

# Tema D

## Ejercicio 1: Tienda de Libros

Una tienda de libros desea representar su inventario utilizando tipos en Haskell. Los tipos de libros que tenemos en cuenta son: **Novela**, **Ensayo**, **Poesía** y **Científico**. La idea es poder detallar para cada tipo de libro las características más importantes. En tal sentido, identificamos las siguientes características de cada uno de los libros a tener en cuenta:

### Novela

- **Nombre**, es un sinónimo de *String* y representa el nombre de la novela.
- **Género**, que es un tipo enumerado con las siguientes opciones: *Drama*, *Romance*, *Misterio*, *Aventura*.
- **Autor**, que es un sinónimo de *String* indicando el autor de la novela.
- **Número de páginas**, que es un sinónimo de *Int* indicando la cantidad de páginas del libro.
- **Precio**, que es un sinónimo de *Int* indicando el precio del libro.

### Ensayo

- **Nombre**, es un sinónimo de *String* y representa el nombre del ensayo.
- **Tema**, que es un tipo enumerado con las siguientes opciones: *Política*, *Filosofía*, *Ciencia*, *Historia*.
- **Autor**, que es un sinónimo de *String* indicando el autor del ensayo.
- **Número de páginas**, que es un sinónimo de *Int* indicando la cantidad de páginas del libro.

### Poesía

- **Nombre**, es un sinónimo de *String* y representa el nombre de la poesía.
- **Estilo**, que es un tipo enumerado con las siguientes opciones: *Clásica*, *Contemporánea*, *Experimental*.
- **Autor**, que es un sinónimo de *String* indicando el autor de la poesía.
- **Número de páginas**, que es un sinónimo de *Int* indicando la cantidad de páginas del libro.
- **Precio**, que es un sinónimo de *Int* indicando el precio del libro.

### Científico

- **Nombre**, es un sinónimo de *String* y representa el nombre del artículo científico.
- **Área de estudio**, que es un tipo enumerado con las siguientes opciones: *Astronomía*, *Física*, *Matemáticas*, *Computación*.
- **Autor**, que es un sinónimo de *String* indicando el autor del libro científico.
- **Número de páginas**, que es un sinónimo de *Int* indicando la cantidad de páginas del libro.

Para ello:

- a) Define el tipo **Libro** que consta de los constructores **Novela**, **Ensayo**, **Poesía** y **Científico** con parámetros descritos anteriormente. Se deben definir también los tipos enumerados y los sinónimos de tipos. Los tipos definidos no deben tener deriving Eq, ni Ord. Agrega deriving Show a todos los tipos.

b) Define la función **cuantasNovelasPorGenero** de la siguiente manera:

```
cuantasNovelasPorGenero :: [Libro] -> Género -> Int
```

Esta función, dada una lista de libros y un valor de género, devuelve un entero indicando la cantidad de libros que hay en la lista del género especificado.

**NOTA:** Dejar como comentario un ejemplo donde se haya probado la función *cuantasNovelasPorGenero* con una lista de al menos 3 Libros.

c) Define la igualdad para el tipo **Libro**: de tal manera que:

- 2 **Novelas** son iguales sólo si tienen el mismo *Nombre*, *Género* y *Autor*,
- 2 **Ensayos** son iguales sólo si tienen el mismo *Nombre*, *Tema* y *Autor*,
- 2 **Poesías** son iguales si tienen el mismo *Nombre*, *Estilo* y *Autor*,
- 2 Libros **Científicos** son iguales si tienen el mismo *Nombre* y *AreaEstudio*.

Como es de suponer, las Novelas, Ensayos, Poesías y Libros Científicos son distintos entre sí.

d) Define la función **hayDosDistintosConsecutivos** de la siguiente manera:

```
hayDosDistintosConsecutivos :: [Libro] -> Bool
```

Dada una lista de libros, esta función debe devolver True en caso de que existan dos libros distintos de manera consecutiva en la lista, y False en caso contrario.

**NOTA: Recordar que se definió la igualdad en el inciso anterior.** Dejar como comentario en el código dos ejemplos en los que se probó la función.

## Ejercicio 2: Seguimiento de alumnos

Queremos desarrollar un programa para que los profesores puedan llevar un registro del progreso de sus estudiantes a lo largo del año escolar. Definiremos un tipo recursivo llamado *EstadoEstudiantes*, que permitirá almacenar el progreso de los estudiantes en la escuela.

El tipo **EstadoEstudiantes** tendrá dos constructores:

- *EvolucionDeEstudiante*, que tiene 4 parámetros:
  - *String*: el nombre completo del estudiante.
  - *Int*: la edad del estudiante.
  - *[Int]*: una lista de calificaciones (cada calificación debe estar entre 0 y 10).
  - *EstadoEstudiantes*: la evolución de los demás estudiantes.
- *NoHayEstudiantes*, que es un constructor sin parámetros, similar a la lista vacía.

Las condiciones para que un estudiante pueda avanzar al siguiente nivel escolar se describen a continuación, basadas en sus calificaciones:

- Para avanzar de año, un estudiante debe tener un promedio general de calificaciones mayor o igual a 7.
  - Si el promedio general es menor a 7 pero mayor o igual a 6, el estudiante puede avanzar de año si tiene una calificación de al menos 8 en la materia más importante (la materia con la calificación más baja en la lista de calificaciones).
  - Si el promedio general es menor a 6, el estudiante no puede avanzar de año.
- a) Programar la función **avanzarDeAño**, que toma como primer parámetro **EstadoEstudiantes** y como segundo parámetro el nombre completo de un estudiante, y devuelve un valor de tipo Bool, indicando si el estudiante puede avanzar de año o no.

```
avanzarDeAño :: EstadoEstudiantes -> String -> Bool
```

**NOTA:** Se puede utilizar la función del preludio **fromIntegral**, que convierte un Integer en un Float. Deja como comentario un ejemplo donde hayas probado **avanzarDeAño** con un parámetro de tipo **EstadoEstudiantes** que tenga al menos 3 estudiantes.

- b) Programar la función **promedioGeneral**, que toma como primer parámetro **EstadoEstudiantes** y como segundo parámetro el nombre completo de un estudiante, y devuelve el promedio general de calificaciones del estudiante si está en la lista, o Nothing en caso contrario.

```
promedioGeneral :: EstadoEstudiantes -> String -> Maybe Float
```

**NOTA:** Se puede usar la función mencionada en el inciso anterior **fromIntegral**. Deja como comentario un ejemplo donde hayas probado la función.