

Práctica Recursividad y Orden Superior

1. Definir la función **promediosAlumnos/1**, que dada una lista de alumnos devuelve una lista de tuplas que tenga el alumno y el promedio (Consideramos la división entera para el promedio y usamos la función `div`).

```
type Nombre = String
type Notas = [Number]
data Persona = Alumno {nombre :: Nombre, notas :: Notas}
Main> promediosAlumnos [(Alumno "juan" [8,6]), (Alumno "maria" [7,9,4]), (Alumno
"ana" [6,2,4])]
[("juan",7), ("maria",6), ("ana",4)]
```

2. Definir la función **promediosSinAplazos/1**, que dada una lista de listas, devuelve la lista de los promedios de cada lista-elemento, excluyendo los que sean menores a 6 que no se cuentan.

```
Main> promediosSinAplazos [[8,6],[6,6,4]]
[7,6]
```

3. Definir la función **aprobo/1**, que dado un alumno devuelve `True` si el alumno aprobó. *Aclaración: se dice que un alumno aprobó si todas sus notas son 6 o más.*

```
Main> aprobo (Alumno "manuel" [8,6,2,4])
False
```

4. Definir la función **aprobaron/1**, que dada una lista de alumnos, devuelve los nombres de los alumnos que aprobaron.

```
Main> aprobaron [Alumno "manuel" [8,6,2,4] , Alumno "elena" [7,9,8,7], Alumno
"ana" [6,2,4,2], Alumno "pedro" [9,6,7,10]]
["elena", "pedro"]
```

5. Definir la función **productos** que dado una lista de nombres de productos y una lista de precios, devuelve una lista de tuplas.

```
Main> productos ["melon", "zapallo", "palta"] [15, 10, 12, 7]
[("melon", 15), ("zapallo", 10), ("palta", 12)]
Definirla usando zip y usando zipWith
```

6. Dado una lista de flores donde cada una está representada de la siguiente forma:

```
data Flor= Flor {nombreFlor :: String, aplicacion:: String, cantidadDeDemanda::
Integer} deriving Show
```

```
rosa = Flor "rosa" "decorativo" 120
jazmin = Flor "jazmin" "aromatizante" 100
violeta= Flor "violeta" "infusión" 110
orquidea = Flor "orquidea" "decorativo" 90

flores = [orquidea, rosa,violeta, jazmin]
```

6.a) Definir `maximaFlorSegun` que permite conocer el nombre de la flor que es máxima según estos criterios

- La cantidad demandada
- La cantidad de letras de la flor
- El resto de la división de la cantidad demandada por 4

Resolverla evitando tener código duplicado y usando recursividad.

6 b) Dada una lista de flores determinar si están ordenadas de mayor a menor por cantidad de demanda.