

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>			
Centro Educativo <b>IES Campanillas (sede PTA)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>		Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Ejercicio de aplicación de POO</b>	Temática: <b>POO</b>	Fecha <b>05/02/2019</b>	Pág. <b>1/2</b>

## EJERCICIO

1. Implemente un tablero de juego para realizar sudokus. Para ello, se deben implementar las siguientes clases:

Clases:	Atributos:	Métodos:
Casilla	Valor: entero entre 1 y 9; 0 para indicar que no está relleno  Modificable: booleano	setValor: recibe entero, devuelve verdadero si se ha podido cambiar el valor (depende del valor del atributo Modificable)
		getValor: devuelve el valor entero que almacena la casilla
		constructor: recibe entero y si es modificable
Tablero	Array de 9x9 objetos de clase Casilla	setCasilla: recibe entero y coordenadas en las que se quiere cambiar el valor, devuelve verdadero si se puede cambiar el valor y se verifica que no se contravienen reglas (úsese el método checkTablero)
		toString: representa en una cadena de caracteres con formato el tablero, junto con su sistema de coordenadas
		checkTablero: comprueba si se cumplen las reglas de juego; devuelve verdadero si no se contravienen reglas
		checkJuego: comprueba si se ha completado el tablero; devuelve verdadero en ese caso
		constructor: recibe array bidimensional de 9x9 enteros, y los valores entre 1 y 9 tendrán atributo modificable a falso; el valor 0 llevará modificable a verdadero

La idea es que, a partir de un tablero inicialmente relleno en una matriz de 9x9 enteros (con el 0 indicaremos un valor por rellenar), el jugador pueda indicar fila y columna donde introducir un valor. El programa debe asegurar que el número que se quiere cambiar o introducir no es de los dados inicialmente y que si el usuario introduce un número que incumpla las reglas de juego, el programa avise y no lleve a cabo el cambio (o lo deshaga).

Las reglas del juego son sencillas:

- Cada casilla debe completarse con un solo dígito, que oscile entre 1 y 9
- No se pueden repetir dígitos en la misma fila, columna o subgrupo de 3x3
- No se pueden eliminar los valores iniciales
- El tablero tiene una solución única (es decir, si se completa, es que se ha rellenado bien).

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>			
Centro Educativo <b>IES Campanillas (sede PTA)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>			Profesor <b>Juan Antonio Jiménez Morales</b>
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Ejercicio de aplicación de POO</b>	Temática: <b>POO</b>	Fecha <b>05/02/2019</b>	Pág. <b>2/2</b>

La salida de toString de la clase Tablero podría arrojar algo similar a esto:

```

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

```