

1. Modificar directorio

Tendrás que modificar el directorio al tuyo personal en la línea 10 del fichero *es.hs*. Donde pone *main_path = texto* poner el tuyo.

```
9 main_path :: String
10 main_path = "D:/Haskell/Sudoku/"
```

2. Cómo jugar

1. Ejecutar *main.hs* en la terminal y llamar a la función *main* para empezar.
2. Las casillas fijas del sudoku vienen en negro y las modificadas por el usuario en azul.
3. Tipos de movimientos. Todos se definen por sí mismos. Comentar que en *Introducir posibilidades* se pueden introducir varios valores a la vez (separados por comas, sin ningún punto) o uno solo. Ejemplo:
 - Anadir nuevos: 1,2,3,4,5
 - Anadir nuevos: 7

Para ver los valores tendrás que seleccionar *Ver posibilidades* y saldrán los que vaya añadiendo el usuario a lo largo de la partida.

4. Guardar partida: primero salir y luego te preguntará si quieres guardar o no.
5. Cuando te dan varias opciones a elegir, si introduces un valor incorrecto normalmente se irá por la última opción, exceptuando los casos en los que sea necesario un valor exacto y ahí sí te volverá a pedir el valor. En las respuestas de sí o no (y/n), entenderá como *sí* a la respuesta *y*, y como *no* a todas las demás.

3. Esquema

Hay tres opciones principales:

1. Jugar: abrir un sudoku (eliges el nivel) y empezar a completarlo.
2. Resolver: le indicas un nivel de sudoku (son los mismos que accedes en jugar) y te da la opción de resolverlo de dos formas:
 - prueba y error
 - pensando

Generalmente *pensando* es más rápido (se aprecia gran diferencia en los últimos sudokus). Adicionalmente he añadido uno, el último (difícil → múltiple solución), que tiene dos soluciones. Resolverlo con:

- prueba y error: te devuelve solo una solución.
 - pensando: te devuelve todas las soluciones (dos en este caso).
3. Partida Guardada: reanudar partida guardada con anterioridad, como ya se mencionó en el punto (1). Opción válida solo cuando tengas partidas guardadas.

4. Resumen

Mencionar que todas las funciones vienen comentadas en los scripts (.hs) por lo que aquí comentaré brevemente la idea del código.

Para el objeto tablero hemos definido dos tipos de datos:

1. Casilla: representa cada una de las casillas del tablero, puede ser un número
 - V : valor modificable
 - VN : valor NO modificable

o una lista de números temporales, Temp, en donde se apunta las posibilidades de esa casilla.

2. Tablero: una matriz con 9x9 casillas (el Sudoku).

Para resolver este tablero hemos definido dos funciones principales:

1. prueba y error: de forma bruta y número a número comprueba todas las posibles opciones hasta solucionar el sudoku. para introducir un número antes hacer una pequeña comprobación de si el número ahí es válido o no, así si hemos llegado a la última casilla y podemos rellenarla, podemos asegurar que lo hemos completado.
2. pensando: se definen funciones para resolver el tablero de una forma lógica.

5. Distribución de archivos

1. funciones.hs : están las principales funciones necesarias para jugar al sudoku. Saber si es posible meter un número en tal casilla (comprobar filas, columnas y el bloque), devolver el valor de la casilla (i,j), hacer el print por pantalla, los tipos de datos, ...
2. prueba_y_error.hs : funciones extra para resolver el sudoku de la forma *prueba y error*. Importamos el fichero *funciones.hs*.
3. pensando.hs : funciones extra para resolver el sudoku de la forma *pensando*. Importaremos el fichero *funciones.hs*.
4. es.hs : funciones de entrada salida. Según las respuestas del usuario llamará a una función u otra.
5. main.hs : fichero final a cargar en la terminal. Hace la llamada principal para preguntar qué modo elegir (uno de los tres del principio) y luego se ayuda de *es.hs* para hacer toda la entrada / salida restante.
6. Tableros : carpeta para guardar los tableros en ficheros (.txt):
 - predeterminados: se guardan todos los sudokus que vienen por defecto.
 - guardados: se guardan (con el nombre puesto por el usuario) las partidas sin terminar que halla guardado el usuario.

Los predeterminados se han guardado de una forma amigable al lector, para poder introducirlos fácilmente (por el contrario necesitan de una función auxiliar de lectura) y los guardados con el propio *show* y *read*, guardamos y cargamos.