuentran disponibles.
- f. Defina una función **cantidad_disponibles** que tome el registro de libros (implementado como un árbol binario de búsqueda) y devuelva la cantidad de libros que se encuentran disponibles.