Práctica 3
Sudoku
数独

Fecha de entrega: 29 de marzo de 2012

Objetivos:

- **La abstracción procedimental: estructuración y reutilización.**
- **Tipos estructurados. Diseño descendente.**

Aplicación:

- **Juego del Sudoku con ayuda al usuario.**

1. Objetivo de la práctica

El Sudoku (en japonés, 数独) es un rompecabezas matemático de colocación que se hizo popular en Japón a finales del siglo XX y que viene causando furor en todo el mundo desde hace unos años.

Su resolución requiere paciencia y ciertas dotes lógicas, y tradicionalmente consiste en rellenar un tablero de 9×9 casillas, es decir 81 casillas, dividido en regiones o cajas de 3×3 con las cifras del 1 al 9 partiendo de algunos números ya dispuestos en algunas de las casillas. La única regla que hay que respetar a la hora de resolver un Sudoku es que no se repita ninguna cifra en una misma **fila**, **columna** o **región**. Suena sencillo, ¿verdad? Pero no te engañes, pues el grado de dificultad puede llegar a ser diabólico.

La Figura 1 muestra un Sudoku de nivel de dificultad bajo en su estado inicial. La Figura 2 muestra su solución.

		3			1	7	2	
					4			
			7			1	4	9
	1	4	8					5
2	8						7	4
7					2	6	8	
9	5	2			8			
			3					
	6	7	9			4		

Fig. 1. Tablero de partida.

4	9	3	5	8	1	7	2	6
1	7	6	2	9	4	8	5	3
5	2	8	7	3	6	1	4	9
6	1	4	8	7	9	2	3	5
2	8	5	6	1	3	9	7	4
7	3	9	4	5	2	6	8	1
9	5	2	1	4	8	3	6	7
8	4	1	3	6	7	5	9	2
3	6	7	9	2	5	4	1	8

Fig. 2. Tablero resuelto.

1.1. Consideraciones generales sobre la práctica

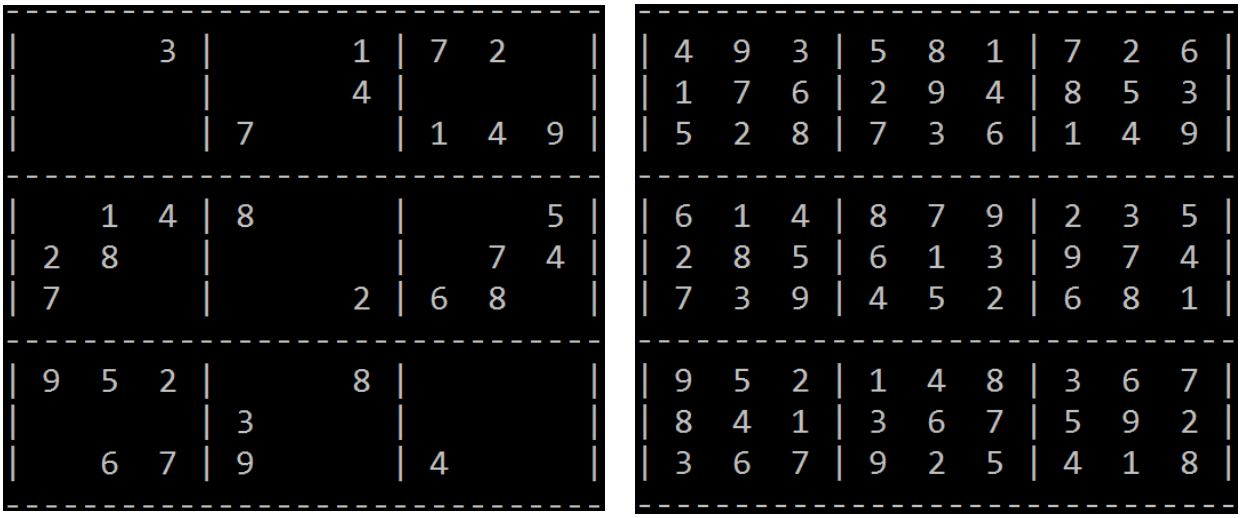
En esta práctica, además de conseguir que funcione adecuadamente, es necesario que tengas en cuenta las siguientes consideraciones mientras la implementes:

- ✓ Utiliza la abstracción procedimental adecuadamente. Es decir, crea las funciones oportunas para no repetir (demasiadas) partes de tu código. Sigue las consideraciones de la Práctica 2, en las que se te indica que crees funciones para cada opción del menú, para pedir los datos al usuario, para realizar el correspondiente proceso y para mostrar el resultado. Además, crea funciones para la carga de datos desde disco y para los distintos tipos de datos. Todo ello, aplicando el diseño descendente.
- ✓ Define los tipos utilizando constantes para los tamaños.
- ✓ Los parámetros de entrada de tipo estructurado es más eficiente pasarlos como constantes por referencia.
- ✓ No utilices variables globales, ni la función `exit()`o la función `system()`.
- ✓ Dentro de las instrucciones de bifurcación condicional o iteración no debes utilizar las instrucciones `continue`, `break` o `return`.
- ✓ Cada función debe tener un único punto de salida (un solo `return` al final).

Si tienes en cuenta todas estas consideraciones, además de crear un programa que funcione, crearás un programa bien estructurado y fácil de entender.

1.2. Visualización del Sudoku

El Sudoku se mostrará en la consola con el siguiente aspecto:



2. Descripción de la práctica

El objetivo de esta práctica es desarrollar una aplicación que permita al usuario jugar al Sudoku. El estado del Sudoku se mostrará en la consola, junto con algunos mensajes informativos.

Ten en cuenta que todas las opciones que signifiquen acciones sobre una casilla concreta de tablero tendrán que solicitar al usuario la fila y la columna de dicha casilla a través de la consola. Hay que solicitar dos enteros en el intervalo [1..9], ambos en la misma línea y separados por algún espacio.

La práctica comenzará solicitando el nombre de dos archivos (*Figura 3*):

- ✓ **Archivo con el estado inicial del Sudoku**. Este archivo contendrá el estado inicial del tablero del Sudoku (*Figura 1*). El archivo contiene una fila por cada casilla del Sudoku que tiene un valor inicial y cada fila contiene 3 enteros con valores en el intervalo [1..9], representando (en este orden) la fila, la columna y el valor de la casilla. El archivo termina con un -1 como fila. Por defecto el nombre será `sudoku1.txt`.
- ✓ **Archivo que contiene la solución del Sudoku**. Para alguna de las opciones es necesario conocer la solución del Sudoku planteado (*Figura 2*). Este archivo contiene una fila por cada fila del Sudoku (9 filas) y cada fila contiene 9 enteros, en el intervalo [1..9], que representan cada uno de los valores de las casilla de cada una de las filas del Sudoku. Se puede dar por sentado que este archivo no contiene errores. Por defecto el nombre será `solsdk1.txt`.

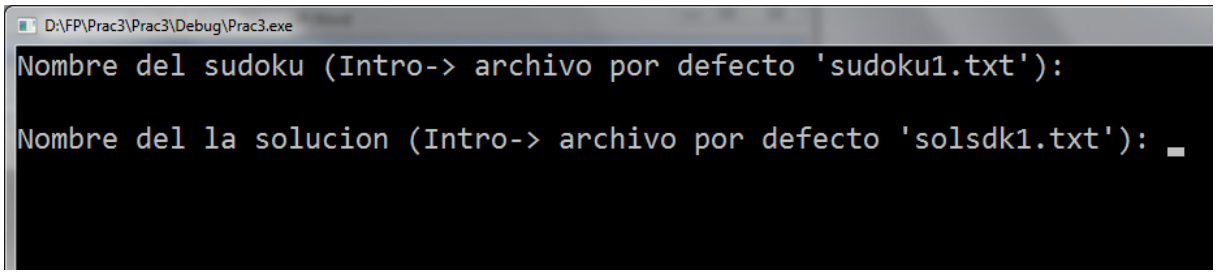


Fig. 3. Solicitud de los nombres de los archivos del estado inicial y de la solución.

Una vez solicitados los nombres de los archivos, tanto del estado inicial como de la solución, deben ser cargados en sendas estructuras de datos que debes definir. Si existe algún fallo durante la carga de una de las estructuras, el programa debe mostrar un mensaje de error en la consola y terminar la ejecución de la aplicación.

Algunos posibles errores son:

- ✓ Algún dato está fuera del intervalo [1..9].
- ✓ El número de datos es superior al máximo posible. Por ejemplo, que en el archivo con el estado inicial haya más filas que casillas tiene el Sudoku (9×9).

Una vez que se hayan cargado los archivos en las estructuras de datos, se mostrará el Sudoku y comenzará el bucle principal de la aplicación con un menú que debe contener (en este orden) las siguientes opciones:

```
0.- SALIR
1.- Ver posibles valores de casilla
2.- Colocar un valor en una casilla
3.- Borrar el valor de una casilla
4.- Mostrar valores incorrectos
5.- Reiniciar tablero
6.- Completar casillas simples
```

La aplicación terminará si el usuario selecciona la opción 0 o si se llena el tablero del Sudoku. En este caso, existen dos finales posibles:

1. El Sudoku ha sido completado correctamente. En este caso, se muestra en la consola el mensaje ENHORABUENA HAS CONSEGUIDO SOLUCIONARLO (*Figura 4*).
2. El Sudoku no ha sido completado correctamente. En este caso, el Sudoku se ha completado, pero no coincide con la solución. Se debe implementar el mismo comportamiento que para la opción 4 de la práctica.

En cualquier caso, la aplicación termina mostrando un mensaje por la consola en espera de que el usuario pulse Intro para terminar.

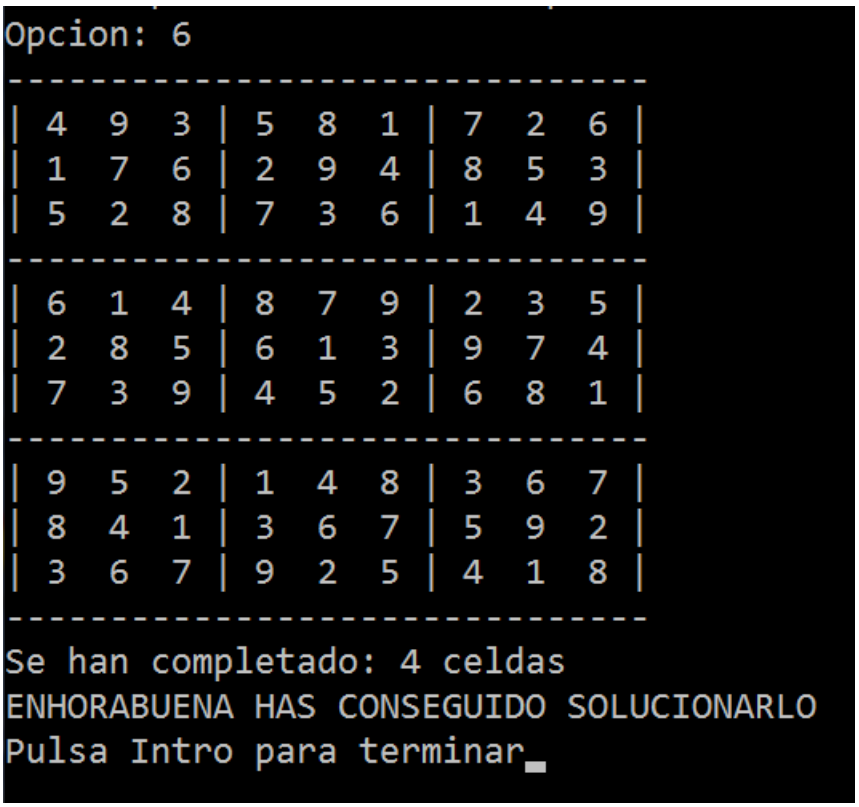


Fig. 4. Fin del programa cuando se resuelve el Sudoku.

2.0. Opción 0 (*Salir*)

La aplicación termina.

2.1. Opción 1 (*Ver posibles valores de casilla*)

Solicita al usuario la casilla (fila y columna separados por algún espacio) y, si está vacía, muestra todos los dígitos válidos (que cumplen las reglas del juego), separados por un espacio, con los que podría rellenarse esa casilla (*Figura 5*). Si la casilla ya tiene un valor se mostrará el mensaje CASILLA NO VACIA.

```
0.- SALIR
1.- Ver posibles valores de casilla
2.- Colocar un valor en una casilla
3.- Borrar el valor de una casilla
4.- Mostrar valores incorrectos
5.- Reiniciar tablero
6.- Completar casillas simples
Opcion: 1
Introduce fila y columna [1..9]: 1 1
4 5 6 8
```

Fig. 5. Opción 1.

2.2. Opción 2 (*Colocar un valor en una casilla*)

Solicita al usuario la casilla y el valor (*Figura 6*) y coloca el valor en la casilla, si se puede (la casilla está vacía y el valor cumple las reglas del juego).

```
Opcion: 2
Introduce fila y columna [1..9]: 1 1
Introduce un valor en el intervalo [1..9]: 8
-------------------------------
| 8     3 |       1 | 7  2    |
|         |       4 |         |
|         | 7       | 1  4  9 |
-------------------------------
|    1  4 | 8       |       5 |
| 2  8    |         |    7  4 |
| 7       |       2 | 6  8    |
-------------------------------
| 9  5  2 |       8 |         |
|         | 3       |         |
|    6  7 | 9       | 4       |
-------------------------------
```

Fig. 6. Opción 2.

- ✓ Si se puede colocar el valor, se mostrará el Sudoku con ese nuevo valor (*Figura 5*).
- ✓ Si no se puede colocar el valor se mostrará un mensaje informando al usuario acerca del error:
  - ✓ Si la casilla ya tiene un valor, se mostrará el mensaje CASILLA NO VACIA.
  - ✓ Si la casilla es una inicial, se mostrará el mensaje NO MODIFICABLE.
  - ✓ Si el dígito a colocar no cumple con las restricciones del Sudoku, se mostrará el mensaje DIGITO NO VALIDO.

2.3. Opción 3 (*Borrar el valor de una casilla*)

Solicita al usuario la casilla y vacía esa casilla. Evidentemente sólo se pueden vaciar casillas rellenas que no sean valores iniciales del juego. Si todo va bien, mostrará el Sudoku tras haber eliminado el valor.

Si no se puede borrar la casilla se mostrará un mensaje informando al usuario acerca del error tanto en la consola como en la ventana gráfica:

- ✓ Si la casilla está vacía, se mostrará el mensaje CASILLA VACIA.
- ✓ Si la casilla es una casilla inicial, se mostrará el mensaje NO MODIFICABLE.

2.4. Opción 4 (*Mostrar valores incorrectos*)

Muestra todos los valores colocados que sean incorrectos. Para ello, inspecciona el tablero actual, contrastándolo con la solución. Los errores deben notificarse en la consola. Para cada error, una línea con los valores de fila y columna entre corchetes y el valor incorrecto entre paréntesis:

```
Opcion: 4
Errores:
[1,1] (8)
[1,2] (4)
```

Si no hay errores, se mostrará el mensaje NO HAY ERRORES.

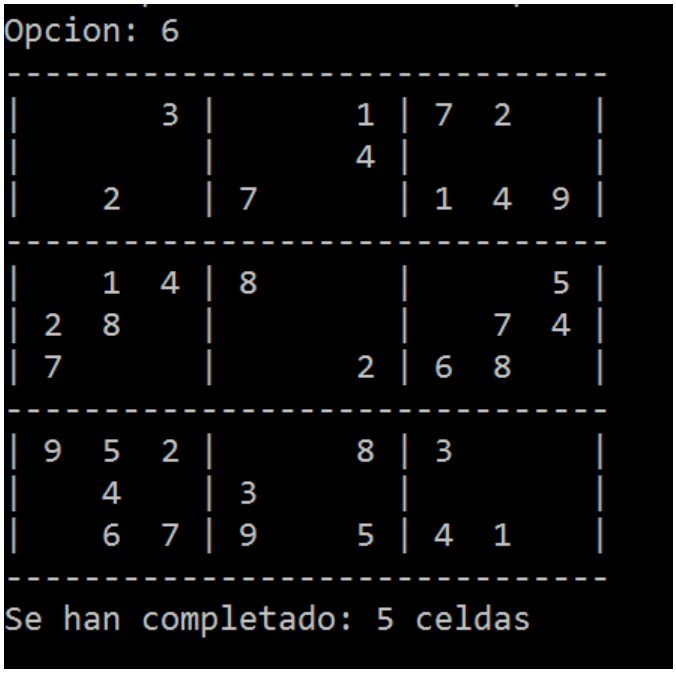
2.5. Opción 5 (*Reiniciar tablero*)

Modifica el Sudoku para que quede en el mismo estado que al comenzar la partida. Una vez modificado el Sudoku lo vuelve a dibujar.

2.6. Opción 6 (*Completar casillas simples*)

Esta opción coloca automáticamente los valores de las casillas que estando vacías tienen un único valor válido y que, por tanto, puede considerarse definitivo para el estado actual del tablero.

Los dígitos que se añaden automáticamente al Sudoku, mostrando éste en su nuevo estado. Además, se mostrará el mensaje COMPLETADAS: *XX*, donde *XX* es el número de casillas que han sido completadas automáticamente:



3. Cuestiones de implementación

Todas las entradas del usuario se deben validar, no admitiendo el avance de la ejecución hasta que sean válidas.

La información acerca del tablero de partida y su solución se encontrarán en sendos archivos de texto. El formato de los archivos está descrito al comenzó de la Sección 1.2 de este enunciado y, además, puedes ver el contenido de los archivos del Sudoku del ejemplo en los archivos: `sudoku1.txt` y `solsdk1.txt`, para el tablero inicial y la solución, respectivamente. Si el usuario no introduce un nombre de archivo (pulsa Intro), se tomará cada vez esos nombres como valores por defecto.

El reinicio del tablero (en caso de ser solicitado) debe realizarse sin nuevos accesos a disco.

4. Entrega de la práctica

Una vez probada la práctica, la entrega de la misma la realizarás a través del CV. En el CV se habilitará una nueva actividad **Entrega de la Practica 3** que te permitirá subir un archivo con el contenido de la práctica.

Utilizando el CV subirás el archivo `sudoku.cpp`.

El sistema de entrega estará abierto hasta las **23:55 del día de entrega** (indicado al comienzo del enunciado de la práctica).