

P3

Práctica 3 Explicación del Sudoku

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores



Luis Hernández Yáñez
Pablo Moreno Ger
Facultad de Informática
Universidad Complutense



Práctica 3: Juego del Sudoku

数独

Especificación

El programa permitirá al usuario gestionar un Sudoku.

El usuario podrá:

- ✓ Crear un juego cargando los datos de archivo.
- ✓ Ver los posibles valores de una casilla.
- ✓ Colocar un valor en una casilla.
- ✓ Borrar el valor de una casilla.
- ✓ Mostrar los valores incorrectos.
- ✓ Reiniciar el tablero.
- ✓ Completar las casillas deducibles.

El programa deberá mostrar repetidamente un menú con la lista de opciones, más otra que permita salir del programa.



Práctica 3: Juego del Sudoku

数独

¿Qué debe hacer el programa?



Práctica 3: Juego del Sudoku

数独

Diseño

El programa principal necesitará mostrar un menú con siete 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

Para cada opción el programa principal deberá realizar la tarea solicitada, apoyándose en otros subprogramas desarrollados al efecto.



Práctica 3: Juego del Sudoku

数独

El tablero



Práctica 3: Juego del Sudoku

数独

Resolución del Sudoku

En cada casilla un número entre 1 y 9.

No se repiten números en cada fila, columna y región.

Inicialmente se proporcionan varias casillas con sus valores adecuados:

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

Sudoku con los valores iniciales

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

Sudoku resuelto



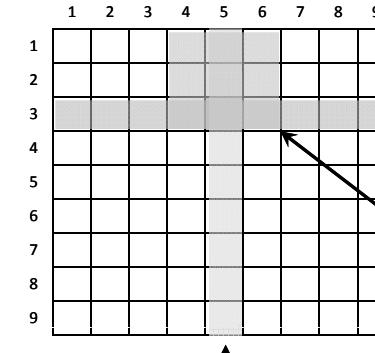
Práctica 3: Juego del Sudoku

数独

El tablero del Sudoku

Tablero de 9 × 9 casillas con regiones de 3 × 3.

Cada casilla está en una fila, una columna y una región.



Regiones de 3 × 3
Fila 3
Región 2
Columna 5

1	2	3
4	5	6
7	8	9



Práctica 3: Juego del Sudoku

数独

Estructuras de datos



Práctica 3: Juego del Sudoku

数独

Estructuras de datos para el tablero del Sudoku

Array bidimensional de 9 x 9 enteros

- ✓ Debemos saber qué casillas están vacías → Usaremos el valor 0.
- ✓ Las casillas con los valores iniciales no se pueden modificar.
Hay que tenerlas identificadas → Array bidimensional de 9 x 9 booleanos.

	0	1	2	3	4	5	6	7	8
0	0	0	3	0	0	1	7	2	0
1	0	0	0	0	0	4	0	0	0
2	0	0	0	7	0	0	1	4	9
3	0	1	4	8	0	0	0	0	5
4	2	8	0	0	0	0	0	7	4
5	7	0	0	0	0	2	6	8	0
6	9	5	2	0	0	8	0	0	0
7	0	0	0	3	0	0	0	0	0
8	0	6	7	9	0	0	4	0	0

¡Cuidado! Los arrays van de 0 a 8



	0	1	2	3	4	5	6	7	8
0	F	F	T	F	F	T	T	T	F
1	F	F	F	F	F	T	F	F	F
2	F	F	F	T	F	F	T	T	T
3	F	T	T	T	F	F	F	F	T
4	T	T	F	F	F	F	F	T	T
5	T	F	F	F	F	T	T	T	F
6	T	T	T	F	F	T	F	F	F
7	F	F	F	T	F	F	F	F	F
8	F	T	T	T	F	F	T	F	F

T = true
F = false



Práctica 3: Juego del Sudoku

数独

Estructuras de datos para el tablero del Sudoku

El Sudoku se habrá resuelto cuando ya no queden casillas vacías y todos los valores sean correctos.

¿Cómo podemos saber cuántas casillas vacías quedan?

- ✓ Podemos calcularlo con un recorrido, mirando cuántas casillas siguen a cero.
 - ✓ Poco eficiente
- ✓ Contador que se decrementa al poner cada valor.
 - ✓ Más eficiente
 - ✓ Un poco más complicado
 - ✓ **Elegimos esta opción.**



Práctica 3: Juego del Sudoku

数独

Estructuras de datos para guardar el estado actual

Estructura de datos tTablero

struct con los siguientes campos:

- ✓ Contador de casillas vacías.
- ✓ Array bidimensional de enteros (0 para las casillas vacías).
- ✓ Array bidimensional de booleanos (true para las casillas con los valores iniciales).



Práctica 3: Juego del Sudoku

数独

¿Y la solución?

Array bidimensional de enteros adicional, todas las casillas rellenas.

¿Lo ponemos dentro de la estructura anterior?

Mejor no:

- ✓ La solución sólo la usaremos para ver si ha terminado la partida.
- ✓ No tiene sentido y pasándosela a todos los subprogramas.



Inicialización del juego



Práctica 3: Juego del Sudoku

Inicialización del juego

Archivo con los valores iniciales.-

- ✓ En cada línea tres enteros: fila, columna y valor
- ✓ Todos los datos han de estar en el intervalo 1..9
- ✓ Línea final con un -1 (centinela)

Posibles errores:

- ✓ No existe el archivo
- ✓ Algún dato no está en el intervalo 1..9

Archivo por defecto: **sudoku1.txt**



Si una posición no aparece en el fichero:
* No es una casilla inicial
* Su valor inicial es cero

1 3 3
1 6 1
1 7 7
1 0 2
2 6 1
3 4 7
3 7 1
3 8 1
3 9 9
4 2 1
4 3 4
4 4 8
4 9 5
5 1 2
5 2 8
5 H 7
5 9 4
6 1 7
6 6 2
6 7 6
6 0 0
7 1 4
7 2 5
7 3 2
7 6 8
8 4 3
9 2 6
9 3 7
9 4 0
9 7 1
1

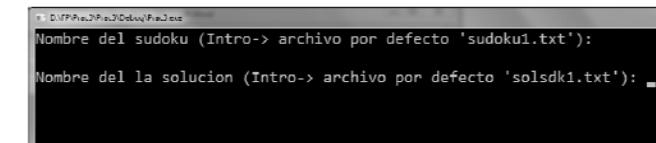


Práctica 3: Juego del Sudoku

Inicialización del juego

La función `main()` comenzará inicializando las estructuras de datos del Sudoku y pedirá dos cadenas de caracteres:

- ✓ Nombre del archivo que contiene los valores iniciales.
- ✓ Nombre del archivo que contiene la solución.



A continuación cargará los valores iniciales, detectando posibles errores.

Si no ha habido errores, cargará la solución (sin errores).



Práctica 3: Juego del Sudoku

Inicialización del juego

Práctica 3: Juego del Sudoku

Inicialización del juego

Archivo con la solución.-

- ✓ 9 filas
- ✓ En cada fila 9 valores (las 9 columnas de esa fila)
- ✓ Todos los datos estarán en el intervalo 1..9

No hay que comprobar errores al leer este archivo.

4 4 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 0 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 0

Se puede dar por sentado que es correcto.

Possible error:

- ✓ No existe el archivo

Archivo por defecto: **solsdk1.txt**



Práctica 3: Juego del Sudoku

数独

La función main()

```
Declaración de variables  
Llamar a la función de inicialización  
Pedir el nombre del archivo de valores iniciales  
Pedir el nombre del archivo con la solución  
Llamar a la función de carga de valores iniciales  
Si ha habido algún error  
    Notificar el error  
Si no  
    Llamar a la función de carga de la solución  
    Repetir  
        Dibujar el Sudoku (estado actual del tablero)  
        Mostrar el menú y obtener la opción (válida) elegida  
        Llamar a la función correspondiente a la opción elegida  
    Hasta que la opción sea la 0 (Salir) o se haya resuelto el Sudoku  
    Felicitar si se ha resuelto  
Pedir al usuario que pulse Intro y terminar
```



Práctica 3: Juego del Sudoku

数独

Primeros pasos

Consigue que el programa inicialice el tablero, lea el estado inicial y la solución de los archivos de entrada, dibuje el tablero y muestre el menú repetidamente hasta que se elija la opción 0.

De momento, no hará nada cuando se elijan las opciones 1 a 6.

Algunos consejos

- ✓ Define las estructuras de datos siempre como tipos.
- ✓ Declara constantes para las dimensiones de los arrays.
- ✓ No te olvides de los prototipos.
- ✓ Declara las variables globales en `main()`, no fuera de las funciones.
- ✓ No redeclares datos que ya están definidos en las bibliotecas: ¡úsalos!
- ✓ Recuerda que debes validar cualquier entrada del usuario.
- ✓ Y que para el usuario las filas y columnas van de 1 a 9.



Práctica 3: Juego del Sudoku

数独

Primeros pasos



Práctica 3: Juego del Sudoku

数独

Primeros subprogramas

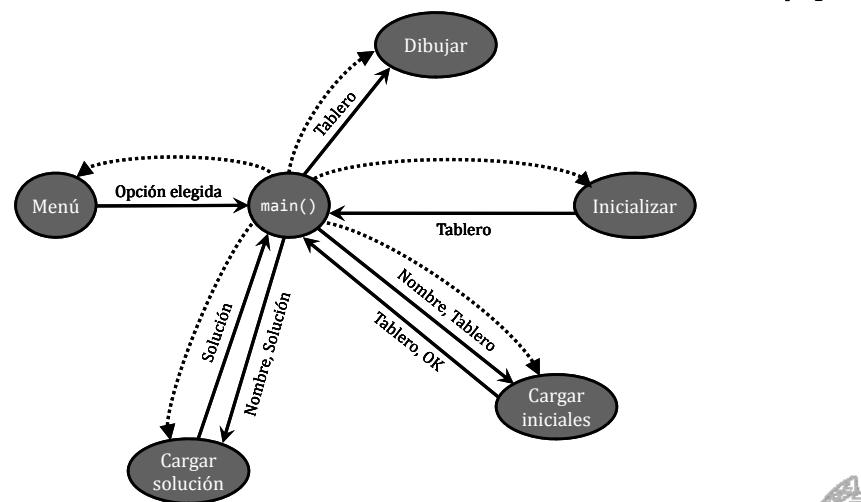
- ✓ Menú
Mostrará el menú y pedirá repetidamente al usuario que elija una opción hasta que ésta sea válida; devolverá la opción elegida.
- ✓ Inicialización
Devolverá la estructura de datos inicializada.
- ✓ Carga de valores iniciales
Aceptará el nombre del archivo con los valores iniciales (`string`) y devolverá el tablero actualizado y un valor `bool` que indicará si la carga se ha podido realizar con éxito (`true`) o ha habido algún error (`false`).
- ✓ Carga de la solución
Aceptará el nombre del archivo con la solución(`string`) y devolverá el array solución inicializado.
- ✓ Dibujo del Sudoku
Acepta el tablero y lo dibuja en la consola.



Práctica 3: Juego del Sudoku

数独

Primeros subprogramas



Luis Hernández Yáñez / Pablo Moreno Ger



Fundamentos de la programación: Práctica 3

Página 20



Práctica 3: Juego del Sudoku

数独

Opción 1

Luis Hernández Yáñez / Pablo Moreno Ger



Fundamentos de la programación: Práctica 3

Página 22



Práctica 3: Juego del Sudoku

数独

Dibujo del tablero en la consola

Simplemente se colocan los dígitos del array bidimensional encerrados en sus zonas.

Cada número ocupando 3 posiciones (un espacio antes y otro después).

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			

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

Luis Hernández Yáñez / Pablo Moreno Ger



Fundamentos de la programación: Práctica 3

Página 21



Práctica 3: Juego del Sudoku

Opción 1

Ver posibles valores de casilla

Comienza pidiendo la fila y la columna de la casilla.

A continuación comprueba si la casilla está vacía:

No vacía: mensaje "CASILLA NO VACIA"

Vacía: Calcula la lista de candidatos y los muestra separados por un espacio

Lista de candidatos:

Necesitamos una estructura en la que guardar listas de dígitos (enteros de 1 a 9)

→ **Array con contador de elementos**

Un dígito X es candidato si:

- ✓ No está en esa fila
- ✓ No está en esa columna
- ✓ No está en la región de esa casilla

```
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
```

Luis Hernández Yáñez / Pablo Moreno Ger



Fundamentos de la programación: Práctica 3

Página 23



Práctica 3: Juego del Sudoku

Opción 1

Más subprogramas

Algunos se utilizarán también en otras opciones.

- ✓ Solicitud de fila y columna de casilla: debe asegurarse de que estén entre 1 y 9 (volver a pedir).
- ✓ ¿Casilla vacía? Dados el juego, la fila y la columna, devuelve bool.
- ✓ Cálculo de los candidatos: dados el juego, una fila y una columna, devuelve la lista de candidatos (dígitos) para esa casilla.
Para cada dígito (1 a 9) comprueba si está en la fila, columna o región.
Si no está, lo añade a la lista de candidatos.
- ✓ ¿Está en la fila? Dados el juego, una fila y un dígito, devuelve bool.
- ✓ ¿Está en la columna? Dados el juego, una columna y un dígito, devuelve bool.
- ✓ ¿Está en la región? Dados el juego, una fila, una columna y un dígito, devuelve bool.
- ✓ Subprograma que implementa la Opción 1 (usará los anteriores).



Práctica 3: Juego del Sudoku



Opción 2



Práctica 3: Juego del Sudoku

Opción 1

Ver posibles valores de casilla

¿Qué casillas están en la región de una dada?

Pasemos a índices de array: $f = \text{fila} - 1$ y $c = \text{columna} - 1$

	c								
0	0	1	2	3	4	5	6	7	8
1									
2									
3									
4									
5									
6					6,4				
7									
8									



$f/3$ nos da 0, 1 o 2 [Ej.: $6/3=2$] Div. entera!

$(f/3)*3$ nos da 0, 3 o 6 [Ej.: $2*3=6$]

Las celdas de la región son las que tienen como fila $(f/3)*3+0$, $(f/3)*3+1$ y $(f/3)*3+2$ [Ej.: 6, 7 y 8]

Lo mismo con las columnas:

$(c/3)*3+0$, $(c/3)*3+1$ y $(c/3)*3+2$

[Ej.: $4/3=1$; $1*3=3$; 3, 4 y 5]

	3	4	5
6		6,4	
7			



Práctica 3: Juego del Sudoku

Opción 2

Colocar un valor en una casilla

Comienza pidiendo la fila y la columna de la casilla,
así como el valor a colocar.

A continuación comprueba si la casilla está vacía:

No vacía:

Casilla con valor inicial: Mensaje "NO MODIFICABLE"

Casilla con valor no inicial: Mensaje "CASILLA NO VACIA"

Vacía:

Si el valor no está en la fila, ni en la columna, ni en la región:

Coloca el dígito en la casilla y dibuja el Sudoku

En caso contrario:

Mensaje "DIGITO NO VALIDO"



Más subprogramas

Algunos se utilizarán también en otras opciones:

- ✓ Solicitud de valor (dígito): debe asegurarse de que esté entre 1 y 9. Devolverá el valor leído.
- ✓ ¿Casilla con valor inicial? Dados el juego, la fila y la columna, devuelve bool.
- ✓ Colocar el valor en la casilla: dados el juego, la fila, la columna y el valor, coloca ese valor en la celda, si es que es posible (vacía y es candidato).
- ✓ ¿Es candidato? Dados el juego, una fila, una columna y un valor, devuelve true si el valor no está en la fila, ni en la columna, ni en la región.
- ✓ Subprograma que implementa la Opción 2.

¡No hay que olvidar actualizar el contador de casillas vacías!



Resto de opciones



Borrar el valor de una casilla

Comienza pidiendo la fila y la columna de la casilla.

A continuación comprueba si la casilla está vacía:

No vacía:

Casilla con valor inicial: Mensaje "NO MODIFICABLE"

Casilla con valor no inicial:

Se quita el dígito del tablero y se vuelve a dibujar el Sudoku

Vacía: Mensaje "CASILLA VACIA"

Subprograma:

- ✓ Subprograma que implementa la Opción 3.

¡No hay que olvidar actualizar el contador de casillas vacías!



Mostrar valores incorrectos

Recorre el tablero detectando los valores que no coincidan con los de la solución.

Además:

Si no hay errores: Mensaje "NO HAY ERRORES"

Si hay errores:

Mensaje "Errores:" seguido de los valores incorrectos. Cada uno en una línea, con las coordenadas entre corchetes y el valor erróneo entre paréntesis:

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

Subprograma:

- ✓ Subprograma que implementa la Opción 4.



Práctica 3: Juego del Sudoku

Opción 5

Reiniciar tablero

Recorre el tablero quitando los valores que no sean iniciales.

Vuelve a dibujar el Sudoku.

Subprogramas:

- ✓ Reinicio del juego a sus valores iniciales: devuelve el juego modificado.
- ✓ Subprograma que implementa la Opción 5.

¡No hay que olvidar ir actualizando el contador de casillas vacías!



Práctica 3: Juego del Sudoku

Recuerda...

- ✓ Usa las constantes en los bucles para detectar el final.
- ✓ No te olvides de los prototipos.
- ✓ No declares parámetros que puedan ser variables locales.
- ✓ Declara cada parámetro de la forma adecuada:
Sólo de entrada (sin &) o de salida (con &).
- ✓ Declara las variables globales en `main()`, no fuera de las funciones.
- ✓ No redeclares datos que ya están definidos en las bibliotecas: ¡úsalos!
- ✓ Recuerda que debes validar cualquier entrada del usuario.
- ✓ Y que para el usuario las filas y columnas van de 1 a 9.



Práctica 3: Juego del Sudoku

Opción 6

Completar casillas simples

Recorre el tablero rellenando las casillas vacías que tengan un único candidato.

Para cada casilla que esté vacía:

Calcula la lista de candidatos

Si la lista contiene un solo candidato:

Coloca el dígito en el tablero

Computa una casilla más completada

Mensaje "Se han completado: X casillas",
donde X es el número de casillas completadas.

Subprogramas:

- ✓ Completar las casillas con un solo candidato: dado el juego,
completa esas casillas y devuelve el número de casillas completadas.
- ✓ Subprograma que implementa la Opción 6.



Práctica 3: Juego del Sudoku

数独

Documentación

Documenta bien el código.

Al menos deberás tener...

- ✓ Un comentario al principio del archivo con tu nombre
y una explicación general de lo que hace el programa.
- ✓ Un comentario justo antes de cada prototipo de función
que explique para qué sirve la función, así como cuáles son los datos
que acepta/devuelve la función.
- ✓ Comentarios en aquellas partes del código que necesiten algo de
explicación porque pueda no quedar claro qué es lo que se hace.



Práctica 3: Juego del Sudoku

数独

Plazo de entrega

La práctica deberá entregarse, a través del CV, como muy tarde el 29 de marzo de 2012 a las 23:55 horas.

Se deberá entregar sólo el archivo sudoku.cpp.

La práctica...

- ✓ Deberá compilar (*tened cuidado de entregar la última versión*).
- ✓ Deberá funcionar correctamente: realizar bien todas las opciones, validar los datos de entrada siempre, comunicar correctamente todas las funciones y estar bien documentado el programa.

Si no se entrega a tiempo la práctica se evaluará como No presentada.

¡Y comprobaremos los envíos ante posibles copias!



Acerca de Creative Commons



Licencia CC (Creative Commons)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

- Reconocimiento (*Attribution*): En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
- No comercial (*Non commercial*): La explotación de la obra queda limitada a usos no comerciales.
- Compartir igual (*Share alike*): La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Pulsa en la imagen de arriba a la derecha para saber más.

