

# PROGRAMACIÓN FUNCIONAL

## 23 de Junio de 2009

1.- (1.5 puntos) Definir, usando la función `map`, una función `plural` que añada la letra 's' a las palabras de una lista que terminen en vocal.

2.- (1 punto) Siendo los tipos de las funciones

```
f :: a -> (b,a)
```

```
g :: b -> a -> c
```

infiere el tipo de `h` sabiendo que `h = (uncurry g).f`

3.- (1.5 puntos) Definir una función que dada una lista `[x1,x2, ...]` produzca la lista de todos sus pares adyacentes, es decir la lista

```
[(x1,x2), (x2,x3), (x3,x4), ...]
```

Si la lista es vacía o tiene un único elemento, el resultado deberá ser la lista vacía.

4.- (1.5 puntos) Dar una definición equivalente, pero sin utilizar la función `reverse` (ni ninguna equivalente a ella), a la siguiente función

```
f = reverse . dropWhile (==0) . reverse
```

¿Cuál es el valor de `f [1,0,0,2,3,0,0,0]`?

5.- (2.5 puntos)

a) Definir la suma binaria sobre el siguiente tipo de datos

```
data DigBin = Cero | Uno
```

Dicha operación suma debe dar como resultado el par formado por el resultado de la suma y el acarreo.

b) Usando la función anterior definir la suma de enteros representados en binario por la lista de sus dígitos, esto es

```
type NumBin = [DigBin]
```

6.- (2 puntos) Dar dos definiciones equivalentes, una con listas intensionales y otra con una estructura cíclica, de una función que devuelva la lista infinita

```
[(1,3), (5,7), (9,11), (13,15), ...]
```