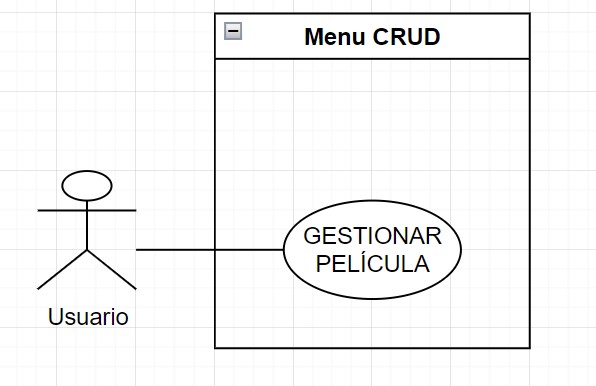
**Desarrollo de programa CRUD con Java**

Alfonso López Delgado

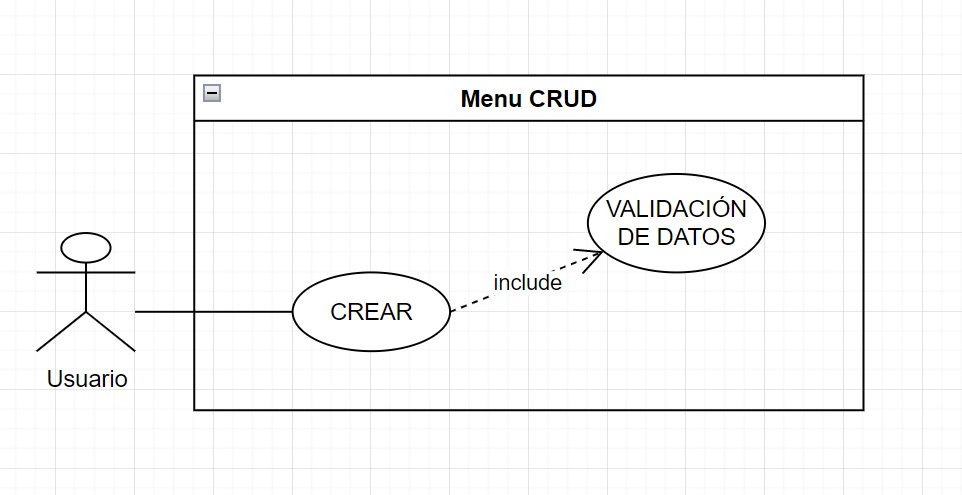
1ºDAM

**Análisis de casos de uso CRUD tema películas**

1. Gestionar película

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Gestionar película | El usuario puede elegir entre 5 opciones para tratar los datos, crear, modificar, borrar, consultar y salir del menú | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Iniciar programa * Mostrar menú * Introducción por teclado | * Dirección opciones | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Mostrar menú | * Mostrar menú * Dato | * Mostrar opción | |

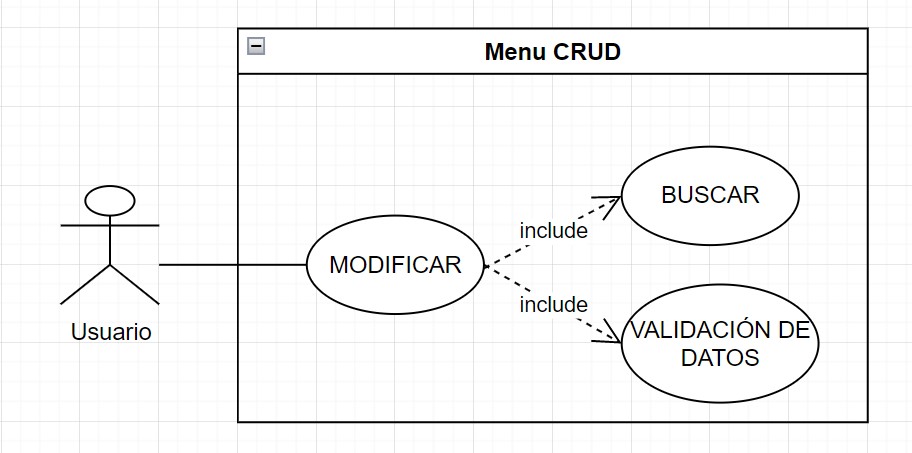
1. Crear



|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Crear película | El usuario crea una película con los datos que se le pide | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Array para almacenar el nombre * Array para almacenar la fecha * Array para almacenar la valoración * Introducción por teclado de película | * Película nueva previa validación de datos. | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Array fecha * Array valoración | * Mensaje | * Dato (nombre, fecha, valoración) | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Validación de datos | Validación de datos introducidos por el usuario | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato introducido por teclado * Dato dentro de los parámetros | * Dato insertado en los arrays dentro de los parámetros establecidos | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Dato | * Mensaje * Dato sin validar | * Dato (nombre, fecha, valoración) validado | |

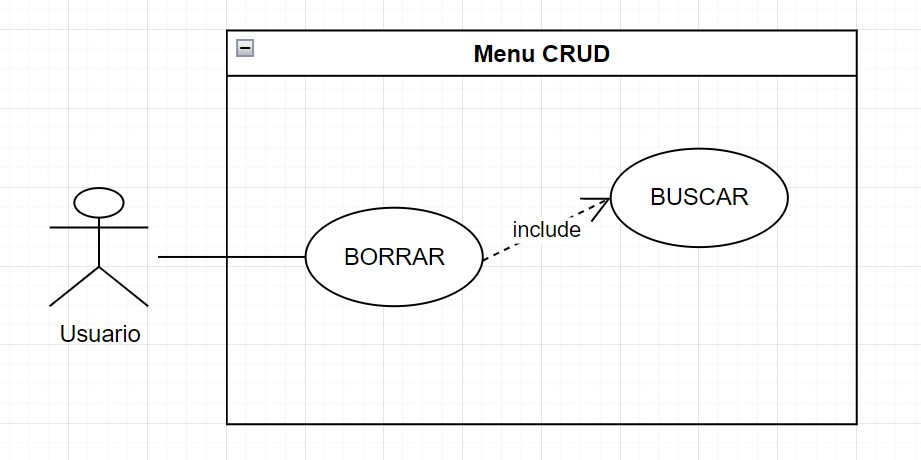
1. Modificar

****

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Modificar | Modificación de algún dato, que haya introducido el usuario previamente. | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato almacenado en Array * Introducción del nuevo dato | * Eliminación del antiguo dato y sustitución por el nuevo | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Dato | * Mensaje | * Dato (nombre, fecha, valoración) nuevos | |

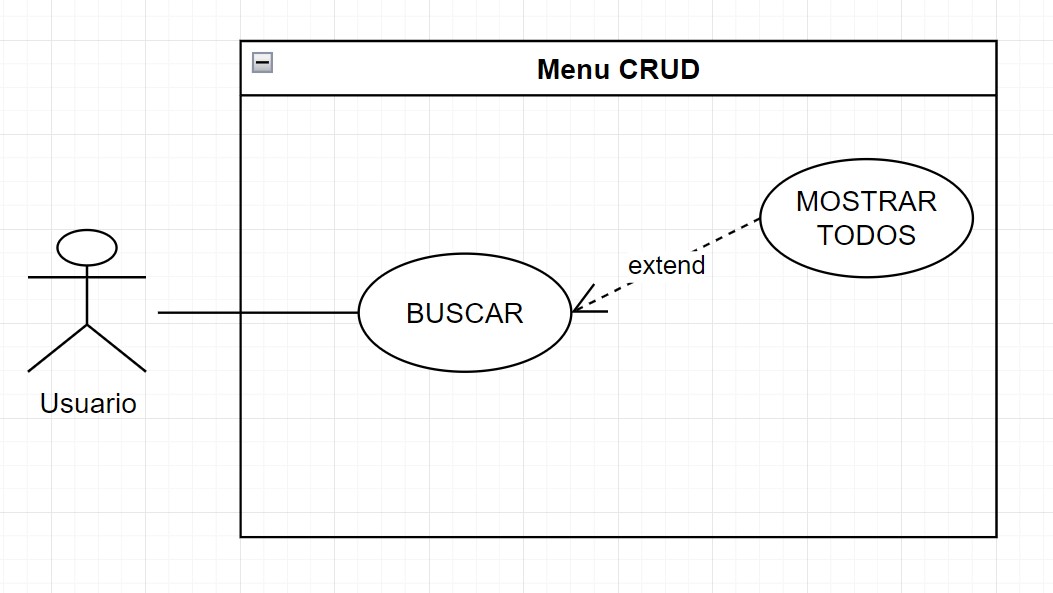
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Buscar | Buscar el dato introducido por el usuario | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato introducido por teclado * Dato almacenado en Array | * Dato encontrado | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Dato posición | * Dato teclado | * Dato posición | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Validación de datos | Validación de datos introducidos por el usuario | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato introducido por teclado * Dato dentro de los parámetros | * Dato insertado en los arrays dentro de los parámetros establecidos | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Array fecha * Array valoración * Dato | * Mensaje * Dato sin validar | * Dato (nombre, fecha, valoración) validado | |

1. Borrar

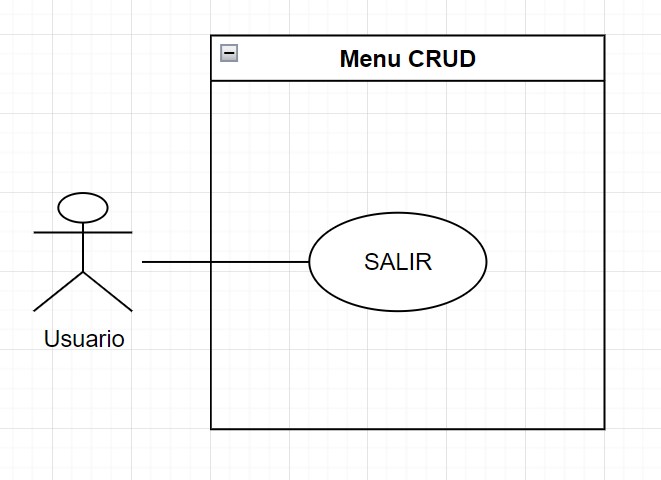
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Borrar | Borrado de un dato en el array mediante la sustitución de otro dato neutral. | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato almacenado en Array | * Sustitución por elemento neutral (como un espacio) | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre | * Mensaje | * Dato (“”, “”, “”) | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Buscar | Buscar el dato introducido por el usuario | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato introducido por teclado * Dato almacenado en Array | * Dato encontrado | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Dato posición | * Dato teclado | * Dato posición | |

1. **Buscar**

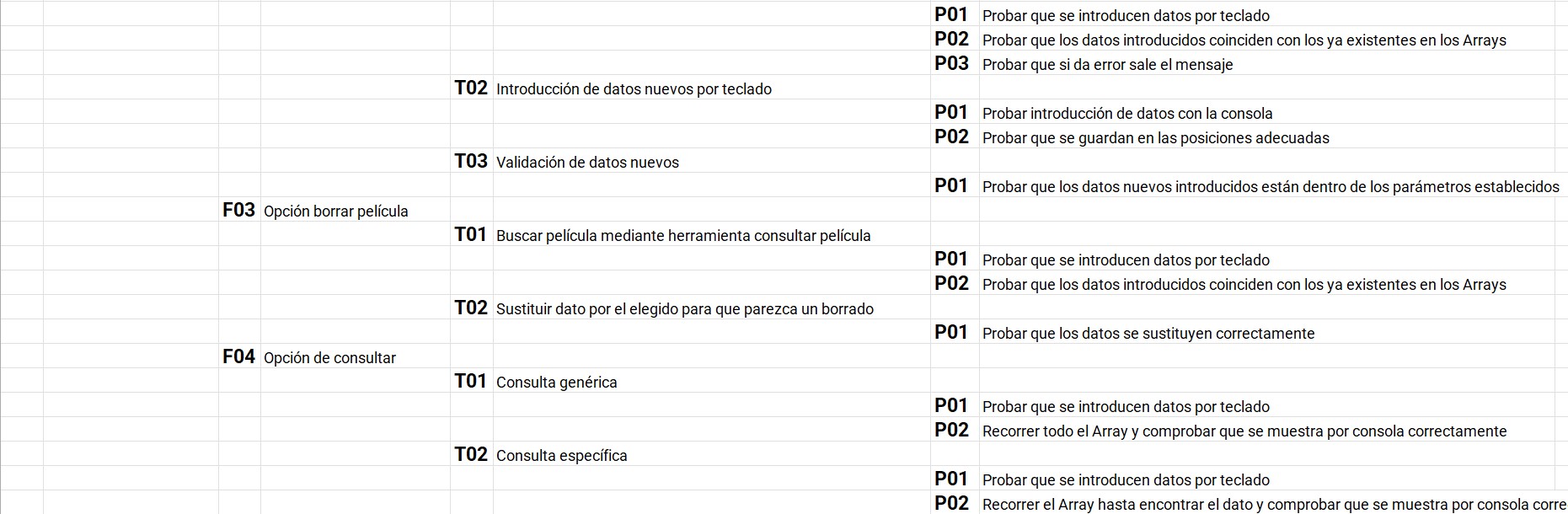
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Buscar | Buscar el dato introducido por el usuario | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Dato introducido por teclado * Dato almacenado en Array | * Dato encontrado * Mensaje | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre * Dato posición | * Dato teclado | * Mensaje | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Mostrar todos | El usuario puede elegir si hacer una búsqueda específica o mostrar todos los datos del array | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Elegir opción mostrar todo | * Mostrar todos los datos del array | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | | * Array nombre |  | * Mensaje | |

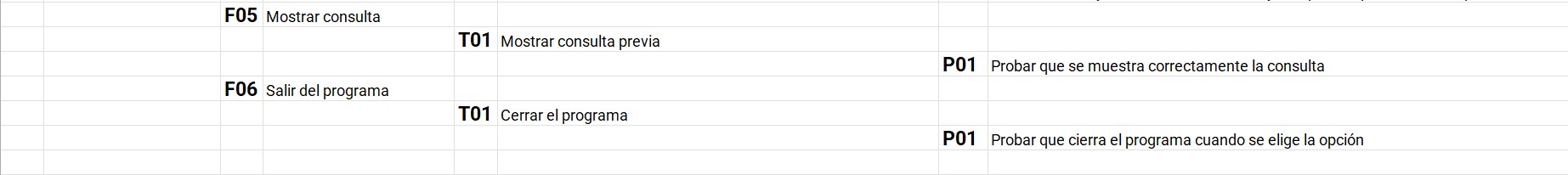
1. Salir

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | NOMBRE | DESCRIPCIÓN | |
| |  |  | | --- | --- | | Salir | Opción salir del menú | |
| |  |  | | --- | --- | | PRECONDICIONES | POSTCONDICIONES | |
| |  |  | | --- | --- | | * Elegir opción | * Mensaje * Apagar consola | |
| |  |  |  | | --- | --- | --- | | DATOS | DATOS ENTRADA | DATOS SALIDA | |
| |  |  |  | | --- | --- | --- | |  |  | * Mensaje | |

**RFTP**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

****

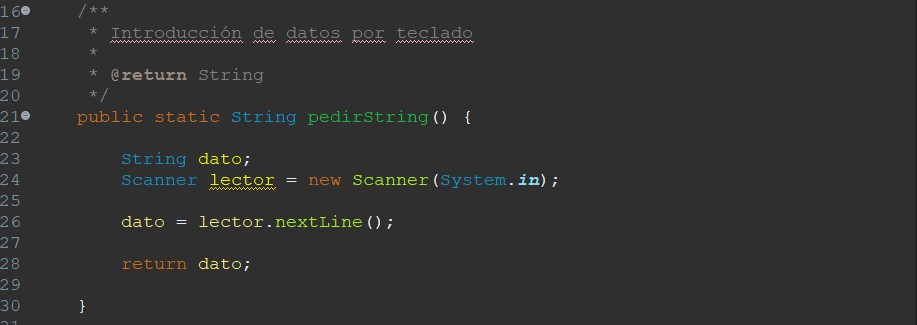
**Código comentado**

Este apartado será una breve explicación del código del programa que se ha realizado.

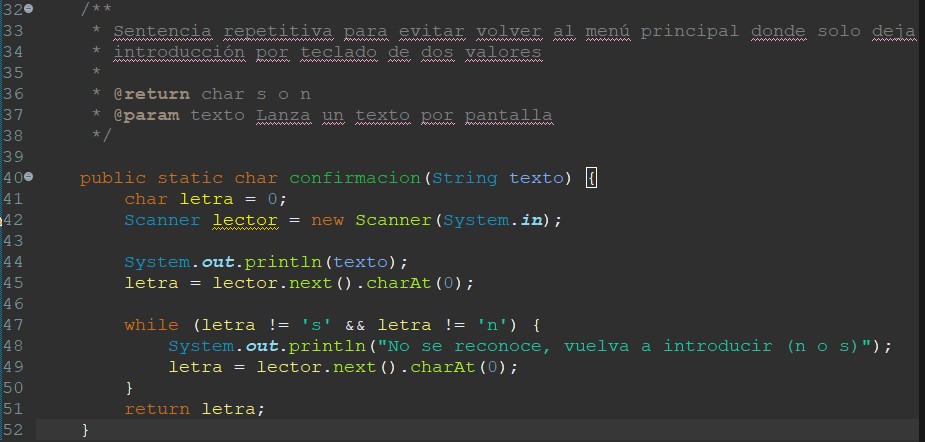
**Clase utilidades**

Es creado con la intención de la reutilización del código que se repetirá en reiteradas ocasiones durante el proyecto.

* **Método pedirString()** 🡪 Creado con la finalidad de introducir por teclado los valores String que necesite el usuario.

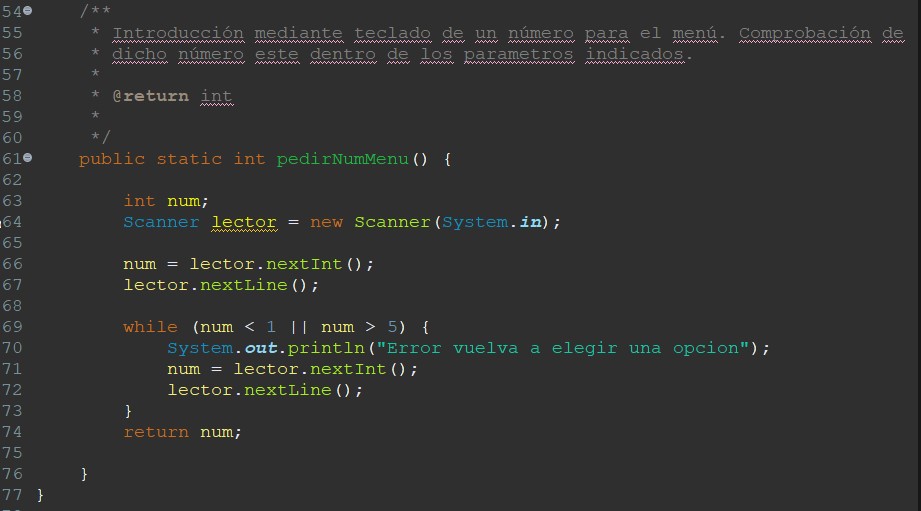
***Explicación del código:*** Creación de método donde retornara un dato String pedida en la variable “dato” mediante la clase Scanner.

* **Método confirmación()** 🡪 Creado con la finalidad de seguir en la misma opción del menú si el usuario lo requiere y evitar salir y volver a entrar.

***Explicación del código:*** Creación de un método donde retornara un dato char. Se pedirá una letra con la variable tipo char “letra” con la clase Scanner. El dato introducido entrará en un bucle while donde se repetirá mientras la letra sea distinto de ‘s’ o ‘n’, unidos por un operador lógico AND, y lanzará un mensaje de “error”.

* **Método pedirNumMenu()** 🡪 Creado con la finalidad de pedir una opción al usuario para elegir en el menú.

***Explicación de código***: Creación de método en el cual retornara un dato int.

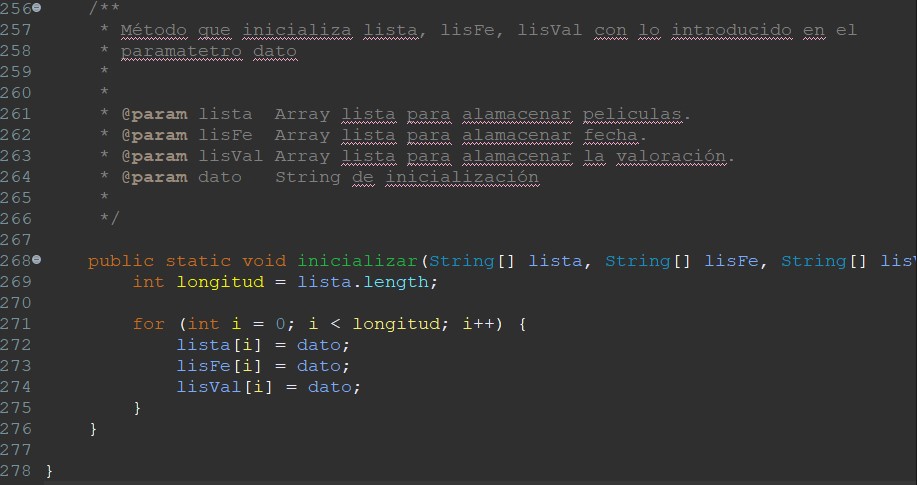
Se pedirá un número recogido con la variable “num”, de tipo int, por teclado mediante la clase Scanner. El dato entra en un bucle while el cual se repetirá mientras dichos parámetros se cumplan, unidos por un operador lógico OR, y lanzará un mensaje por pantalla de “error”.

**Clase menuFinal**

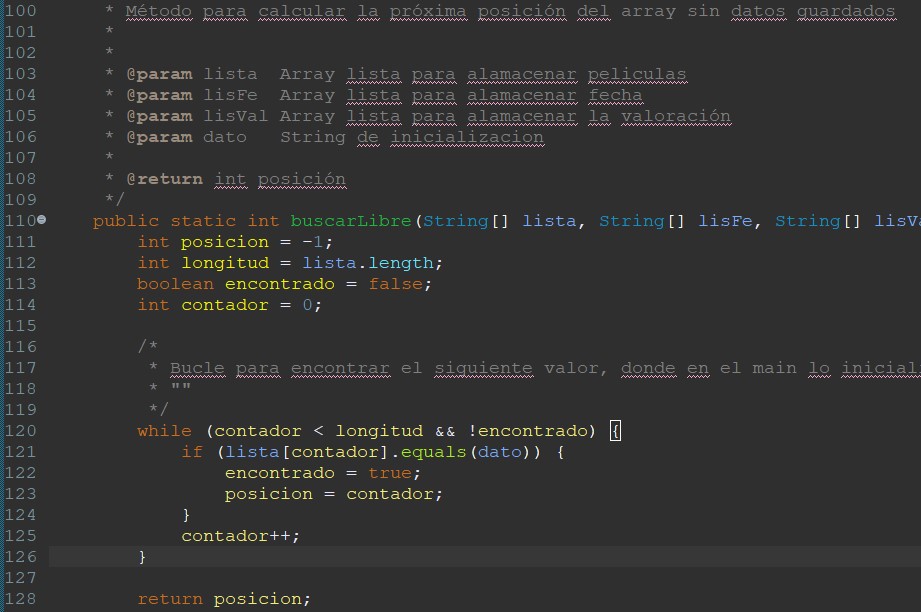
* **Método menú()** 🡪 La estética del menú que se imprimirá por consola.

***Explicación del código:*** Mediante el conjunto de instrucciones System.out.println(“”); se le dará una forma gráfica.

* **Método inicializar(String[] lista, String[] lisFe, String[] lisVal, String dato)** 🡪 Su finalidad es rellenar todas las posiciones de los tres arrays lista, lisFe, lisVal con un dato.

***Explicación del código:*** Creación de un método donde no devolverá nada y le dirá al programa que rellene los parámetros declarados. Mediante la variable “longitud” y la función “length” recogeremos la longitud del array y con un bucle for recorreremos todo el array introduciendo en cada posición del array el parámetro “dato”.

* **Método buscarLibre(String[] lista, String[] lisFe, String[] lisVal, String dato) 🡪** Su finalidad es buscar la primera posición del array donde haya un hueco libre.

***Explicación del código:*** Creación de un método donde retornará una Int el cual será la posición donde se localiza el primer hueco libre. Mediante un bucle while con la condición de que se ejecute mientras la variable contador sea menor que la longitud del array AND la variable booleana distinta de la declarada, y anidando una sentencia if comparando cada posición del array con el “dato”. Cuando la variable booleana es “true” sale del while con la variable “posición” igual al número de vueltas que ha dado el contador y retornará la posición donde se ha encontrado el primer “dato”.

* Método buscar**(String[] lista, String[] lisFe, String[] lisVal) 🡪** Su finalidad es buscar la posición del dato introducido por el usuario.

Explicación del código: Mediante la llamada de la clase utilidades.pedirString(); creada previamente, se pedirá un dato tipo String que se guardará en la variable “buscar”.