



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Primer Parcial (Haskell)

2do cuatrimestre 2023

Algoritmos y Estructuras de Datos I - Introducción a la Programación

Integrante	LU	Correo electrónico
Fausto N. Martínez	363/23	fnmartinez@dc.uba.ar



Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Viva la democracia

La elección periódica de los gobernantes es la base de los Estados Modernos. Este sistema, denominado "democracia" (término proveniente de la antigua Grecia), tiene diferentes variaciones, que incluyen diferentes formas de elección del/a máximo/a mandatario/a. Por ejemplo, en algunos países se eligen representantes en un colegio electoral (EEUU). En otros se vota a los/as miembros del parlamento (España). En nuestro país elegimos de forma directa la fórmula presidencial (Presidente/a y Vicepresidente/a) cada 4 años.

A continuación presentamos una serie de ejercicios que tienen como objetivo implementar funciones para sistema de escrutinio de una elección presidencial. Leer las descripciones y especificaciones e implementar las funciones requeridas en Haskell, utilizado solamente las herramientas vistas en clase.

Las fórmulas presidenciales serán representadas por tuplas ($\text{String} \times \text{String}$), donde la primera componente será el nombre del candidato a presidente, y la segunda componente será el nombre del candidato a vicepresidente.

En los problemas en los cuales se reciban como parámetro secuencias de fórmulas y votos, cada posición de la lista votos representará la cantidad de votos obtenidos por la fórmula del parámetro formulas en esa misma posición. Por ejemplo, si la lista de fórmulas es $[("Juan\ Pérez", "Susana\ García"), ("María\ Montero", "Pablo\ Moreno")]$ y la lista de votos fuera $[34, 56]$, eso indicaría que la fórmula encabezada por María Montero obtuvo 56 votos, y la lista encabezada por Juan Pérez obtuvo 34 votos.

1.1. Ejercicio 1 - Votos en Blanco

```
problema votosEnBlanco(formulas : seq < String × String >, votos : seq < ℤ >, cantTotalVotos : ℤ) : ℤ{
  requiere : { formulasValidas(formulas) }
  requiere : { |formulas| = |votos| }
  requiere : { Todos los elementos de votos son mayores o iguales que 0 }
  requiere : { La suma de todos los elementos de votos es menor o igual a cantTotalVotos }
  asegura : { res es la cantidad de votos emitidos que no correspondieron a ninguna de las fórmulas que se presentaron }
}
```

1.2. Ejercicio 2 - Fórmulas Válidas

```
problema formulasValidas(formulas : seq < String × String >) : Bool{
  requiere : { True }
  asegura : { (res = true) ↔ formulas no contiene nombres repetidos, es decir que cada candidato está en una única fórmula (no se puede ser candidato a presidente y a vicepresidente ni en la misma fórmula ni en fórmulas distintas) }
}
```

1.3. Ejercicio 3 - Porcentaje de Votos

```
problema porcentajeDeVotos(presidente : String, formulas : seq < String × String >, votos : seq < ℤ >) : ℝ{
  requiere : { La primera componente de algun elemento de formulas es presidente }
  requiere : { formulasValidas(formulas) }
  requiere : { |formulas| = |votos| }
  requiere : { Todos los elementos de votos son mayores o iguales que 0 }
  requiere : { Hay al menos un elemento de votos que es mayor estricto que 0 }
  asegura : { res es el porcentaje de votos que obtuvo la fórmula encabezada por presidente sobre el total de votos afirmativos }
}
```

Para resolver este ejercicio pueden utilizar la siguiente función que devuelve como Float la división entre dos números de tipo Int:

```
division :: Int → Int → Float
division a b = (fromIntegral a) / (fromIntegral b)
```

1.4. Ejercicio 4 - Próximo Presidente

```
problema proximoPresidente(formulas : seq < String × String >, votos : seq < ℤ >) : String{
  requiere : {La primera componente de algun elemento de formulas es presidente}
  requiere : {formulasValidas(formulas)}
  requiere : {|formulas| = |votos|}
  requiere : { Todos los elementos de votos son mayores o iguales que 0}
  requiere : { Hay al menos un elemento de votos que es mayor estricto que 0}
  requiere : {|formulas| > 0}
  asegura : {res es el candidato a presidente de formulas más votado de acuerdo a los votos contabilizados en votos}
}
```