

# **PRÁCTICA DE LENGUAJES DE PROGRAMACIÓN: CUATRO EN RAYA HASKELL**

**Nombre:** Oliver Urones García  
**Asignatura:** Lenguajes de Programación  
**Curso:** 2º G.I.I.S.I.

## Explicación de las funciones

Función para convertir un String a un Int  
`atoi::String->Int`

Función que crea el tablero inicial  
parámetros: `m n` -> dimensiones del tablero  
devuelve: Una lista de listas que será el tablero  
`crea_tablero :: Int -> Int -> [[Int]]`

Función que realiza una jugada  
Se le pasa el tablero, el turno del jugador la fila y la columna.  
parámetros: `tablero turno fila columna` ->  
devuelve: Una lista de listas que será el tablero con la nueva jugada hecha  
`jugada :: [[Int]] -> Int -> Int -> Int -> [[Int]]`

Función que realiza una jugada para el ordenador  
Se le pasa las dimensiones del tablero (`n,m`), el tablero el turno y la columna  
parámetros: `n m tablero turno columna`  
devuelve: Una lista de listas con el tablero con la jugada hecha por el ordenador  
`jugada_ordenador :: Int -> Int -> [[Int]] -> Int -> Int -> [[Int]]`

Función que genera una lista de pares `[(1,-1),(2,-2),...((n `div` 2),-(n `div` 2))]`  
Esta lista servirá para que el ordenador elija una columna más al centro usando los valores de la lista como índices para las columnas  
parámetros `n` -> Anchura del tablero  
devuelve: Una lista de pares  
`genera_direcciones :: Int -> [(Int, Int)]`

Función que convierte la lista de pares devuelta por la función `genera_direcciones` en una lista  
`[(1,-1),(2,-2),...((n `div` 2),-(n `div` 2))] --> [1,-1,2,-2,...,(n `div` 2,-(n `div` 2))]`  
parámetros: `(x:xs)` -> Lista de pares  
devuelve: Una lista  
`genera_lista_direcciones :: [(Int, Int)] -> [Int]`

Función que elige la columna más al medio donde se puede poner una ficha  
Se le pasa el tablero, la altura, la columna y la lista devuelta por `genera_lista_direcciones`  
parámetros: `tablero m columna (x:xs)`  
devuelve: El índice de la columna encontrada  
`columna_mas_al_centro :: [[Int]] -> Int -> Int -> [Int] -> Int`

Función que pone un 1 o un 2 en el tablero en función del jugador que le toque jugar

Se le pasa la lista correspondiente a la fila donde va la ficha, la columna el turno del jugador y un índice para recorrer la lista

parámetros: (x:xs) columna turno n

devuelve: La lista con la ficha colocada en la posición correspondiente

pon\_ficha :: [Int] -> Int -> Int -> Int -> [Int]

Función que busca la máxima fila donde se puede poner una ficha en una columna

Se le pasa el tablero, la fila y la columna

parámetros: tablero fila columna

devuelve: La fila máxima donde se puede poner una ficha en la columna indicada

busca\_fila :: [[Int]] -> Int -> Int -> Int

Función que genera las coordenadas donde el jugador contrario puede poner una ficha

Se le pasa las dimensiones del tablero y el tablero

parámetros: n m tablero

devuelve: Las coordenadas del tablero donde el contrario puede poner una ficha

posiciones\_contrario :: Int -> Int -> [[Int]] -> [(Int, Int)]

Función que devuelve una lista con las coordenadas donde el contrario contrario hace 4 en raya

Si en esta lista hay 2 o más elementos el ordenador pierde

Se le pasa las dimensiones del tablero y el tablero

parámetros: n m tablero

devuelve: Una lista de coordenadas

comprueba\_posiciones :: Int -> Int -> [[Int]] -> [(Int, Int)]

Función para empezar la partida

Se le pasa las dimensiones del tablero el turno y un tercer parámetros que si es 3 indicará que se juega contra el ordenador, si es 2 indicará que los jugadores son humanos

En esta función se piden por pantalla los datos necesarios al jugador para realizar una jugada

parámetros: tablero n m turno 3

devuelve: El turno del jugador que gana

partida :: [[Int]] -> Int -> Int -> Int -> Int -> IO (Int)

Función que comprueba si se ha producido un empate

El empate se produce cuando en la primera fila del tablero (tablero!!0) no hay ningún 0, es decir, no hay hueco para poner una ficha

parámetros: tablero

devuelve: Falso si hay un 0 en la primera fila del tablero, y verdadero si no lo hay

comprueba\_empate :: [[Int]] -> Bool

Función que cambia el turno del jugador

parámetros: turno

devuelve: 1 o 2

cambia\_turno :: Int -> Int

Lista con las coordenadas para comprobar el cuatro en raya

(0,1) -> Horizontal

(1,0) -> Vertical

(1,2) -> Diagonal principal

(-1,1)-> Diagonal secundaria

direccion=[(0,1),(1,0),(1,2),(-1,1)]

Función que genera las coordenadas consecutivas del tablero donde el valor corresponde con el turno

Se le pasa el tablero, las dimensiones del tablero, el turno, la coordenada donde se ha puesto la ficha (x0,y0)

La lista s va desde -3 hasta 3 para que genere la lista de todas las direcciones a partir de la coordenada (x0,y0)

parámetros: tablero n m turno x0 y0

devuelve: Una lista de quintuplas en la que los dos primeros componentes son coordenadas, los dos siguientes es la dirección y el último es el valor de la ficha en las coordenadas que corresponderá con el turno que se le pasa como parámetro

genera\_coordenadas :: [[Int]] -> Int -> Int -> Int -> Int -> Int -> [(Int, Int, Int, Int, Int)]

Funciones que generan las coordenadas donde hay fichas consecutivas (en horizontal, vertical o las dos diagonales) que corresponden al turno

Se le pasa el tablero, las dimensiones del tablero, el turno, la coordenada donde se ha puesto la ficha (x0,y0) y la dirección (ix,iy)

parámetros: tablero n m turno x0 y0 ix iy

devuelve: Una lista de quintuplas en la que los dos primeros componentes son coordenadas, los dos siguientes es la dirección y el último es el valor de la ficha en las coordenadas

coge\_horizontales :: [[Int]] -> Int -> Int -> Int -> Int -> Int -> Int -> [Int]

coge\_verticales :: [[Int]] -> Int -> Int -> Int -> Int -> Int -> Int -> [Int]

coge\_diagonal :: [[Int]] -> Int -> Int -> Int -> Int -> Int -> Int -> [Int]

coge\_otra\_diagonal :: [[Int]] -> Int -> Int -> Int -> Int -> Int -> Int -> [Int]

Función que comprueba si se ha hecho cuatro en raya, para que sea cuatro en raya la lista que devuelve debe tener 4 o más elementos y que todos sean igual al turno que se le pasa como parámetro

Se le pasa el tablero, las dimensiones el turno y la coordenada donde se ha puesto la ficha

parámetros: tablero n m turno x0 y0

devuelve: True si se ha hecho 4 en raya o False si no se ha hecho 4 en raya

hay\_cuatro\_en\_raya :: [[Int]] -> Int -> Int -> Int -> Int -> Bool

Función que comprueba que todos los elementos de una lista son iguales al turno  
Se le pasa el turno y la lista que se quiere comprobar  
parámetros turno lista  
devuelve: True si todos los elementos de la lista son igual a turno y Falso si no lo son  
todos\_iguales :: Int -> [Int] -> Bool

Función principal en la que se reciben los parámetros, se crea el tablero y se elige el modo de juego según la longitud de los argumentos recibidos  
Muestra el jugador que gana la partida  
main = do

...  
...  
...