

Las medidas

Se nos pide que hagamos un análisis de las decisiones económicas de un gobierno haciendo un programa en Haskell.

Vamos a modelar a un ciudadano con su *profesión*, su *sueldo anual*, la *cantidad de hijos* y una *lista de bienes*:

```
type Bien = (String,Float)
data Ciudadano = UnCiudadano {profesion :: String, sueldo :: Float,
                             cantidadDeHijos :: Float, bienes :: [Bien] } deriving Show
```

Como se ve, la profesión es un String, el sueldo es un entero, la cantidad de hijos un entero y los bienes son tuplas con descripción y valor del bien. Algunos ciudadanos de ejemplo:

```
homero = UnCiudadano "SeguridadNuclear" 9000 3 [("casa",50000), ("deuda",-70000)]
frink = UnCiudadano "Profesor" 12000 1 []
krabappel = UnCiudadano "Profesor" 12000 0 [("casa",35000)]
burns = UnCiudadano "Empresario" 300000 1
        [("empresa",1000000), ("empresa",500000), ("auto",200000)]
```

Tenemos también una ciudad:

```
springfield = [homero, burns, frink, krabappel]
```

Se piden las siguientes funciones. Sólo puede usarse recursividad en los puntos 4 ó 5, aunque no es necesaria.

1. **diferenciaDePatrimonio:** recibe una ciudad y dice cuál es la diferencia entre el ciudadano que más patrimonio tiene y el que menos patrimonio tiene. El patrimonio de cada ciudadano se obtiene sumando el valor de su sueldo y de sus bienes.
2. **tieneAutoAltaGama:** recibe un ciudadano y dice si tiene un auto de alta gama, ó sea, si tiene entre sus bienes un auto que valga más de 100000.
3. **Medidas:** se aplican a un ciudadano, y lo retornan modificado.
 - a. **auh:** Hace que aumente el sueldo de la persona en 1000 por cada hijo, si el patrimonio de la persona es menor a 0 (en otro caso, el ciudadano no cambia).

```
Main> auh (UnCiudadano "Doctor" 2000 2 [("deuda",-3000)])
UnCiudadano "Doctor" 4000 2 [("deuda",-3000)]
```
 - b. **impuestoGanancias:** si el sueldo supera el mínimo dado , disminuye su sueldo el 30% de la diferencia. Si no supera el mínimo, queda igual.

```
Main> impuestoGanancias 20000 (UnCiudadano "Doctor" 20100 0 [])
UnCiudadano "Doctor" 20070 0 []
```
 - c. **impuestoAltaGama:** si el ciudadano tiene algún auto de alta gama, disminuye su sueldo en un 10% del valor del auto, sino no disminuye nada.

```
Main> impuestoAltaGama (UnCiudadano "Profesor" 30000 0 [("auto",120000)])
UnCiudadano "Profesor" 18000 0 [("auto",120000)]
```
 - d. **negociarSueldoProfesion:** Esta medida recibe una profesión y un porcentaje. Si el ciudadano tiene esa profesión, entonces aumenta su sueldo el porcentaje indicado. Si no es su profesión, entonces queda igual.



Sólo en los puntos siguientes puede usarse recursividad, aunque no es necesaria:

4. Un gobierno está representado por la lista de años que gobernó, y por la lista de medidas que aplicó.

```
data Gobierno = UnGobierno {años :: [Float], medidas :: [Ciudadano->Ciudadano ]}
```

- Escribir la función constante **gobiernoA**, que devuelve un gobierno que gobernó entre 1999 y 2003, y aplicó impuesto a las ganancias de 30.000, negoció sueldo de profesores por 10%, negoció sueldo de empresarios por 40%, y aplicó impuesto a vehículos de alta gama y también aplicó la auh.
- Escribir la función constante **gobiernoB**, que devuelve un gobierno que gobernó desde 2004 hasta 2008, y aplicó impuesto a las ganancias de 40.000, negoció sueldo de profesores por 30%, y negoció sueldo de camioneros por 40%.
- Hacer la función **gobernarUnAño**, que recibe un gobierno y una ciudad, y aplica todas las medidas del gobierno a cada uno de los ciudadanos de la ciudad. Retorna la ciudad cambiada.
- Hacer la función **gobernarPeriodoCompleto**, que recibe un gobierno y una ciudad, y gobierna a esa ciudad todos los años del período (Es decir, gobierna tantos años a la ciudad como haya durado el mandato).
- Hacer la función **distribuyóRiqueza**, que dice si un gobierno hizo justamente eso en una ciudad. Esto es cuando luego de gobernar por todo su período, la diferencia de patrimonio es menor que al iniciar.

5. Un ciudadano especial

- Crear al ciudadano **kane**, que no tiene hijos, es empresario, tiene 100000 de sueldo, e infinitos trineos, todos llamados "Rosebud", y cada uno \$5 más caro que el anterior.

```
Main> kane
```

```
UnCiudadano "Empresario" 100000 0
```

```
[ ("Rosebud",5), ("Rosebud",10), ("Rosebud",15) ..... y así...
```

- Decir qué sucede con lo siguiente. Justificar conceptualmente ambas respuestas.

i. `Main> gobernarUnAño gobiernoA [kane]`

ii. `Main> gobernarUnAño gobiernoB [kane]`

6. Dar el tipo más genérico de f1:

```
f1 x y z = map (*x) . filter (y z)
```