

### Ejercicio 3

[2 puntos]

Una forma de estructurar el sistema de ficheros dentro de un computador es mediante un **árbol de directorios**. Cada **directorio** puede contener varios elementos, en particular **ficheros** u otros directorios (**subdirectorios**). A los elementos del árbol (directorios o ficheros) se les denomina **nodos**, y cada uno de ellos tiene un nombre (cadena de texto). El nodo a partir del que parte todo el árbol de directorios es un directorio y se le denomina directorio **raíz**. Dos nodos dentro del mismo directorio no pueden tener el mismo nombre (se distingue entre mayúsculas y minúsculas). Un fichero tiene un **tamaño** concreto (en bytes), mientras que el tamaño de un directorio es la suma de los tamaños de todos los nodos que contiene.

Para referenciar de forma textual cualquier nodo dentro de ese árbol de directorios se utiliza su *path* o **ruta**: una secuencia de directorios y subdirectorios desde el directorio raíz hasta llegar al elemento referenciado, que en este ejercicio se representará mediante una lista de cadenas de texto que llevan desde la raíz hasta el nodo específico. Sobre este tema, y utilizando programación funcional mediante el lenguaje Haskell, **se pide**:

- (a) Diseña un tipo de datos `ArbolDirectorios` que represente un arbol de directorios.
- (b) Diseña una función `stat` que devuelva el tamaño de un nodo del arbol indicado por una ruta. Si la ruta no existe en el arbol de directorios entonces devuelve cero.

```
stat :: ArbolDirectorios -> [String] -> Integer
```

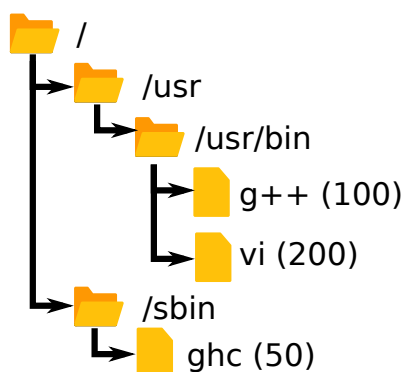
- (c) Diseña una función `find` que devuelva una lista con todas las rutas posibles de un arbol de directorios. Se deberá incluir tanto la ruta que lleva a cada directorio como la ruta que lleva a cada uno de los subdirectorios.

```
find :: ArbolDirectorios -> [[String]]
```

Ejemplo:

**Arbol de directorios**      **Resultados**

ejemplo



```
*Main> stat ejemplo ["sbin","ghc"]
50
*Main> stat ejemplo ["usr","bin","g++"]
100
*Main> stat ejemplo ["usr","bin","vi"]
200
*Main> stat ejemplo []
350
*Main> stat ejemplo ["usr"]
300
*Main> stat ejemplo ["var","share"]
0
*Main> find ejemplo
[[[] , ["usr"] , ["usr","bin"] , ["usr","bin","g++"] ,
 ["usr","bin","vi"] , ["sbin"] , ["sbin","ghc"]]
```

Se permite definir tipos de datos y funciones auxiliares adicionales si lo consideras necesario. Se valorará positivamente la **brevedad y concisión** de las soluciones propuestas. Se puede utilizar cualquier función del `Prelude` de Haskell, pero no otras funciones del lenguaje.