

# Parcial 1 - Algoritmos I Taller: Tema C

Debés entregar el código completo en los campos correspondientes de cada ejercicio del formulario de entrega y subir el archivo .hs correspondiente. Este código debe poder ejecutarse en Haskell sin errores. Te recomendamos para ello que pruebes con diferentes ejemplos antes de entregar.

## Descripción del problema

En un sistema de gestión de inmobiliarias tenemos que representar inmuebles con distintas propiedades.

### Ejercicio 1:

Definir el tipo `Inmuable` que consta de tres constructores `Departamento`, `Casa`, `LocalComercial`, con los siguientes parámetros:

1. El constructor `Departamento` debe tomar como parámetro la cantidad de metros cuadrados, la cantidad de habitaciones, la dirección (consistente de un string indicando la calle y un entero indicando el número), el número de piso en el que se encuentra, y un booleano indicando si tiene portero eléctrico o no.
2. El constructor `Casa` debe tomar como parámetro la cantidad de metros cuadrados, la dirección (consistente de un string indicando la calle y un entero indicando el número), un booleano indicando si tiene patio, y un parámetro que indica si el techo es de loza o de madera (solo puede tener una de esas dos opciones).
3. En constructor `LocalComercial` debe tomar como parámetro la cantidad de metros cuadrados, la dirección (consistente de un string indicando la calle y un entero indicando el número) y un booleano indicando si tiene vidriera o no.

AYUDA: conviene definir un tipo separado para representar la Dirección.

### Ejercicio 2:

A partir del tipo definido en el punto anterior, definí los siguientes términos:

```
depto1 :: Inmuable
depto1 = <COMPLETAR>
```

que debe representar un departamento de 45 metros cuadrados, 1 habitación, ubicado en Santa Fé al 300, segundo piso, sin portero eléctrico.

```
casa1 :: Inmuable
casa1 = <COMPLETAR>
```

que debe representar una casa de 90 metros cuadrados, ubicada en Pueyrredón al 600, con patio y techo de loza.

```
comercio1 :: Inmuable
comercio1 = <COMPLETAR>
```

que debe representar un local comercial de 30 metros cuadrados ubicado en Colón al 2000, sin vidriera.

### Ejercicio 3:

Definir las siguientes funciones:

1. `esDeptoAlto :: Inmuable -> Bool`  
Devuelve True si el inmueble es un departamento ubicado en piso 8 o superior.
2. `esCasaGrande :: Inmuable -> Bool`  
Devuelve True si el inmueble es una casa con más de 100 metros cuadrados y tiene patio.
3. `esComercioConVidriera :: Inmuable -> Bool`  
Devuelve True si el inmueble es un local comercial con vidriera.

### Ejercicio 4:

Definir la función

```
casasConPatio :: [Inmuable] -> Int
```

que, dada una lista de inmuebles, devuelve la cantidad de casas con patio.

### Ejercicio 5:

Definir el tipo recursivo `cola de inmuebles`:

```
data ColaInmuable = Vacía | Encolada Inmuable ColaInmuable
    deriving Show
```

A partir de este tipo, definir la función

```
casasConTechoMadera :: ColaInmuable -> ColaInmuable
```

que, dada una cola de inmuebles `q`, devuelve la cola que contiene únicamente las casas que tienen techo de madera, en el mismo orden en que aparecen en `q`.