

Programación Declarativa

2008-2009

Práctica 4

1. Redefina las funciones ***nth***, ***append***, ***concat***, ***flatten*** y ***map***, del módulo *List*, sin utilizar ese módulo ni la función predefinida (**@**).
2. Defina una función ***del_fst*** : **'a -> 'a list -> 'a list** que elimine la primera aparición (si la hay) de un elemento en una lista.
Es decir, ***del_fst* x l** será la lista **l** si **x** no aparece en **l**, o, en otro caso, la concatenación de dos listas **l1** y **l2** (**l1 @ l2**) tales que **x** no aparece en **l1** y que **l = l1 @ x :: l2**.
(Así, por ejemplo, **del_fst 3 [2; 3; 5; 3; 2]** = **[2; 5; 3; 2]** y **del_fst 4 [2; 3; 5; 3; 2]** = **[2; 3; 5; 3; 2]**).
3. Defina una función ***del_all*** : **'a -> 'a list -> 'a list** que elimine todas las apariciones de un elemento en una lista.
(Así, por ejemplo, **dell_all 3 [2; 3; 5; 3; 2]** = **[2; 5; 2]** y **del_all 4 [2; 3; 5; 3; 2]** = **[2; 3; 5; 3; 2]**).
4. Utilizando el tipo de dato **num** del módulo **Num** de la **librería Num**, redefina la función ***fact*** para que tenga tipo **int -> Num.num**, de modo que puedan calcularse valores del factorial arbitrariamente grandes.

```
let rec fact n = if n = 0 then 1 else n * fact (n-1);;
```

5. Del mismo modo, redefina la función ***fib*** para que tenga tipo **int -> Num.num**.

```
let rec fib n = if n < 2 then n else fib (n-1) + fib (n-2);;
```

6. ¿Para qué valores se salían de rango las funciones ***fact*** y ***fib*** cuando estaban definidas con tipo **int -> int**?
7. Redefina las funciones ***fact***: **int -> Num.num** y ***fib***: **int -> Num.num** utilizando sólo recursividad terminal.