

# Introducción a la Programación

## Primer Parcial - Turno Noche

- El parcial se aprueba con 6 puntos
- Podrás utilizar las siguientes funciones del prelude
  - Listas: head, tail, last, init, length, elem, ++
  - Tuplas: fst, snd
  - Operaciones lógicas: &&, ||, not
  - Constructores de listas: (x:xs), []
  - Constructores de tuplas: (x,y)
- Si querés utilizar Hunit para testear tu código [acá](#) tenés un script de ejemplo.

## ¡Vamos campeón!

En Exactas se está jugando un torneo de fútbol y la facultad le pidió a los alumnos de IP programar algunas funcionalidades en Haskell. Los datos con los que contamos para esto son los nombres de los equipos que participan del torneo, los nombres de los arqueros titulares de cada uno de dichos equipos, y la cantidad de goles recibidos por esos arqueros. Los nombres de los equipos y sus respectivos arqueros serán modelados mediante tuplas de tipo `(String, String)`, donde la primera componente representa el nombre del equipo, y la segunda representa el nombre del arquero titular de dicho equipo.

En los problemas en los cuales se reciban como parámetros secuencias *arquerosPorEquipo* y *goles*, cada posición de la lista *goles* representará la cantidad de goles

recibidos por el arquero del equipo que se encuentra en esa misma posición de *arquerosPorEquipo*. Por ejemplo, si la lista de *arquerosPorEquipo* es `[("Sacachispas","Neyder Aragón"), ("Fénix","Nahuel Galardi")]` y la lista de goles es `[3, 5]`, eso indicaría que Neyder Aragón recibió 3 goles, y Nahuel Galardi 5.

---

## 1) Atajaron Suplentes [1 punto]

problema *atajaronSuplentes* (*arquerosPorEquipo*: `seq<String x String>`, *goles*:`seq< Z >`, *totalGolesTorneo*: `Z`) : `Z` {  
  **requiere**: {*equiposValidos*(*arquerosPorEquipo*)}  
  **requiere**: {|*arquerosPorEquipo*| = |*goles*|}  
  **requiere**: {Todos los elementos de *goles* son mayores o iguales a 0}  
  **requiere**: {La suma de todos los elementos de *goles* es menor o igual a *totalGolesTorneo*}  
  **asegura**: {*res* es la cantidad de goles recibidos en el torneo por arqueros que no son los titulares en sus equipos}  
}

---

## 2) Equipos Válidos [3 puntos]

problema *equiposValidos* (*arquerosPorEquipo*: `seq<String x String>`) : `Bool` {  
  **requiere**: {True}  
  **asegura**: {(*res* = true)  $\Leftrightarrow$  *arquerosPorEquipo* no contiene nombres de clubes repetidos, ni arqueros repetidos, ni jugadores con nombre de club}  
}

---

### 3) Porcentaje de Goles [3 puntos]

problema porcentajeDeGoles (*arquero*: String,  
*arquerosPorEquipo*: seq<String x String>, *goles*: seq<Z >) :  
R {  
  **requiere**: {La segunda componente de algún elemento  
  de *arquerosPorEquipo* es *arquero*}  
  **requiere**: {equiposValidos(*arquerosPorEquipo*)}  
  **requiere**: {|*arquerosPorEquipo*| = |*goles*|}  
  **requiere**: {Todos los elementos de *goles* son mayores o  
  iguales a 0}  
  **requiere**: {Hay al menos un elemento de *goles* mayores  
  estricto a 0}  
  **asegura**: {*res* es el porcentaje de goles que recibió  
  *arquero* sobre el total de goles recibidos por arqueros  
  titulares}  
}

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)
```

---

### 4) Valla menos vencida [3 puntos]

problema vallaMenosVencida (*arquerosPorEquipo*:  
seq<String x String>, *goles*: seq<Z >) : String {  
  **requiere**: {equiposValidos(*arquerosPorEquipo*)}  
  **requiere**: {|*arquerosPorEquipo*| = |*goles*|}  
  **requiere**: {Todos los elementos de *goles* son mayores o

iguales a 0}

**requiere:**  $\{|goles| > 0\}$

**asegura:**  $\{res$  es alguno de los arqueros de  
*arquerosPorEquipo* que menos goles recibió de acuerdo a  
*goles*  
 $\}$

---