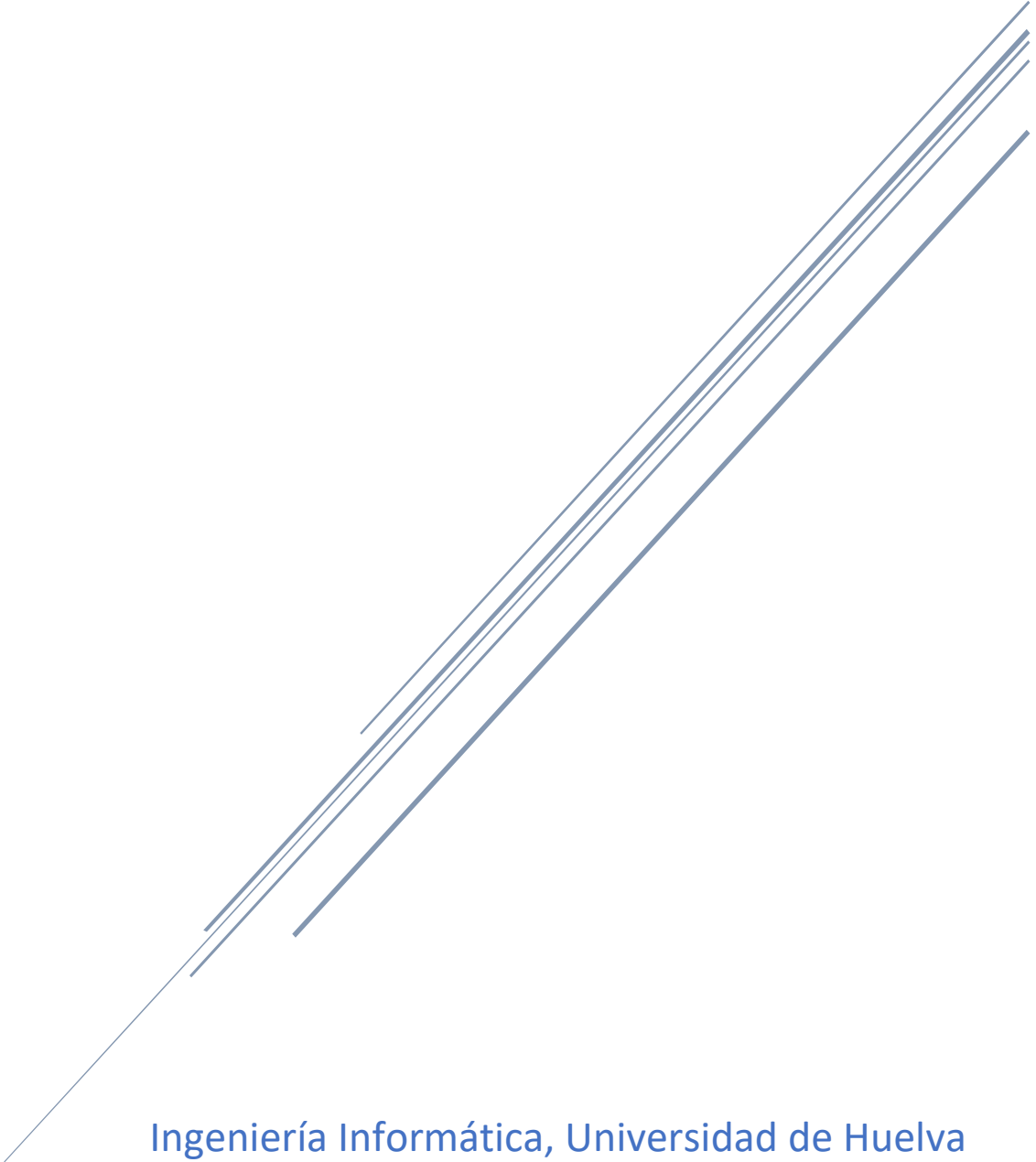


# PRÁCTICA 3: ENTRADA Y SALIDA

Alba Márquez-Rodríguez



Ingeniería Informática, Universidad de Huelva  
Modelos Avanzados de Computación

Contenido

Enunciado ..... 2

Código Fuente ..... 3

    AuxiliarFunctions ..... 3

    Código principal..... 3

Descripción del código ..... 4

    AuxiliarFunctions ..... 4

    Código principal..... 5

Ejecución ..... 6

## Enunciado

Implementación de un Juego. El programa le pide al jugador humano que piense un número entre 1 y 100 y tratará de acertar el número que ha pensado preguntando al jugador. El jugador responderá encontrado, mayor o menor y en función de la respuesta, se realizará una modificación del número buscado mejorando el ejercicio que vimos en clase que realizaba una búsqueda secuencial.

Esta modificación se realizará calculando el nuevo número de la siguiente forma:

$\text{proximo} = (x+y) \text{ div } 2$

El programa finalizará su ejecución cuando el número pensado por el jugador haya sido encontrado.

Una posible ejecución sería para el número 68: Main> juego\_busqueda\_mejorada

Piensa un número entre el 1 y el 100.

Es el número 50? [es mayor? / es menor? / Encontrado]: mayor

Es el número 75? [es mayor? / es menor? / Encontrado]: menor

Es el número 62? [es mayor? / es menor? / Encontrado]: mayor

Es el número 68? [es mayor? / es menor? / Encontrado]: encontrado

El juego ha finalizado

Es imprescindible la utilización de módulos propios para la resolución del ejercicio, que contenga funciones como: pregunta, próximo, mostrar\_error.. etc

## Código Fuente

Para la implementación se han utilizado dos ficheros, el fichero de funciones auxiliares y el fichero principal con la práctica.

### AuxiliarFunctions

```
1 module AuxiliarFunctions(  
2     next,  
3     next_float,  
4     toInt,  
5     error_code  
6 ) where  
7  
8 --next_float :: Int -> Int -> Float  
9 next_float x y = ((x+y) / 2.0)  
10  
11 next :: Int -> Int -> Int  
12 next x y = div (x+y) 2  
13  
14 toInt :: Float -> Int  
15 toInt x = round x  
16  
17 -- ERRORES:  
18 -- 1: Numero fuera de los limites por abajo (<0)  
19 -- -1: Numero fuera de los limites por arriba (>100)  
20 -- 0: Palabra introducida no dentro de las ordenes  
21  
22 error_code code =  
23     if (code == 0) then print $ "Error, introduzca 'mayor', 'menor' o 'encontrado'"  
24     else if (code == (-1)) then print $ "Error, el numero no puede ser menor de 0"  
25     else if (code == 1) then print $ "Error, el numero no puede ser mayor de 100"  
26     else print $ "Codigo de error no contemplado"  
27
```

### Código principal

```
1 import AuxiliarFunctions  
2  
3 main = do  
4     busca_numero 0 100  
5     putStrLn $ "FIN DEL JUEGO"  
6  
7 busca_numero :: Int -> Int -> IO()  
8 busca_numero start end = do  
9     putStrLn $ "Es el numero el " ++ (show (next start end)) ++ "?"  
10    respuesta <- getLine  
11    if (respuesta == "encontrado") then  
12        return ()  
13    else if (respuesta == "mayor") then  
14        if (next start end >= 100) then do  
15            error_code 1  
16            busca_numero start end  
17        else  
18            busca_numero (next start end) end  
19    else if (respuesta == "menor") then  
20        if (next start end <= 0) then do  
21            error_code (-1)  
22            busca_numero start end  
23        else  
24            busca_numero (start) (next start end)  
25    else  
26        do  
27            error_code 0  
28            busca_numero start end
```

## Descripción del código

### AuxiliarFunctions

Se han implementado 3 funciones auxiliares, una de ella está duplicada ya que primero se realizó con floats y finalmente se terminó con la de enteros.

**Next:** aplica la formula dada en la práctica para calcular el siguiente número. Como se ha indicado se aplica de forma que devuelva en una versión un número float y en otra un número entero.

**tolnt:** se diseñó para transformar un número de float a entero pero finalmente no se terminó utilizando.

**Error\_code:** código para la gestión de errores según el código pasado por parámetro. Simplemente hace un print mostrando por pantalla el error que ha saltado.

Estas funciones serán llamadas y usadas en el código principal.

### Código principal

Para llamar al código tendremos que correrlo con “main” en WinHugs. Esto llamará a la primera que llama a busca\_numero con los parámetros inicializados con 0 y 100. Ya que el número a buscar estará entre estos dos.

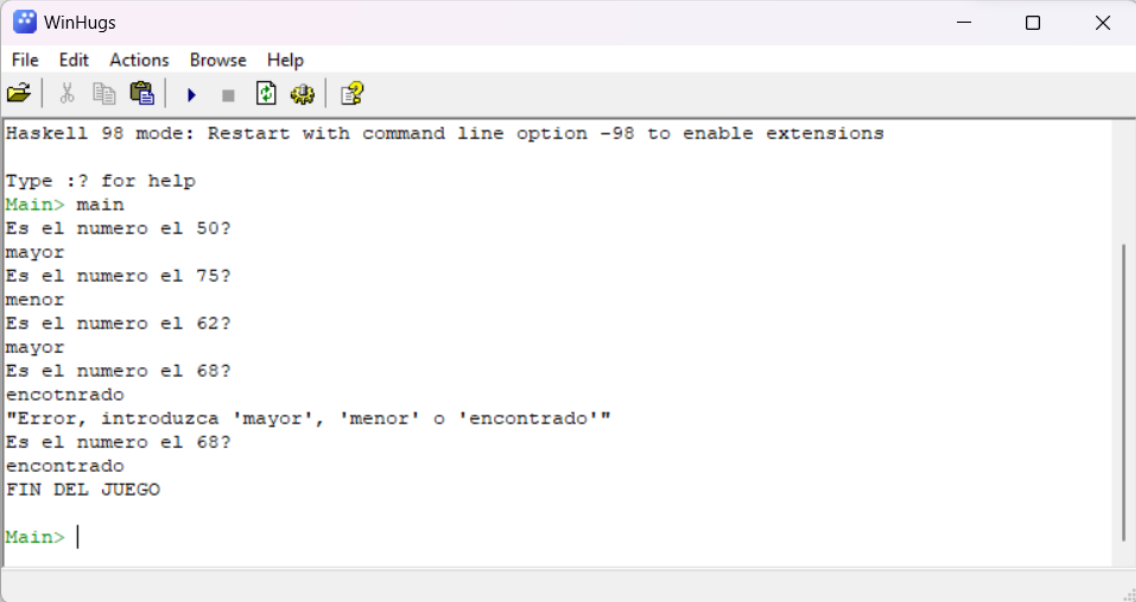
Lo primero que hará será preguntar si el número es el que está buscando la persona. Lo pregunta llamando a la función next que aplica a los dos números pasados por parámetro la formula dada en la práctica.

Con getLine obtenemos el input del usuario. Lo siguiente será analizar este input y dependiendo de lo dado recibiremos un output diferente. Hay 3 posibilidades:

- **Encontrado:** si el número ha sido encontrado el programa termina
- **Menor:** en el caso de ser menor buscará el número en la mitad inferior de los parámetros introducidos para la iteración actual. Y volverá a correr busca\_numero con estos nuevos parámetros. En el caso de que el siguiente numero se salga de los límites (0-100) sacará un error y volverá a ejecutar el programa con los parámetros iniciales antes del error.
- **Mayor:** en el caso de ser mayor buscará el número en la mitad superior de los parámetros introducidos para la iteración actual. Y volverá a correr busca\_numero con estos nuevos parámetros. En el caso de que el siguiente numero se salga de los límites (0-100) sacará un error y volverá a ejecutar el programa con los parámetros iniciales antes del error.

En el caso de no ser ninguno de estos saldrá el error\_code 0 que quiere decir que el input no es ninguno de los esperados y repetirá de nuevo la pregunta con los mismos parámetros para que el usuario introduzca una opción posible (las anteriores).

## Ejecución



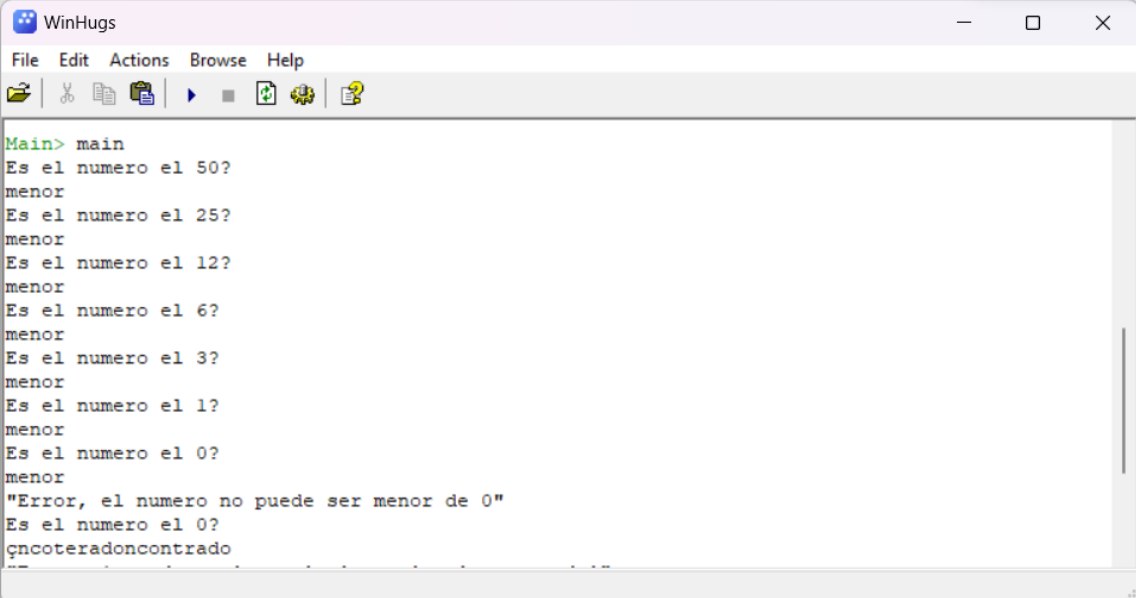
WinHugs

File Edit Actions Browse Help

Haskell 98 mode: Restart with command line option -98 to enable extensions

```
Type :? for help
Main> main
Es el numero el 50?
mayor
Es el numero el 75?
menor
Es el numero el 62?
mayor
Es el numero el 68?
encotnrado
"Error, introduzca 'mayor', 'menor' o 'encontrado'"
Es el numero el 68?
encontrado
FIN DEL JUEGO

Main> |
```



WinHugs

File Edit Actions Browse Help

```
Main> main
Es el numero el 50?
menor
Es el numero el 25?
menor
Es el numero el 12?
menor
Es el numero el 6?
menor
Es el numero el 3?
menor
Es el numero el 1?
menor
Es el numero el 0?
menor
"Error, el numero no puede ser menor de 0"
Es el numero el 0?
cncoteradoncontrado
```