



Universidad
de Huelva



Universidad de Huelva

GRADO EN INGENIERÍA INFORMÁTICA

PRÁCTICA 3

ENTRADA, SALIDA Y MÓDULOS EN HASKELL

Autor: Alberto Fernández Merchán

Asignatura: Modelos Avanzados de Computación

1. Introducción

En esta práctica se pide programar un *juego* en el que el usuario piense un número del 1 al 100 y nuestro programa trate de adivinarlo haciendo uso de las indicaciones del jugador.

En el caso de que el usuario indique que el número se ha encontrado, se termina el juego. Por el contrario, si el usuario indica que el número que ha pensado es menor o mayor, se reduce el intervalo a la mitad.

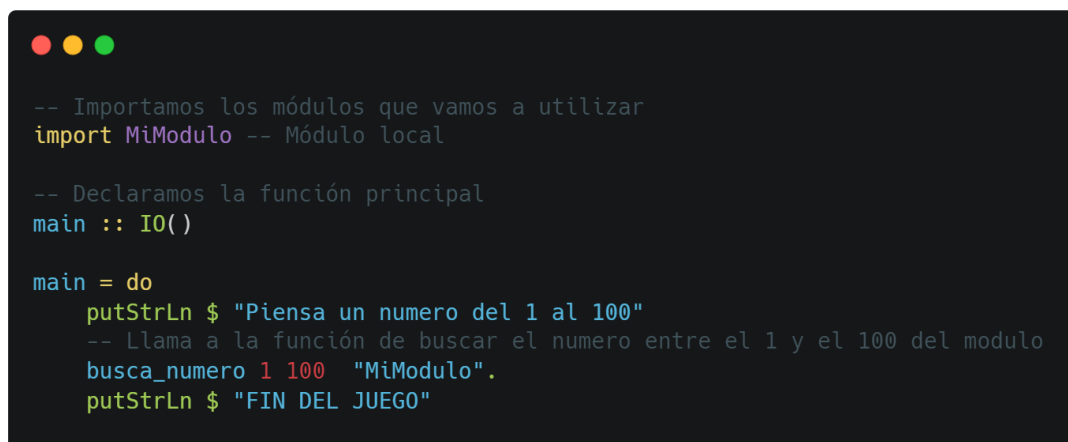
2. Organización del código

El código se organiza en dos ficheros: el **fichero principal** (*main.hs*) y el **fichero módulo** (*MiModulo.hs*) que contiene las funciones que utilizará el principal.

2.1. Main

El fichero principal lo único que hace es:

- Importar el módulo local para poder utilizar la función **busca_numero**.
- Declarar y definir la función principal utilizando la cláusula **do** para que ejecute todo el código seguido.



```
-- Importamos los módulos que vamos a utilizar
import MiModulo -- Módulo local

-- Declaramos la función principal
main :: IO()

main = do
    putStrLn $ "Piensa un numero del 1 al 100"
    -- Llama a la función de buscar el numero entre el 1 y el 100 del modulo
    busca_numero 1 100 "MiModulo".
    putStrLn $ "FIN DEL JUEGO"
```

Figura 1: Programa Principal

2.2. MiModulo

El módulo que he creado contiene únicamente dos funciones:

1. **generaProximo**: Se encarga de generar el próximo valor por el que va a preguntar el programa. Lo hace calculando el valor medio entre el primer valor del intervalo y el último.
2. **busca_numero**: Se encarga de ejecutar el *juego*. Muestra el valor calculado y, dependiendo de la respuesta del usuario, termina el juego o **reduce el intervalo de búsqueda**. Para obtener la respuesta del usuario se ha utilizado la función **getLine** y se ha usado también el módulo *Data.Char* junto con la función *map* para convertir a minúsculas la respuesta del usuario.

```

module MiModulo
( generaProximo,
  busca_numero
) where
import Data.Char -- Módulo para trabajar con los caracteres
generaProximo :: Int -> Int -> Int
generaProximo a b = div (a + b) 2

-- Declaramos la función que ejecuta el juego.
-- a: el número mínimo
-- b: el número máximo
busca_numero :: Int -> Int -> IO()
busca_numero a b = do
    -- El número que pregunta será el que se encuentre a la mitad del intervalo
    let proximo = generaProximo a b
    -- Muestra por pantalla el mensaje
    putStrLn $ "Es el " ++ (show proximo) ++ "?"
    -- Obtiene la respuesta del usuario
    respuesta <- getLine
    -- La convierte en minúsculas
    let resp = map toLower respuesta

    -- Si a = b entonces solo hay una opción.
    if (a == b) then
        putStrLn $ "La unica opcion es: " ++ (show proximo) ++ "."
    -- Si es el número, entonces termina
    else if (resp == "encontrado") then
        return ()
    -- Si es mayor, llama a la función con el intervalo mayor.
    else if (resp == "mayor") then
        busca_numero proximo b
    -- Si es menor, llama a la función con el intervalo menor.
    else if (resp == "menor") then
        busca_numero a proximo
    -- En otro caso se considera un error y se llama a la funcion
    -- con los mismos parámetros.
    else
        do
            print $ "Error"
            busca_numero a b

```

Figura 2: Módulo de funciones que usa el programa

3. Funcionamiento

A continuación se muestran capturas del funcionamiento del programa. En este caso estaremos buscando el número **33**:

```
--  
Main> main  
Piensa un numero del 1 al 100  
Es el 50?  
menor  
Es el 25?  
mayor  
Es el 37?  
menor  
Es el 31?  
mayor  
Es el 34?  
menor  
Es el 32?  
mayor  
Es el 33?  
encontrado  
FIN DEL JUEGO
```

Figura 3: Funcionamiento del programa.

Para el número **83** sería la siguiente secuencia:

```
Main> main  
Piensa un numero del 1 al 100  
Es el 50?  
mayor  
Es el 75?  
mayor  
Es el 87?  
menor  
Es el 81?  
mayor  
Es el 84?  
menor  
Es el 82?  
mayor  
Es el 83?  
encontrado  
FIN DEL JUEGO
```

Figura 4: Funcionamiento del programa.