



CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma

MÓDULO: PROGRAMACIÓN 1º B

EVALUACIÓN: 2ª – UT5 **FECHA:** 05/02/2025

Ejercicio 1.

En la clase **TestMain** se recibe una cuadrícula de sudoku rellena de un jugador y debes comprobar si ha sido rellena correctamente.

La cuadrícula de un sudoku consiste en $9 \times 9 = 81$ celdas divididas en 9 subcuadrículas de $3 \times 3 = 9$ celdas. Para que la cuadrícula esté correcta, cada fila, columna y subcuadrícula debe contener los números del 1 al 9 sin repetirse.

Utiliza las clases que encontrarás en el **paquete ejercicio1**.

Clase TestMain

- Utiliza esta clase para comprobar que el validador de sudokus funciona correctamente. Esta clase utiliza 5 cuadrículas de sudoku con diferentes características.
- **NO modifiques el código de esta clase.**
- La salida de la ejecución debe ser:


```
CORRECTO
Falla fila 5
Falla columna 2
Falla subcuadrícula [1][1]
Falla fila 1
```

Clase SudokuValidator

Esta clase contiene un solo atributo que guarda la cuadrícula que hay que validar.

Completa los siguientes métodos:

- **validarFila(int f)** → Comprueba que la fila f contiene los números del 1 al 9 sin repetirse.
- **validarColumna(int c)** → Comprueba que la columna c contiene los números del 1 al 9 sin repetirse.
- **validarSubCuadrícula(int f, int c)** → Comprueba que la subcuadrícula de 3×3 que comienza en la fila f y columna c contiene los números del 1 al 9 sin repetirse.

	<p>CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma</p> <p>MÓDULO: PROGRAMACIÓN 1º B</p> <p>EVALUACIÓN: 2ª – UT5 FECHA: 05/02/2025</p>
---	--

- **validar()** → Comprueba que todas las filas, columnas y subcuadrículas son correctas. **Muestra por pantalla el primer error que encuentra:**
 - **Si falla una fila** muestra: "Falla fila f", siendo f el nº de la fila comenzando en el valor 1.
 - **Si falla una columna** muestra: "Falla columna c", siendo c el nº de la columna comenzando en el valor 1.
 - **Si falla una subcuadrícula 3x3** muestra: "Falla cuadrícula [f][c]", siendo f y c los números de la fila y la columna donde comienza la subcuadrícula comenzando en el valor 1.
 - **Si está correcto** muestra: "CORRECTO".

Ejercicio 2.

El ejercicio 2 presenta tres clases: **Vehículo**, que representa un vehículo con matrícula, marca y propietario; **Aparcamiento**, que simula un aparcamiento gestionando la inserción, eliminación y búsqueda de vehículos mediante métodos como insertarVehiculo, eliminarVehiculo y buscarPorMatricula. Además, **Aparcamiento** incluye métodos para mostrar el estado del aparcamiento, contar plazas libres, e invertir el orden de los vehículos horizontal y verticalmente. Por último, en la clase **Main** se dejan los métodos que se deben desarrollar comentados.

Clase Vehículo


- **No modifiques el código de esta clase.**

Clase Aparcamiento

Atributos:

- La clase Aparcamiento tiene un atributo llamado **aparcamiento que es una matriz bidimensional de objetos Vehículo**. Esta matriz representa las plazas de aparcamiento disponibles.
- También tiene un **atributo entero llamado filas** que representa el número de filas en la matriz aparcamiento.
- Además, cuenta con un **atributo entero llamado columnas** que representa el número de columnas en la matriz aparcamiento.

Constructor:

	<p>CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma</p> <p>MÓDULO: PROGRAMACIÓN 1º B</p> <p>EVALUACIÓN: 2ª – UT5 FECHA: 05/02/2025</p>
---	--

- El constructor de la clase Aparcamiento recibe dos enteros, N y M, que representan el número de filas y columnas, respectivamente, para inicializar el aparcamiento.

Métodos:

mostrarAparcamiento(): Este método imprime en la consola el estado actual del aparcamiento. Si una plaza está vacía, se muestra [LIBRE]; de lo contrario, se muestra la información del vehículo estacionado en esa plaza (matrícula, marca y propietario)

buscarPorMatricula(String matricula): Este método busca un vehículo en el aparcamiento dada su matrícula. Si lo encuentra, devuelve una cadena con la información del vehículo usando el método toString() de la clase Vehiculo; de lo contrario, devuelve el mensaje "No se ha encontrado matrícula buscada"


contarPlazasLibres(): Este método cuenta y devuelve el número de plazas de aparcamiento que están vacías (es decir, donde el valor en la matriz aparcamiento es null).

insertarVehiculo(Vehiculo vehiculo): Este método intenta insertar un vehículo en la primera plaza libre que encuentre en el aparcamiento. Si no hay plazas libres, imprime el mensaje "No se puede insertar vehículo, no hay plazas libres."

eliminarVehiculo(String matricula): Este método busca un vehículo por su matrícula y, si lo encuentra, lo elimina del aparcamiento estableciendo la plaza correspondiente en null. Devuelve true si el vehículo fue eliminado, o false en caso contrario, Si no encuentra el vehículo, imprime "Vehículo no encontrado." y devuelve false.

invertirHorizontal(): Este método invierte horizontalmente las filas del aparcamiento, intercambiando la primera fila con la última, la segunda con la penúltima, y así sucesivamente.

invertirVertical(): Este método invierte verticalmente las columnas del aparcamiento dentro de cada fila, intercambiando la primera columna con la última, la segunda con la penúltima, y así sucesivamente para todas las filas.

	<p>CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma</p> <p>MÓDULO: PROGRAMACIÓN 1º B</p> <p>EVALUACIÓN: 2ª – UT5 FECHA: 05/02/2025</p>
---	--

Clase Main

- Se debe hacer uso del código dado, para eso se debe quitar los comentarios de las líneas de código dadas.
- En el apartado **Aparcamiento lleno con los siguientes vehículos** se debe añadir el código en el **Main** para insertar vehículos hasta que el aparcamiento se llena. La **matricula introducida** debe respetar el siguiente formato “**abcd01**” más el número de la plaza, la **marca introducida** debe respetar el siguiente formato “**marca**” más el número de la plaza y el **propietario introducido** debe respetar el siguiente formato “**prop_id**” más el número de la plaza.

Resultado del APARCAMIENTO LLENO:

```
*****
*****Aparcamiento lleno con los siguientes vehiculos*****
*****
[abcd010-marca0-prop_id0] [abcd011-marca1-prop_id1] [abcd012-marca2-prop_id2]
[abcd013-marca3-prop_id3] [abcd014-marca4-prop_id4] [abcd015-marca5-prop_id5]
[abcd016-marca6-prop_id6] [abcd017-marca7-prop_id7] [abcd018-marca8-prop_id8]
Hay 0 plazas libres.
```

RESULTADOS FINALES del MAIN:

```
*****
***** Aparcamiento Estado Inicial *****
*****
[LIBRE] [LIBRE] [LIBRE]
[LIBRE] [LIBRE] [LIBRE]
[LIBRE] [LIBRE] [LIBRE]
Hay 9 plazas libres.

*****
*****Aparcamiento lleno con los siguientes vehiculos*****
*****
[abcd010-marca0-prop_id0] [abcd011-marca1-prop_id1] [abcd012-marca2-prop_id2]
[abcd013-marca3-prop_id3] [abcd014-marca4-prop_id4] [abcd015-marca5-prop_id5]
[abcd016-marca6-prop_id6] [abcd017-marca7-prop_id7] [abcd018-marca8-prop_id8]
Hay 0 plazas libres.

*****
*Intentar añadir Vehículo aunque el aparcamiento está lleno *****
*****
No se puede insertar vehículo, no hay plazas libres.
```



CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma

MÓDULO: PROGRAMACIÓN 1º B

EVALUACIÓN: 2ª – UT5 **FECHA:** 05/02/2025

```
*****
***** Buscar por matrícula *****
*****
No se ha encontrado matrícula buscada
Marca: marca8, Matrícula: abcd018, Propietario: prop_id8


*****
***** Eliminar matrícula *****
*****
Vehículo no encontrado.
Vehículo no encontrado.
[LIBRE] [LIBRE] [abcd012-marca2-prop_id2]
[LIBRE] [LIBRE] [abcd015-marca5-prop_id5]
[abcd016-marca6-prop_id6] [abcd017-marca7-prop_id7] [LIBRE]

*****
***** Invertir Horizontal *****
*****
[abcd016-marca6-prop_id6] [abcd017-marca7-prop_id7] [LIBRE]
[LIBRE] [LIBRE] [abcd015-marca5-prop_id5]
[LIBRE] [LIBRE] [abcd012-marca2-prop_id2]
```

```
*****
***** Invertir Vertical *****
*****
[LIBRE] [abcd017-marca7-prop_id7] [abcd016-marca6-prop_id6]
[abcd015-marca5-prop_id5] [LIBRE] [LIBRE]
[abcd012-marca2-prop_id2] [LIBRE] [LIBRE]

*****
***** Invertir Vertical y Horizontal*****
*****
[LIBRE] [LIBRE] [abcd012-marca2-prop_id2]
[LIBRE] [LIBRE] [abcd015-marca5-prop_id5]
[abcd016-marca6-prop_id6] [abcd017-marca7-prop_id7] [LIBRE]

Process finished with exit code 0
```

	CICLO FORMATIVO: Desarrollo de Aplicaciones Multiplataforma		
	MÓDULO: PROGRAMACIÓN 1º B		
	EVALUACIÓN: 2ª – UT5 FECHA: 05/02/2025		

RÚBRICA

Apartado	Elemento	Puntuación	
SudokuValidator	validarFila	0,5	
	validarColumna	0,5	
	validarSubcuadrícula	1	
	validar	1	
Ejercicio 2			
Aparcamiento	atributos	0,25	
	Constructor	0,25	
	mostrarAparcamiento	0,5	
	buscarPorMatricula	0,5	
	contarPlazasLibres	0,25	
	insertarVehículo	1,25	
	eliminarVehículo	0,25	
	invertirHorizontal	1,5	
	invertirVertical	2	
Main	Llenar el aparcamiento con los formatos adecuados.	0,25	
Total			