

Un garaje ofrece como servicios cocheras fijas mensuales a sus clientes. Los datos de cada vehículo se mantienen a través de la siguiente lista:

```
autos = [("karl", "gtm066", 2000, "Volkswagen Gol"),
         ("achi", "fhu468", 2009, "Ford Focus"),
         ("mike", "rvm363", 1979, "Dodge 1500"),
         ("pato", "edi807", 2004, "Ford Ka"),
         ("rodri", "dfy642", 2006, "Volkswagen Gol")]
```

Cada tupla representa:

- El nombre del cliente
- La patente del auto
- El año de patentamiento
- El modelo del auto

Por otra parte se tiene el monto que le corresponde a cada modelo de auto en el siguiente formato:

```
precios = [(400, ["Volkswagen Gol", "Ford Focus"]),
           (560, ["Dodge 1500", "Ford Falcon", "Renault 4"]),
           (260, ["Ford Ka", "Suzuki Fun"])]
```

El formato que sigue la tupla es:

- Valor mensual a cobrar
- Lista de marcas de auto a las que se le cobra ese valor

Se cuenta con estas funciones:

```
modelo (_, _, _, m) = m
cliente (c, _, _, _) = c
```

```
anio (_, _, a, _) = a
patente (_, p, _, _) = p
```

```
find criterio = head . filter criterio
```

Codificar las funciones indicadas a continuación. En la resolución tenga presente que deben aparecer aplicados, al menos una vez, los siguientes conceptos:

- Aplicación parcial
- Composición
- Funciones de orden superior
- Listas por comprensión

No se puede usar recursividad, a menos que esté indicado en el punto.

Se pide

1) a) Saber cuál es el precio de una marca

```
>precioDe "Volkswagen Gol"
400
```

b) Saber cuánto se le cobra a un cliente

```
>cuantoLeCobra "karl"
400 ← porque Karl tiene un Volkswagen Gol y a esa marca se le cobra $ 400
```

2) Saber cuál es el dueño del auto más viejo.

```
>quienEsElMasRaton  
"mike"
```

En este punto sí se puede usar recursividad (y sólo en éste).

3) a. Queremos saber cuáles son las patentes de los autos que cumplen un determinado criterio. Desarrollar esa función.

```
>quienesCumplen unCriterio  
["patente 1", "patente 2", ...]
```

b. Utilizar la función desarrollada en 3.a para resolver:

- Qué autos fueron patentados en el año 2000
- Qué autos pagan más de 500 \$ mensuales de cochera
- Qué autos no son de "achi"
- Qué autos tienen un dueño cuyo nombre es de más de 5 letras (sí, al usuario le gusta jugar con requerimientos)

Nota: no se pueden definir funciones auxiliares

4) Queremos saber cuánto recauda el garaje (para eso hay que saber cuánto le cobra a cada cliente)

```
>recaudacionTotal  
2020 (2 Gol que pagan 400 (800) + 1 Ford Ka que paga 260 + 1 Ford  
Focus que paga 400 + 1 Dodge 1500 que paga 560)
```

5) Ahora nos interesa conocer la recaudación discriminando por modelo

```
>recaudacionPorModelo  
[("Volkswagen Gol", 800), ("Ford Focus", 400),  
 ("Dodge 1500", 560), ("Ford Ka", 260)]
```

Nota: para evitar que aparezcan los modelos duplicados se puede utilizar esta función

```
sinRepetidos [] = []  
sinRepetidos (x:xs) | elem x xs = sinRepetidos xs  
                  | otherwise = x:sinRepetidos xs
```

6) Definir una función que permita determinar si todos los clientes que le paso como parámetro son 0 km (el auto tiene que ser modelo 2010)

```
>todosSon0Km [("susana", "ioc182", 2010, "BMW"),  
              ("mirtha", "iae563", 2009, "Suzuki fun")]
```

False ← no todos son modelo 2010

7) Inferir los tipos de la función funcionHeavy

```
funcionHeavy a b c d | length a > c b = d  
                   | otherwise      = 0
```