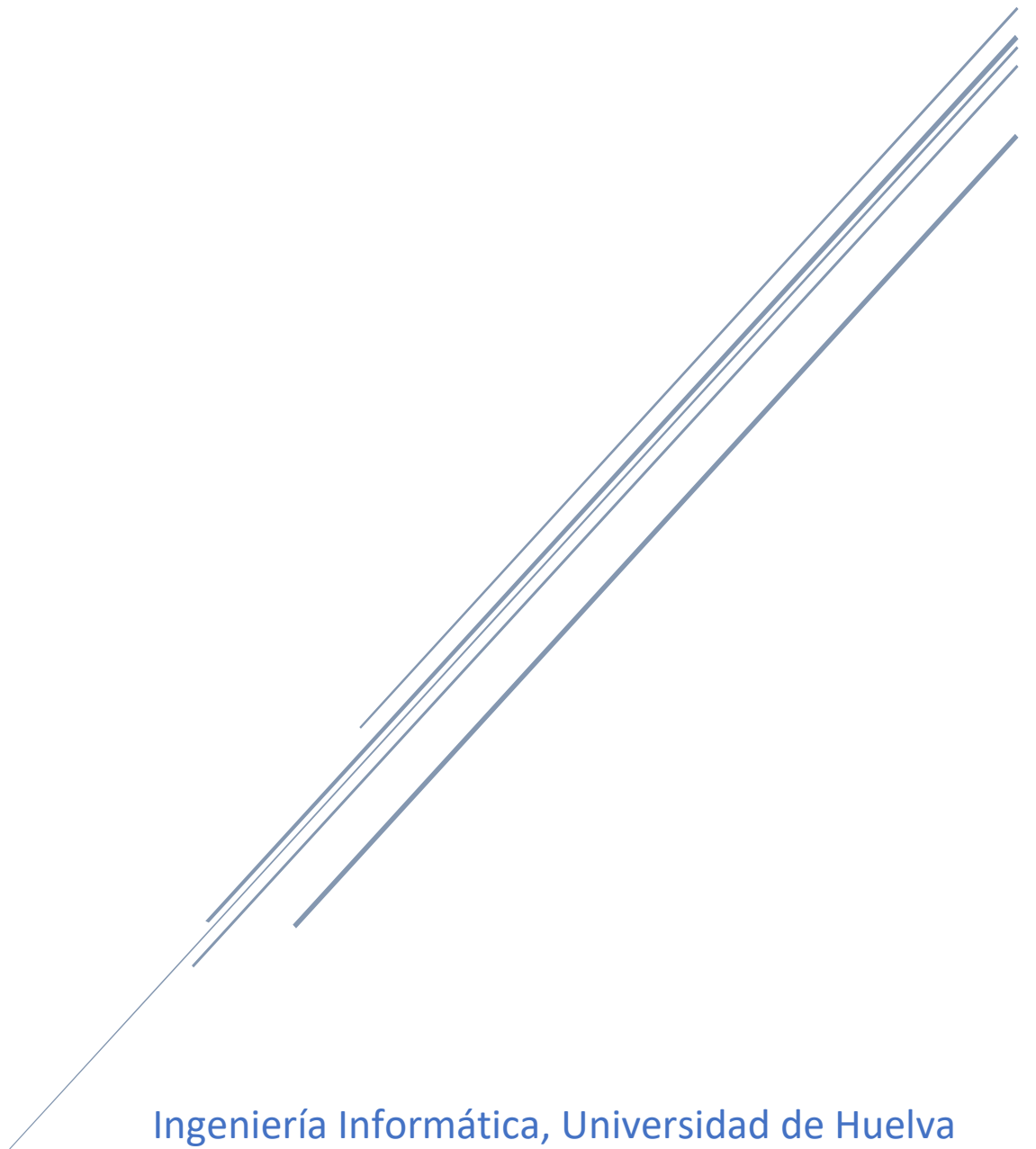


# PRÁCTICA 1

Alba Márquez Rodríguez



Ingeniería Informática, Universidad de Huelva  
Modelos Avanzados de Computación

cambia\_el\_primer(a,b)

```
-- 1 cambia_el_primer(a,b): cambia el primer valor de la lista b por el valor de a
--function declaration
cambia_el_primer :: Int -> [Int] -> [Int]
--function definition
cambia_el_primer a list = a:(tail list)

-- Hugs> :reload
-- Main> cambia_el_primer 1 [2,3,4]
-- [1,3,4]
```

Con tail list se coge la parte final de la lista menos el primer elemento. Esto devuelve una lista a la que hay que concatenarle el elemento pasado como parámetro al principio.

cambia\_el\_n(a,n,b)

```
-- 2 cambia_el_n(a,n,b): cambia el valor de la posicion n de la lista a por el valor de a
--function declaration
cambia_el_n :: Int -> Int -> [Int] -> [Int]
--function definition
cambia_el_n a n list = (take (n-1) list) ++ [a] ++ (reverse(take ((length list)-(n))(reverse(list))))

--cambia_el_n 0 3 [1..10]
--[1,2,0,4,5,6,7,8,9,10]

{-----}
```

Se concatenan 3 listas:

1º la lista original se cogen los n-1 primeros elementos

2º el elemento que se quiere insertar

3º De la lista invertida se cogen los primeros, al ser invertida sería equivalente a coger los últimos  $\text{length}(\text{lista}) - n$  elementos. Para volver a poner los números en el orden original se vuelven a invertir.

Haciendo al principio n-1 se consiguen coger todos los elementos menos el que vamos a cambiar por el elemento que vamos a insertar.

get\_mayor\_abs(a)

```
{-----}
-- 3 get_mayor_abs(a): devuelve el mayor numero en valor absoluto de la lista a
--function declaration
get_mayor_abs :: [Int] -> Int
--function definition
get_mayor_abs list = maximum ([abs(maximum list)] ++ [abs(minimum list)])

--Main> get_mayor_abs [1,4,100,124,12,6,0]
--124
{-----}
```

Se coge el valor máximo y mínimo de la lista y se toman sus valores absolutos. De estos dos valores absolutos se vuelve a hacer el máximo y este será el valor absoluto del mayor número en valores absolutos.

Esto se ha hecho entendiendo que si tenemos una lista [-10,5,7,8] la salida será 10. Ya que se cogerá -10 como mínimo y 8 como máximo y al hacerle los absolutos quedará como 10 y 8, cuyo máximo es 10.

num\_veces(a,b)

```
{-----}

-- 4 num_veces(a,b): devuelve la cantidad de veces que aparece el valor a en la lista b

--function declaration
num_veces:: Int -> [Int] -> Int
--function definition
num_veces a list = length(filter (==a) list)

{-----}
```

Devuelve el length de una lista que solo contenga el valor ==elemento introducido por parámetro haciéndole un filter a la lista original.

palabras\_mayores\_n(n,a)

```
{-----}  
  
-- 5 palabras_mayores_n(n,a): devuelve una lista con las palabras mayores que n  
--function declaration  
palabras_mayores_n :: Int -> [String] -> [String]  
--function definition  
palabras_mayores_n n list = filter (\x -> length x > n) list
```

Siendo x cada elemento de list, si su lengthes superior a el número introducido como n. Entonces se queda en la lista que saldrá como output.

## Ejercicios con las funciones vistas

1)

```
Hugs> [minimum [-10..10]] ++ [maximum [-10..10]]  
[-10,10]  
..
```

De una lista coge el mínimo y el máximo y crea una lista con estos dos valores.

2)

```
Hugs> (tail [1..10]) ++ [head [1..10]]  
[2,3,4,5,6,7,8,9,10,1]  
.....
```

Dada una lista coge el primer elemento y lo inserta al final de la lista

3)

```
Hugs> reverse([0] ++ (reverse([-10..10])))  
[-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,0]  
Hugs>
```

Hace lo mismo que la función anterior pero llamando a otras funciones. (Otro enfoque).

4)

```
Hugs> length(filter (\x -> x > 0) [-10..10])  
10  
..
```

Devuelve el número de elementos de la lista que cumple la condición  $x > 0$

5)

```
Hugs> (filter (\x -> x < 5) [1..10]) ++ [5] ++ (filter (\x -> x > 5) [1..10])  
[1,2,3,4,5,6,7,8,9,10]  
Hugs> (filter (\x -> x < 5) [1..10]) ++ [0] ++ (filter (\x -> x > 5) [1..10])  
[1,2,3,4,0,6,7,8,9,10]
```

Dada una lista de enteros la separa en dos listas, la primera los números menores a  $n$  (en este caso  $n=5$ ) y la segunda los números mayores a  $n$  ( $n=5$ ). Entre las dos listas inserta un número para separarlas. El número elegido en el ejemplo es 0.