

La Unidad de Tecnologías de la Información (UTI) de nuestra Facultad nos ha encargado que desarrollemos un nuevo sistema para el registro de alumnos. En este sistema se guarda la información de cada alumno, que está representada como una tupla de dos elementos: el primero es el nombre completo del alumno y el segundo una lista con las notas de los finales que rindió.

Para implementar este sistema nos enviaron las siguientes especificaciones y nos pidieron que hagamos el desarrollo enteramente en Haskell, utilizando los tipos requeridos y solamente las funciones que se ven en la materia Introducción a la Programación / Algoritmos y Estructuras de Datos I (FCEyN-UBA).

Ejercicio 1 (2 puntos)

```
problema aproboMasDeNMaterias (registro: seq seq Char x seq Z , alumno: seq Char , n: Z) : Bool {
  requiere: {No hay nombres de alumnos repetidos en registro}
  requiere: {Las notas de registro son todas iguales o mayores a cero y menores o iguales a 10}
  requiere: {n > 0}
  requiere: {El alumno se encuentra en el registro }
  asegura: {res = true <=> el alumno tiene más de n notas de finales mayores o iguales a 4 en el registro}
}
```

Ejercicio 2 (2 puntos)

```
problema buenosAlumnos (registro: seq seq Char x seq Z ) : seq seq Char {
  requiere: {No hay nombres de alumnos repetidos en registro}
  requiere: {Las notas de registro son todas iguales o mayores a cero y menores o iguales a 10}
  asegura: {res es la lista de los nombres de los alumnos que están en registro cuyo promedio de notas
es mayor o igual a 8 y no tiene aplazos (notas menores que 4)}
}
```

Para resolver el promedio pueden utilizar la función del Preludio de Haskell fromIntegral que dado un valor de tipo Int devuelve su equivalente de tipo Float.

Ejercicio 3 (2 puntos)

```
problema mejorPromedio (registro: seq seq Char x seq Z ) : seq Char {
  requiere: {No hay nombres de alumnos repetidos en registro}
  requiere: {Las notas de registro son todas iguales o mayores a cero y menores o iguales a 10}
  requiere: {|registro| > 0 }
  asegura: {res es el nombre del alumno cuyo promedio de notas es el más alto; si hay más de un alumno
con el mismo promedio de notas, devuelve el nombre de alumno que aparece primero en registro}
}
```

Ejercicio 4 (3 puntos)

```
problema seGraduoConHonores (registro: seq seq Char x seq Z , cantidadDeMateriasDeLaCarrera: Z,
alumno: seq Char ) : Bool {
  requiere: {No hay nombres de alumnos repetidos en registro}
  requiere: {Las notas de registro son todas iguales o mayores a cero y menores o iguales a 10}
  requiere: {cantidadDeMateriasDeLaCarrera > 0}
  requiere: {El alumno se encuentra en el registro }
  requiere: {|buenosAlumnos(registro)| > 0}
  asegura: {res <=> true si aproboMasDeNMaterias(registro, alumno, cantidadDeMateriasDeLaCarrera -1)
= true y alumno pertenece al conjunto de buenosAlumnos(registro) y el promedio de notas de finales de
alumno está a menos (estrictamente) de 1 punto del mejorPromedio(registro)}
}
```

Ejercicio 5 (1 punto)

Conteste marcando la opción correcta. El Testing es una técnica de verificación que sirve para:

- Demostrar que un programa es correcto.

- Probar propiedades de un programa.
- Encontrar fallas en un programa.