

# Recuperatorio: Tema B - Haskell

## Ejercicio 1

a) Definir el tipo `Vehiculo` que consta de dos constructores `Moto` y `Auto`, que toman los siguiente parámetros:

- Constructor `Moto`: Tiene dos parámetros, el primero de tipo `Color` y el segundo de tipo `Potencia`
- Constructor `Auto`: Tiene tres parámetros, el primero de tipo `Color`, el segundo de tipo `EstiloAuto` y el tercero de tipo `Potencia`

Además:

- El tipo `Color` tiene constructores `Rojo`, `Azul`, `Blanco` y `Gris`, todos sin parámetros. El tipo **no debe** estar en la clase `Eq`.
- El tipo `EstiloAuto` tiene constructores `Sedan` (que indica que el auto tiene cuatro puertas) y `Coupe` (indica que es un auto con dos puertas), ambos sin parámetros. El tipo **no debe** estar en la clase `Eq`.
- El tipo `Potencia` es un sinónimo de `Int` que indica la cantidad de caballos de fuerza.

b) Programar la función

```
potencia_vehiculo :: Vehiculo -> Potencia
```

que devuelve la cantidad de caballos de fuerza que tiene el vehículo

c) Incluir el tipo `Vehiculo` en la clase `Ord` de manera tal que un vehículo se considere menor o igual según su potencia. Además incluir al tipo `Vehiculo` en la clase `Show`

## Ejercicio 2

a) Programar de manera recursiva la función

```
pintarCoupes :: [Vehiculo] -> Color -> [Vehiculo]
```

que dada una lista de vehículos `vs` y un color `c` devuelve una lista con los elementos de `vs`, cambiando el color de los autos que son de estilo `Coupe` al color `c`.

b) Escribir una lista de vehículos con **al menos tres elementos**, donde **debe haber al menos un auto** `Coupe` y **al menos una moto**.

c) Escribir el resultado de `pintarCoupes` para la lista del punto b)

## Ejercicio 3

Basados en el tipo `ListaAsoc` del *Proyecto 2*, programar recursivamente la función:

```
la_es_cota_inf :: ListaAsoc a b -> a -> Bool
```

que dada una lista de asociaciones `la` y una clave `x` devuelve `True` si `x` es menor o igual que todas las claves de `la` y `False` en caso contrario. Completar el tipado de la función para incluir los *type classes* necesarios para programarla.