

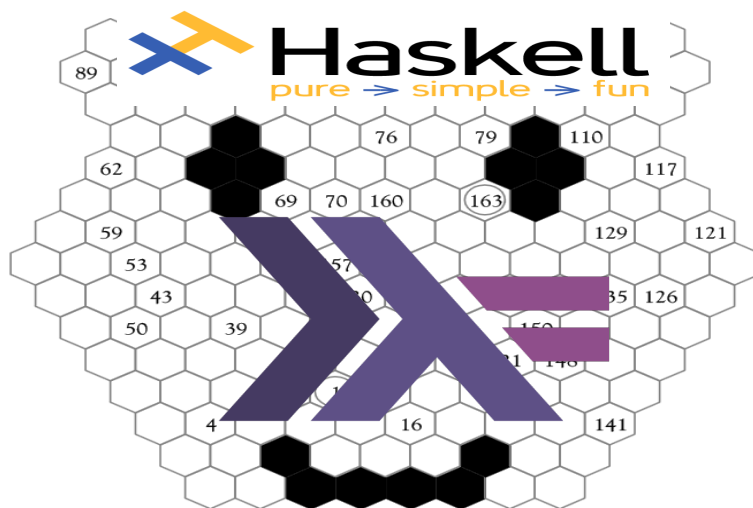
INFORME DEL PROYECTO II DE PROGRAMACION DECLARATIVA

Dianelys Cruz, Jordan Pla (C-311)

Universidad de La Habana

HIDATO - HASKELL

Hidato es un juego de lógica creado por el Dr. Gyora Benedek, un matemático israelí. El objetivo de Hidato es rellenar el tablero con números consecutivos que se conectan horizontal, vertical o diagonalmente.



Haskell

Haskell es un lenguaje de programación estandarizado multipropósito, funcionalmente puro, con evaluación no estricta y memorizada, y fuerte tipicación estática. Su nombre se debe al lógico estadounidense Haskell Curry, debido a su

aportación al cálculo lambda, el cual tiene gran influencia en el lenguaje.

Algoritmo de Solución

Se representa el tablero como un tipo compuesto de 3 elementos: una lista compuesta por listas para representar las casillas, la posición del menor número del tablero y la posición del mayor representados por tuplas. Las casillas están representadas con números enteros. La definición del tablero es:

```
data Tablero = Tablero {  
    matriz :: [[Int]],  
    minPos :: (Int,Int),  
    maxPos :: (Int,Int)  
} | Null deriving (Show, Eq)
```

Donde Null es un valor especial que se usa para representar cuando un tablero es inválido. Se tiene el conocimiento de que existen otras estructuras en Haskell para representar la ausencia de valor, pero se decidió modificar la definición Tablero con el objetivo de ganar en simplicidad y flexibilidad con los métodos definidos.

Asumimos que los números usados en el juego son naturales mayores o igual que 1. Por ello se usa el 0 para representar las casillas que están vacías y se desean completar y como un tablero no es necesariamente rectangular se usa además -1 para representar las casillas en las que no se pueden poner ningún número (obstáculos).

La función principal para dar solución a un Hidato se llama **soluciona** y recibe como parámetro un **Tablero** y como salida imprime el tablero solucionado.

Algoritmo de Generación

La función principal para generar un Hidato se llama **genera** y recibe como parámetro dos números **n** y **m** que serán las dimensiones del tablero. Como salida se devuelve un Tablero.

Se representan distintas formas del tablero: cascadaTab, cruzTab, estandarTab, TriangularSupTab, TriangularInfTab.

Cada vez que se genera un tablero puede ser de cualquiera de estas formas aleatoriamente.

Manual para usuarios

Para ejecutar el programa se ejecuta las siguientes instrucciones:

```
ghci>tablero <- genera n m
```

```
ghci>printTablero tablero
```

```
ghci>soluciona tablero
```