



Práctica de la
Asignatura de **Programación III**
Versión 1.0

Escuela Técnica Superior de
Ingeniería Informática

UNED

Curso 2005-2006

1.- ENUNCIADO DE LA PRÁCTICA

El Su-doku es un juego muy popular entre lectores de prensa de Japón y EE.UU. y recientemente entre muchos diarios europeos.

El objetivo del juego es rellenar una matriz de 9x9 casillas con dígitos del 1 al 9. Cada Su-doku consiste en completar con dígitos una matriz parcialmente vacía, salvo algunos valores ya rellenos de manera que:

- Cada dígito aparece una y solo una vez en cada fila.
- Cada dígito aparece una y solo una vez en cada columna
- Cada dígito aparece una y solo una vez en cada uno de los nueve cuadrados de 3x3 en los que se puede subdividir la matriz inicial

Ejemplo:

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

y la solución:

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

2.- REALIZACIÓN DE LA PRÁCTICA

La práctica constará de una memoria y de un programa correctamente compilable y que resuelva el problema planteado mediante alguno de los esquemas expuestos en la asignatura.

En la memoria se detallarán las respuestas a las preguntas planteadas y un listado comentado del código fuente. Se valorará que el código esté comentado de forma que se facilite su lectura.

*****La práctica se realizará en Java*****

En caso de que el Tutor lo autorice expresamente y por escrito, el alumno podrá entregar la práctica en “Modula-2”.

2.1.- Diseño del Algoritmo

Contesta y desarrolla las siguientes preguntas:

1. Describe el tipo de problema y esquema algorítmico utilizado.
2. Expón las estrategias locales consideradas en la asignación de valores. Concrétalas en condiciones de poda.
3. Analiza el coste en términos de tratabilidad computacional.

2.2.- Funcionalidad de la práctica

El programa se deberá poder invocar **desde la línea de comandos** de Linux (ó Windows) con la siguiente sintaxis:

```
$> sudoku [-t][-a][-h] [fichero]
```

El programa leerá los datos del fichero que se le pase como argumento. Ej:

```
C:\> sudoku entrada.txt <ENTER>
```

o también desde la entrada estándar (stdin):

```
C:\> sudoku < entrada.txt <ENTER>
```

o también usando *pipe* (*tubería*)

```
C:\> type entrada.txt | sudoku <ENTER>
```

y saca por la salida estándar la matriz resuelta.

2.2.1.- Formato de la entrada y salida de datos

El fichero de entrada representará a una única matriz inicial del juego. La matriz se representa por filas de la siguiente forma:

- Si la celda contiene un dígito el fichero contendrá dicho dígito
- Si la celda contiene un blanco (no tiene dígito) el fichero tendrá el carácter ‘*’ (asterisco)

Se ignorará toda la fila restante a partir de que se encuentre el carácter ‘#’. Se ignoran los blancos y las líneas en blanco.

Por ejemplo (Linux):

```
$> cat entrada.txt <ENTER>
```

```
#-----
# Ejemplo 1.10.05
# -----
```

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

```
# -----
```

```
$>
```

Un ejemplo de ejecución a partir del fichero anterior es (Windows):

```
C:\> type entrada.txt | sudoku <ENTER>
```

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

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

C:\>

2.2.2.- Argumentos y Parámetros

La práctica se invoca usando la siguiente sintaxis:

```
sudoku [-t][-a][-h] [fichero]
```

Opciones:

-t: Realiza un test de corrección a la matriz de entrada. Si es incompleta o incorrecta según las reglas del juego devuelve¹ 1 en caso contrario devuelve 0. Sirve para comprobar errores sintácticos en la entrada.

-a: Modo traza. Muestra toda la secuencia de valores que van añadiendo a la matriz de entrada hasta completarla correctamente. Hay que tratar que sea lo más ilustrativa posible sobre el proceso del problema por parte del algoritmo

-h: Modo ayuda. Muestra la sintaxis y los créditos.

Si no tiene argumentos, el programa muestra la matriz inicial, y la matriz final resuelta.

➔ **TODOS** estos requisitos son obligatorios para aprobar la práctica.

3.- IMPLEMENTACIÓN

3.1.- Compiladores

Compilador de Java

Se usará Java 1.4 o superior. Todo lo necesario está disponible en <http://java.sun.com>. Cualquier entorno de desarrollo es útil para realizar la práctica. Basta en cualquier caso un editor de línea.

¹ [http://java.sun.com/j2se/1.5.0/docs/api/java/lang/System.html#exit\(int\)](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/System.html#exit(int))

Compilador de Modula-2 (excepcionalmente podrá seguir usándose)

El compilador de **Modula-2** está disponible en los Centros Asociados y en la web de la asignatura, en la dirección <http://www.lsi.uned.es>. También estará en las direcciones de las asignaturas de Programación I y Programación II de la UNED. Está asimismo disponible en el CDROM de la carrera.

Puede utilizarse cualquier compilador de Modula-2 bajo Windows/DOS o Linux. Los más conocidos para Windows/DOS son el *TopSpeed* (comercial), *GPM Total* (gratuito), y *FST* (gratuito).

En entorno Linux está disponible el *Mocka* y para Solaris está el *Ulm Módula-2*.

3.2.- Estudio del Coste

Estudiar y calcular si es posible la complejidad del algoritmo. Realizar un estudio teórico del coste del algoritmo implementado considerando las condiciones de poda realizadas.

3.3.- Casos de Prueba

Usar como caso de prueba el ejemplo inicial de la sección 1.

3.4.-Diseño del programa y estructuras de datos

Se seguirá en el diseño del programa un enfoque modular, encapsulando en objetos las estructuras de datos utilizadas, y los métodos de las mismas. Esto afecta en particular a las estructuras de datos de tipo pila, cola, montículo, conjunto etc.

Un ejemplo de diseño modular como el que se exige en la práctica se detalla en el capítulo 5 del libro de apoyo “*Esquemas Algorítmicos*” (Cuadernos de la UNED N° 167).

En el caso de tener que decidir entre varias implementaciones del mismo esquema, es obligatorio que se utilice la más eficiente.

4.- ENTREGA DE LA PRÁCTICA

PARA TODOS LOS ALUMNOS (Plan nuevo y plan antiguo)

1. La asistencia a las sesiones de prácticas es obligatoria SIN excepciones.
2. El calendario y procedimiento para asistir a las sesiones de prácticas está publicado en su Centro Asociado
3. El plazo de entrega de la documentación y de la práctica lo establece el Tutor de prácticas de cada Centro Asociado.
4. El Tutor califica la práctica, informa al alumno y en su caso la revisa de acuerdo con los horarios y procedimiento que establezca el Centro Asociado.
5. La Sede Central NO informa de las calificaciones de las prácticas
6. TODOS los alumnos deberán registrarse a través del Curso Virtual (acceso desde CiberUNED) con el Tutor/a con el que hayan asistido a las sesiones presenciales obligatorias a fin de que su práctica pueda ser calificada.
7. No se recogerá documentación alguna de alumnos que no hayan asistido a las sesiones obligatorias.
8. La práctica se debe aprobar en la misma o anterior convocatoria para que se pueda calificar la asignatura. En caso contrario la calificación será de suspenso.
9. La práctica se entregará empaquetada en un *.ZIP tanto en el entorno WebCT como al Tutor. La falta de cualquiera de ellas será motivo suficiente para quedar excluida de la convocatoria.

TUTORES

1. Este año NO hay hojas de lectura óptica.
2. La práctica se califica mediante un entorno web accesible desde los Cursos Virtuales
3. Todos los tutores deben ponerse en contacto con D. Miguel Rodríguez Artacho (prog3@lsi.uned.es) para tener acceso a la aplicación de entrega de las notas de práctica
4. Los alumnos se dan de alta en la aplicación de corrección y calificación de prácticas accesible desde WebCT de acuerdo con las indicaciones del Tutor. Aquel alumno que no se dé de alta no aparecerá en la lista de presentados a la misma y no constará su nota en el entorno de calificación y revisiones.

Material que hay que entregar al Tutor

Se confeccionará una memoria escrita con índice:

1. Respuesta a las cuestiones planteadas en este enunciado
2. Un ejemplo de ejecución para el caso de prueba
3. Estudio del coste del algoritmo tal como se describe en el apartado 3.2
4. Un listado del código fuente COMPLETO

Se adjuntará un CD con el nombre y apellidos del alumno escrito en el mismo y conteniendo:

1. Fuentes de cada uno de los módulos
2. Ejecutable del programa principal
3. Ficheros de los casos de prueba
4. Memoria de la práctica en Word, LaTeX o StarOffice.

El disquete y toda la documentación que se entregue deberá tener bien legible: **Nombre y apellidos, C. Asociado, teléfono y e-mail.**

El alumno debe asegurarse de que no se da ninguna de las siguientes circunstancias, ya que implican automáticamente una calificación de SUSPENSO:

- **Código:** El código no compila, no está desarrollado en Java, no se corresponde con el pseudocódigo recogido en la documentación, no es original, está copiado de la red, academia, compañero, etc., o no sigue un diseño OO encapsulado o modular.
- **Ejecutable:** El ejecutable no termina, se queda sin memoria con ejemplares pequeños o aborta sin justificación. El ejecutable no lee los ficheros previstos en el formato adecuado. No trata los argumentos o no se ajusta a las especificaciones.
- **Documentación:** No se presenta en papel o está incompleta.
- **Soporte:** No se puede leer, o contiene un virus de cualquier tipo. A este respecto, las prácticas en las que se detecte cualquier tipo de virus estarán suspensas.

Entrega en WebCT

Se entregará en el entorno WebCT como trabajo, de acuerdo con las instrucciones que se pondrán en el entorno. Esta entrega es **obligatoria**. Las prácticas se procesarán automáticamente mediante un software de tratamiento masivo para detectar copias. Los alumnos con prácticas copiadas estarán suspensos y serán objeto de apertura de expediente disciplinario.

La práctica debe realizarse de forma individual y se entregará al TUTOR en las fechas que fije cada Centro Asociado.

El Centro Asociado publicará las notas de las prácticas. El equipo docente **NO INFORMARÁ DE LAS NOTAS DE PRÁCTICAS**, debiendo el Alumno dirigirse al Centro Asociado para conocerlas.

ANEXO I

Bibliografía y tutoriales de Java

La referencia más útil es el libro “*Piensa en Java*” (“*Think in Java*”) de Bruce Eckel, disponible en varias editoriales.

Hay una versión completa y gratuita en la red en <http://www.mindview.net/Books/TIJ>.

En www.javahispano.org hay multitud de tutoriales en español.

CURSOS GRATUITOS DE SUN Microsystems (www.sun.com)

Para los alumnos matriculados, están disponibles Cursos gratuitos ofrecidos por Sun Microsystems (<http://www.sun.com>) accesibles desde el portal de entrada <https://learningcenter-sai.sun.com/> utilizando el código: **ES549874** (Código corporativo asociado a la UNED. Sólo para alumnos y personal de la Universidad)

Alumnos sin TUTOR

No hay sesiones de prácticas para alumnos sin Tutor, ya que no es posible según la normativa de la UNED matricularse en aquellos Centros que no disponen de Tutor de esta asignatura.

Cada alumno contactará con el Centro Asociado más cercano a su domicilio para organizarse la asistencia a las sesiones presenciales de prácticas.