



## Práctica 4: Práctica Complementaria para el Parcial

1. Escriba definiciones en ML de las siguientes funciones. En cada caso, pueden utilizar funciones auxiliares para facilitar la definición de la función pedida:

- a. **cubos**: dada una lista de enteros, devuelve una lista con los cubos de cada elemento. *Ejemplo*: `cuadrados([1, 2, 3]) = [1, 8, 27]`.
- b. **orLista**: dada una lista de booleanos, devuelve `true` si al menos un elemento de la de la lista es `true`. *Ejemplos*: `orLista([true, true, false]) = true` y `orLista([false, false]) = false`.
- c. **primerosLista**: dada una lista de listas de enteros, devuelva una lista con el primer elemento de cada lista. Si alguna de las listas es vacía, poner un `-1` en ese caso. *Ejemplo*: `primerosLista [[1, 2, 3], [4, 5], [], [6, 7]] = [1, 4, -1, 6]`.
- d. **longitudPalabras**: dada una lista de strings, devuelve una lista con la longitud de cada string en la lista argumento. *Ejemplo*: `longitudPalabras ["hola", "ser", "cuadro"] = [4, 3, 6]`
- e. **elim**: dados un entero y una lista de enteros, devuelve la lista luego de eliminar todas las apariciones del entero dado. *Ejemplo*: `elim(4, [1, 2, 3, 4, 5, 6, 4, 7]) = [1, 2, 3, 5, 6, 7]`

2. Queremos definir la función `repetir`, que dado un entero `n`, nos devuelva una lista de longitud `n`, con todos sus elementos iguales a `n`. *Es decir*: `repetir(8) = [8, 8, 8, 8, 8, 8, 8, 8]`, `repetir(3) = [3, 3, 3]`, `repetir(0) = []`.

- a. Definir `repNM`, que dados un `n` y un `m`, repite `m` veces el número `n` en una lista. *Ejemplo*: `repNM(5, 3) = [5, 5, 5]`.
- b. Utilizar `repNM` para definir `repetir`.

3. 2. Analice la siguiente función `alts` definida a continuación:

```
fun alts ([], xs, ys) = (xs, ys)
  | alts ([x], xs, ys) = (x::xs, ys)
  | alts (x::y::t, xs, ys) = alts (t, x::l1, y::l2);
```

- a. Dé el tipo de la función `alts`.
- b. Explique, paso por paso, la ejecución de la función: `alts([2, 4, 2, 1, 5], [], [])`.
- c. ¿Qué hace la función `alts`?

4. Defina una función `letraAparece` que dados un carácter y una cadena, defina si el carácter aparece en la cadena o no. Es decir: `letraAparece("#a", "hola") = true` y `letraAparece("#b", "hola") = false`.

Ayuda: puede ser conveniente definir una función auxiliar que tome un carácter y una lista de caracteres como argumentos, y utilizar esa función para definir `letraAparece`.

5. Se utilizará la siguiente definición de árbol:

```
- datatype ('etiqueta) arbol =
    Nodo of 'etiqueta * 'etiqueta arbol list;
datatype 'a arbol = Nodo of 'a * 'a arbol list
```

- a. Dé 2 ejemplos de este tipo de dato.
- b. Explique qué tipo tiene esta función y qué calcula:  

```
fun sum(Nodo(a, nil)) = a
  | sum(Nodo(a, t::ts)) = sum(t) + sum(Nodo(a, ts));
```
- c. Defina una función `prod` que tome un elemento de tipo `int` `arbol` y calcule el producto de todos los nodos.
- d. Defina una función `andTree` que tome un elemento de tipo `bool` `arbol` y devuelva `true` únicamente si todos los nodos tienen la etiqueta `true`.