

Parcial del Proyecto II - Tema E

Ejercicio 1: En los primeros tres años de la carrera de *Licenciatura en Ciencias de la Computación* hay ciertas materias de gran relevancia en cuanto a la algoritmia y la programación. Vamos a representar algunas de estas materias con el tipo `Materia`. Deben definir este tipo el cual consta de los constructores `IntroAlgo`, `Algo1`, `Algo2`, `Sistemas`, `Discreta2` e `IngSoft1`. A continuación deben definir el tipo `InfoMateria` como un sinónimo de `(String, Int)`. Luego definir la función:

```
datos_de_materia :: Materia -> InfoMateria
```

que nos dice el nombre de la materia y el año académico al que corresponde. A saber:

- “Introducción a los Algoritmos” y “Algoritmos y Estructuras de Datos I” son materias de 1° año.
- “Algoritmos y Estructuras de Datos II” y “Sistemas Operativos” son materias de 2°
- “Matemática Discreta II” e “Ingeniería del Software I” son materias de 3° año.

Definir la función utilizando *pattern matching*. **No usar definición por casos.**

Ejercicio 2: Ahora vamos a definir un tipo que nos ayudará a llevar la cuenta de qué materias hemos cursado y en qué año lo hicimos. Primeramente definir el tipo `Año` que es un sinónimo del tipo `Int`. Luego definir el tipo recursivo `Cursadas` que consta de dos constructores:

- `AgregarCursada`: Que registra qué materia cursamos (un valor del tipo `Materia`), el año en que la cursamos (un valor del tipo `Año`) y por último un valor del tipo `Cursadas` al cual se le está añadiendo la nueva cursada.
- `Ninguna`: No tiene parámetros y es el equivalente a la lista vacía.

Finalmente programar la función:

```
cantidad_cursadas :: Cursadas -> Año -> Int
```

que dada una instancia `cs` del tipo `Cursadas` y un año `a` del tipo `Año`, cuenta cuantas materias se cursaron el año `a`.

Ejercicio 3: Basados en los tipos `Persona`, `Ro1`, etc definidos en el Proyecto 2 programar la función:

```
cuantos_est_mayores :: [Persona] -> Int -> Int
```

que dada una lista de personas `xs` (del tipo `[Persona]`) y un año `a` (del tipo `Int`) la función devuelve la cantidad de estudiantes de la lista `xs` tienen edad mayor o igual a 18. El cálculo de la edad se hace simplemente restándole el año `a` al año de nacimiento registrado para el estudiante.

Ejercicio 4*: Basados en el tipo `ListaAsoc` del ejercicio 5 del Proyecto 2, programar la función:

```
la_filter :: ListaAsoc a b -> (a -> Bool) -> ListaAsoc a b
```

que dada una lista de asociaciones **la** (del tipo `ListaAsoc a b`) y una predicado **p** (de tipo `a -> Bool`), devuelve una nueva lista de asociaciones con los elementos de **la** cuyas claves satisfacen el predicado **p**.