Logotipo

Descripción generada automáticamenteUniversidad Católica del Norte   
Facultad de Ingeniería y Ciencias Geológicas

Departamento de Ingeniería de Sistemas y Computación

UNIVERSIDAD CATOLICA DEL NORTE

FACULTAD DE INGENIERÍA Y CIENCIAS GEOLÓGICAS

**Taller Numero 1**

**Programación Avanzada.**

**Profesor a cargo:**Tomás Alberto Reiman Beltran.

**Ayudante :**Edgardo Antonio Ortiz Gonzales.

**Integrantes.**

-Ángel Cuevas.

**Rut:**

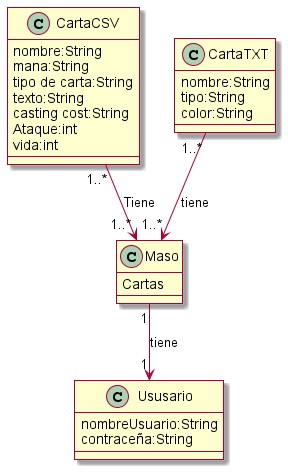
**-**21.009.361-3

**Correos:**

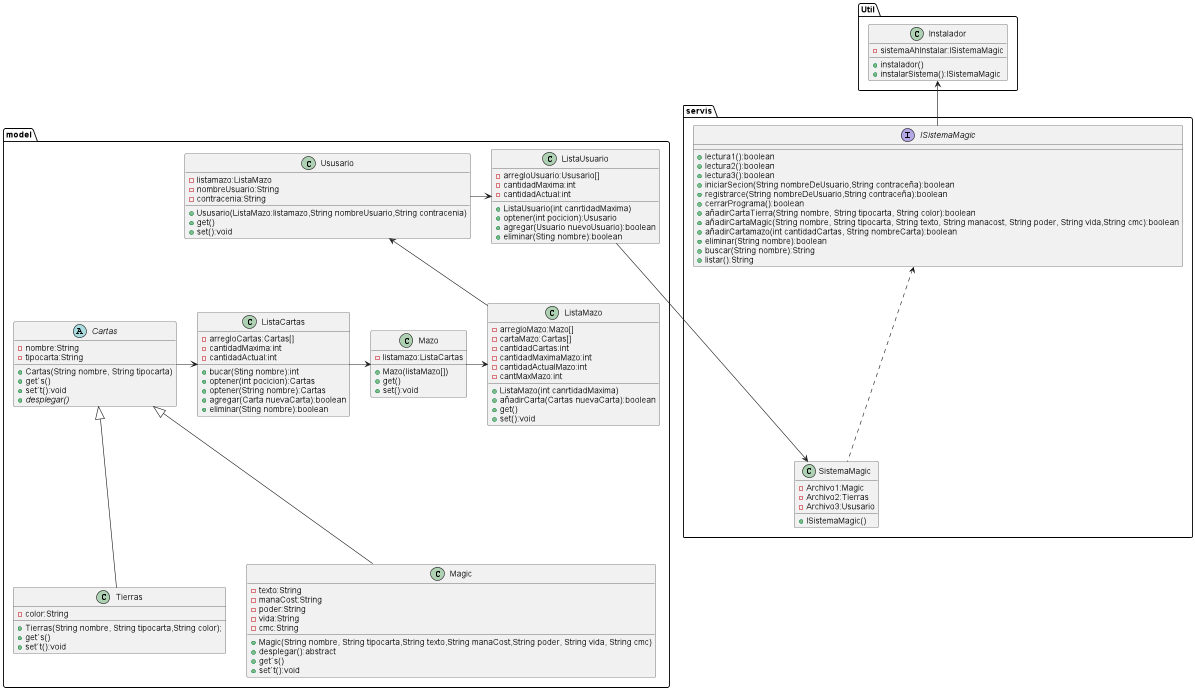
**-**[angel.cuevas@alumnos.ucn.cl](mailto:angel.cuevas@alumnos.ucn.cl)

**Paralelo.**

**-**C1



* Primero tenemos que crear el usuario que va a tener el maso siendo sus parámetros de nombre Usuario y contraseña Usuario.
* Aquí podemos encontrar el modelo de dominio que se tendrá la clase de carta CSV con el nombre, mana, tipo de carta, texto, casting, ataque y la vida siendo los parámetros de la clase con todos los parámetros como String exceptuando las de ataque y vida.
* Después tenemos la clase CartaTXT con el nombre, color y tipo, con todos los parámetros como string.
* El maso va a estar constituido por las cartas de los archivos txt que habremos hecho en la lectura de archivo.



**Clase abstracta cartas:** Tenemos los parámetros de nombre, tipo de carta que son los parámetros comunes que se podrán extender a las demás clases creando sus respectivos constructores y get´s () and set´s ().

**Clase Magic:** Aquí está la primera clase que extiende de la clase abstracta de cartas podemos encontrar los demás atributos que serían texto, mana, vida, poder y cmc construyendo su constructor y añadiendo los métodos get and set y esto datos se obtendrán de la primera lectura de archivo que se dirigirá a la lista cartas.

**Clase Tierras:** Aquí tendremos la clase de tierras en el cual ingresaremos los parámetros de color como se extiende de la clase Cartas le añadiremos en el constructor los atributo y agregaremos los métodos get a set.

**Lista carta:** Aquí tenemos Carta que llamara a la clase Carta para tener sus datos definimos una cantidad máxima y cantidad actual que sirve para recorrer la lista y donde podremos agrega, buscar, eliminar y obtener un usuario también al momento de leer el archivo txt es donde las cartas podrán almacenarse.

**Clase usuario:** Crearemos una clase usuarios para poder tener a los registrados aquí, tenemos los parámetros de nombre usuario, contraseña usuario y mi idea principal era poder almacenar los mazos en esta clase para tener la relación entre el usuario y sus mazos.

**Mazo:** Aquí se definió el vector del mazo para poder almacenar las cartas con un rango de 60 y señalando los métodos get y set.

**Lista Mazo:** Aquí se tendrán los vectores que se ocuparán en el mazo junto a las delimitaciones de cantidad máxima, cantidad actual y un método booleano para poder añadir la carta a la lista mazo.

**Lista usuario:** aquí tenemos los usuarios que llamamos a la clase usuario para tener sus datos definimos una cantidad máxima y cantidad actual que sirve para recorrer la lista y donde podremos agrega, buscar, eliminar y obtener un usuario.

**Tierras.TXT:** Aquí tenemos la información que se guardara en la segunda lectura de archivo donde tenemos los parámetros de nombre, tipo y color.

**Magic.TXT:** Son la información de todas las clases magic que se encuentran en el archivo txt.

**IsistemaCarta:** Aquí tenemos la interfaz que se encargara de extender todos los puntos que se no pregunta en el taller como las lecturas de archivo, las instancias de iniciar sesión, cerrar sesión agregar carta, buscar carta etc.

**SistemaCarta:** Aquí se implementará todo lo que se tenia en el ISsitemaCarta para realizar las funciones que después seremos capaces de llamar en el main como lo de buscar, agregar, lectura, iniciar sesión etc.

**Instalador:** Aquí instalaremos lo que tenemos en el IsistemaCarta llamando al sistema, luego crearemos un instalador que llamara a un new sistema y por último crear un instalar sistema que retornara el propio sistema creado.

**Formato de los Contratos:**

|  |  |
| --- | --- |
| OPERACIÓN | lecturaDeArchivo1. |
| DESCRIPCION | Se realizará la lectura y guardado de datos en las variables |
| PRECONDICIONES | Tener un archivo de texto.txt con los valores y las variables a considerar. |
| POSTCONDICIONES | Luego de realizar y tener los archivos estos se van a guardar en variables que se utilizaran más adelante. |

|  |  |
| --- | --- |
| OPERACIÓN | lecturaDeArchivo2 |
| DESCRIPCION | Se realizará la lectura y guardado de datos en las variables. |
| PRECONDICIONES | Tener un archivo de texto.txt con los valores y las variables a considerar. |
| POSTCONDICIONES | Luego de realizar y tener los archivos estos se van a guardar en variables que se utilizaran más adelante. |
|  |  |

|  |  |
| --- | --- |
| OPERACIÓN | lecturaDeArchivo3 |
| DESCRIPCION | Se realizará la lectura y guardado de datos en las variables. |
| PRECONDICIONES | Tener un archivo de texto.txt con los valores y las variables a considerar. |
| POSTCONDICIONES | Luego de realizar y tener los archivos estos se van a guardar en variables que se utilizaran más adelante. |

|  |  |
| --- | --- |
| OPERACIÓN | Eliminar |
| DESCRIPCION | Se buscará la carta para ser eliminada |
| PRECONDICIONES | Tener cartas ya leídas en el archivo.txt  Lista no este vacía |
| POSTCONDICIONES | Se eliminará la carta del sistema |
|  |  |

|  |  |
| --- | --- |
| OPERACIÓN | Agregar |
|  |  |
| DESCRIPCION | Se busca agregar una carta nueva diferente del archivo.txt ya leído |
| PRECONDICIONES | Que la lista no está lena y que los datos ingresados sean correctos |
| POSTCONDICIONES | Se agregará la carta a la lista de cartas |

|  |  |
| --- | --- |
| OPERACIÓN | Agregar Usuario |
| DESCRIPCION | Se busca crear un usuario nuevo para que pueda ingresar o tener un mazo en el sistema |
| PRECONDICIONES | Que el nombre de usuario o contraseña no exista ya en el sistema |
| POSTCONDICIONES | El usuario se agregará a la lista |

|  |  |
| --- | --- |
| OPERACIÓN | Buscar carta |
| DESCRIPCION | Se buscará la carta en el sistema |
| PRECONDICIONES | Que la carta exista en la lista  Que la lita no este vacía |
| POSTCONDICIONES | Se desplegará por pantalla si la carta se encuentra o no en el sistema |

|  |  |
| --- | --- |
| OPERACIÓN | Iniciar sesión |
| DESCRIPCION | Al momento de estar en el main se preguntará por el nombre de usuario |
| PRECONDICIONES | Si existe usuario en la lista usuario se podrá ingresar al menú principal |
| POSTCONDICIONES | Se accede al menú principal para poder realizar las operaciones. |

|  |  |
| --- | --- |
| OPERACIÓN | Registrarse |
| DESCRIPCION | Se registrará un nuevo usuario |
| PRECONDICIONES | Tener una clase usuario con los parámetros. |
| POSTCONDICIONES | Se almacenará el usuario registrado en la lista de usuarios |

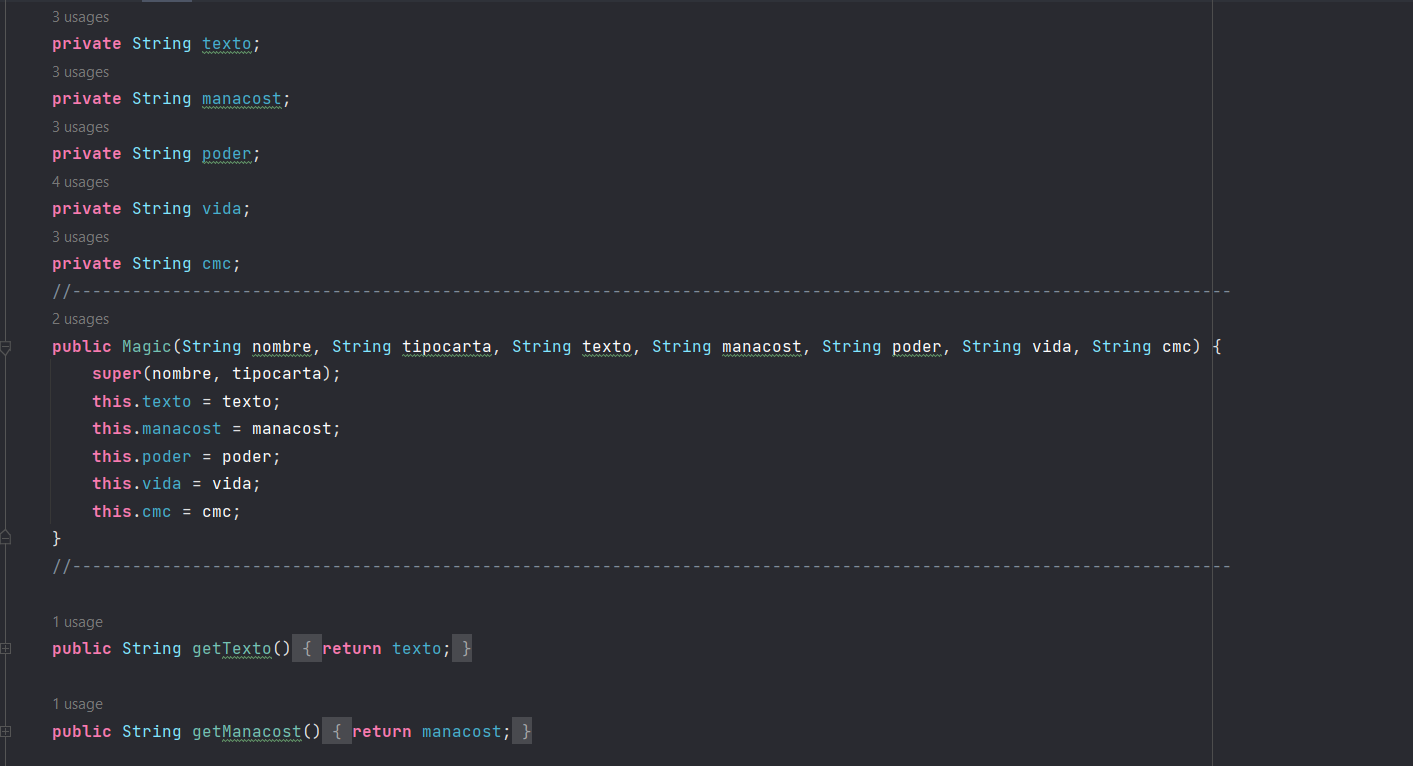
|  |  |
| --- | --- |
| OPERACIÓN | Cerrar Programa |
| DESCRIPCION | Al momento de terminar el main se guardará los usuarios registrados en un Ususarios.txt |
| PRECONDICIONES | Tener un archivo Usuarios.txt vacío para ver las cosas. |
| POSTCONDICIONES | Se guardarán los usuarios ingresados en el txt. |

|  |  |
| --- | --- |
| OPERACIÓN | listar |
| DESCRIPCION | Cuando e pida la opción se listarán las cartas que posee el mazo. |
| PRECONDICIONES | Tener cartas añadidas en la lista mazo |
| POSTCONDICIONES | Se listaran las cartas añadidas. |

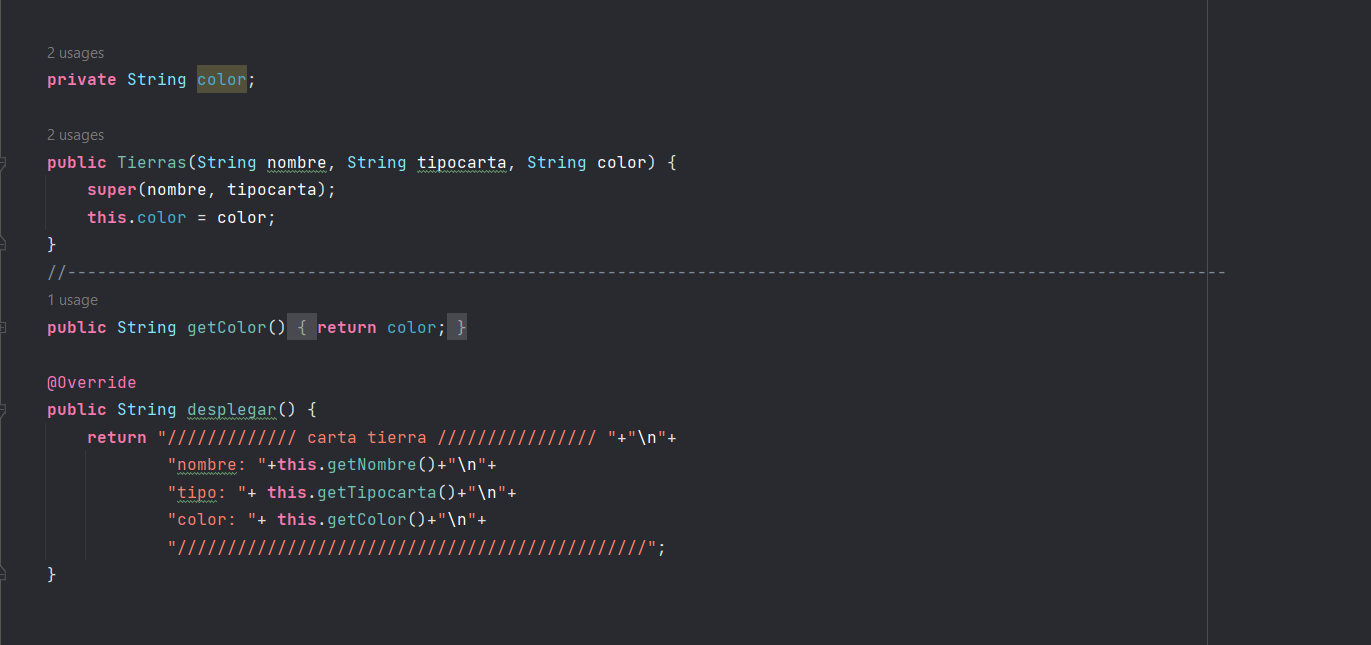
Entrega 2 código.

Esta es la clase abstracta la cual se encargará de tener los atributos generales que se van a encargar de extenderse a las demás clases como el nombre y el tipo de carta, también tendremos un sub programa publico abstracto llamado desplegar el cual se encargara de ejecutar los datos que se encuentren almacenado en la listaMagic

Unas de las clases que va a extender de la clase abstracta tierras será la clase magic la cual se encargara de tomar los atributos de la clase Cartas, añadiéndolo a su constructor y clase junto con el sub programa de desplegar que se le entregaran los datos de esta clase para que sea capas de desplegar la carta buscada ó listar las cartas solicitadas.



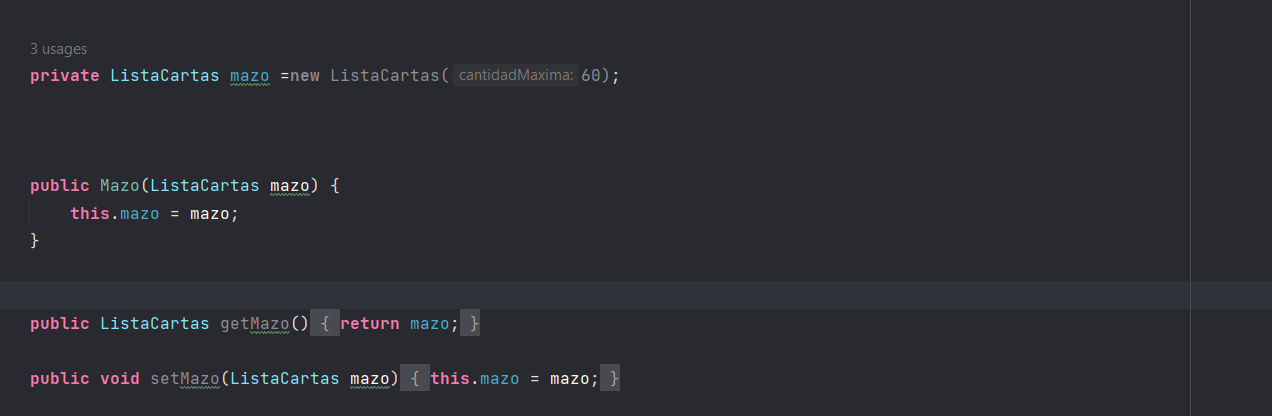
Esta clase tienen la misma función de la clase anterior posee un atributo de color y extiende los atributos de la clase carta.



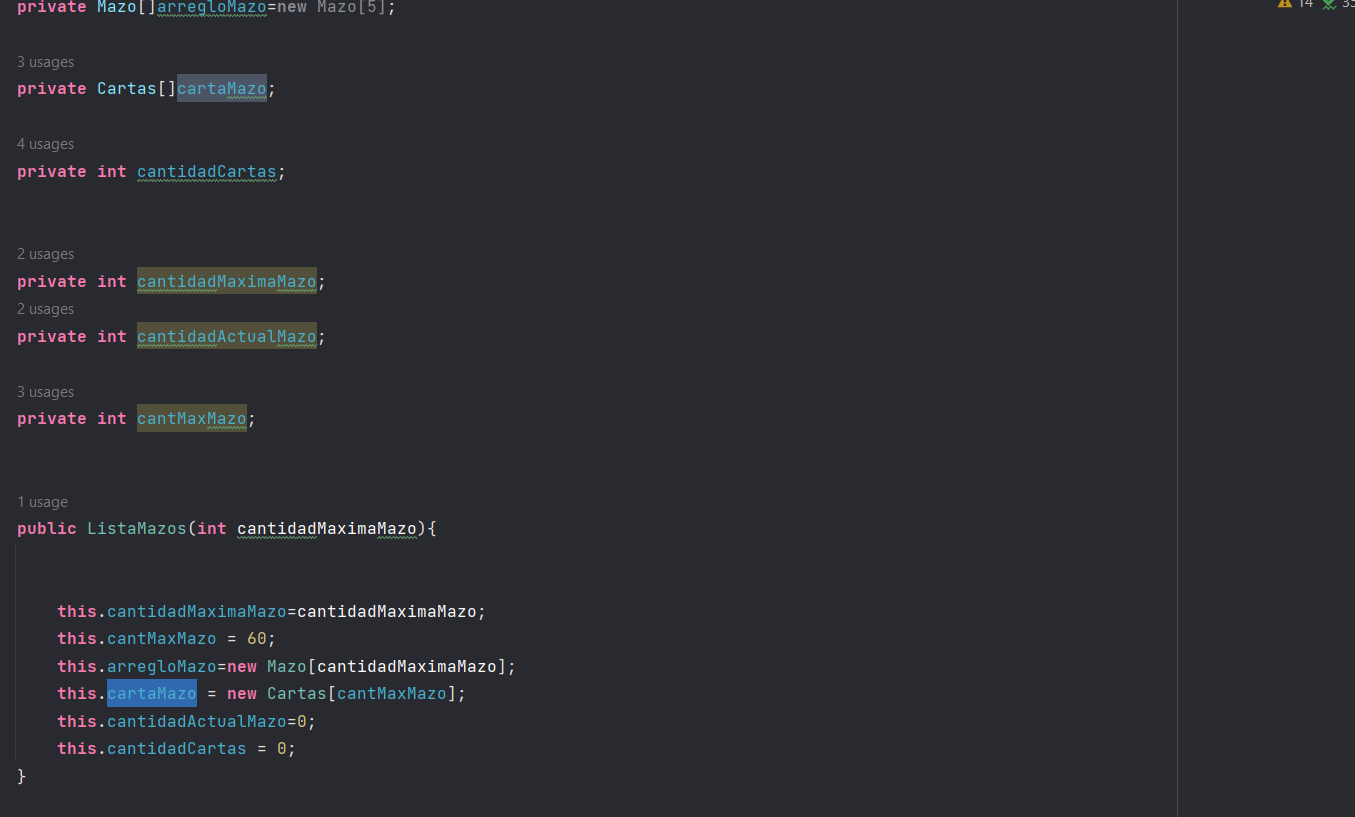
En la lista cartas es donde se podrá almacenar las cartas leídas por el Magic.txt, siento posible llamar dichos datos con el método de desplegar de las clases anteriores en esta clase podremos buscar por posición dentro de la lista, obtener la posición tanto por el lugar en que se encuentra como por el nombre de la carta, agregar una carta y eliminar la carta.



Después tenemos la clase mazo la cual se encargará de tener una lista de mazo con un rango de 60 cartas el cual será considerado como el mazo.



Aquí tendremos la lista mazo el cual se encargará de encapsular esta información de lo que se encuentra en el mazo que serian las cartas que se irán agregando, también tendremos la opción de añadir carta que se encargara de añadir carta en cartaMazo.

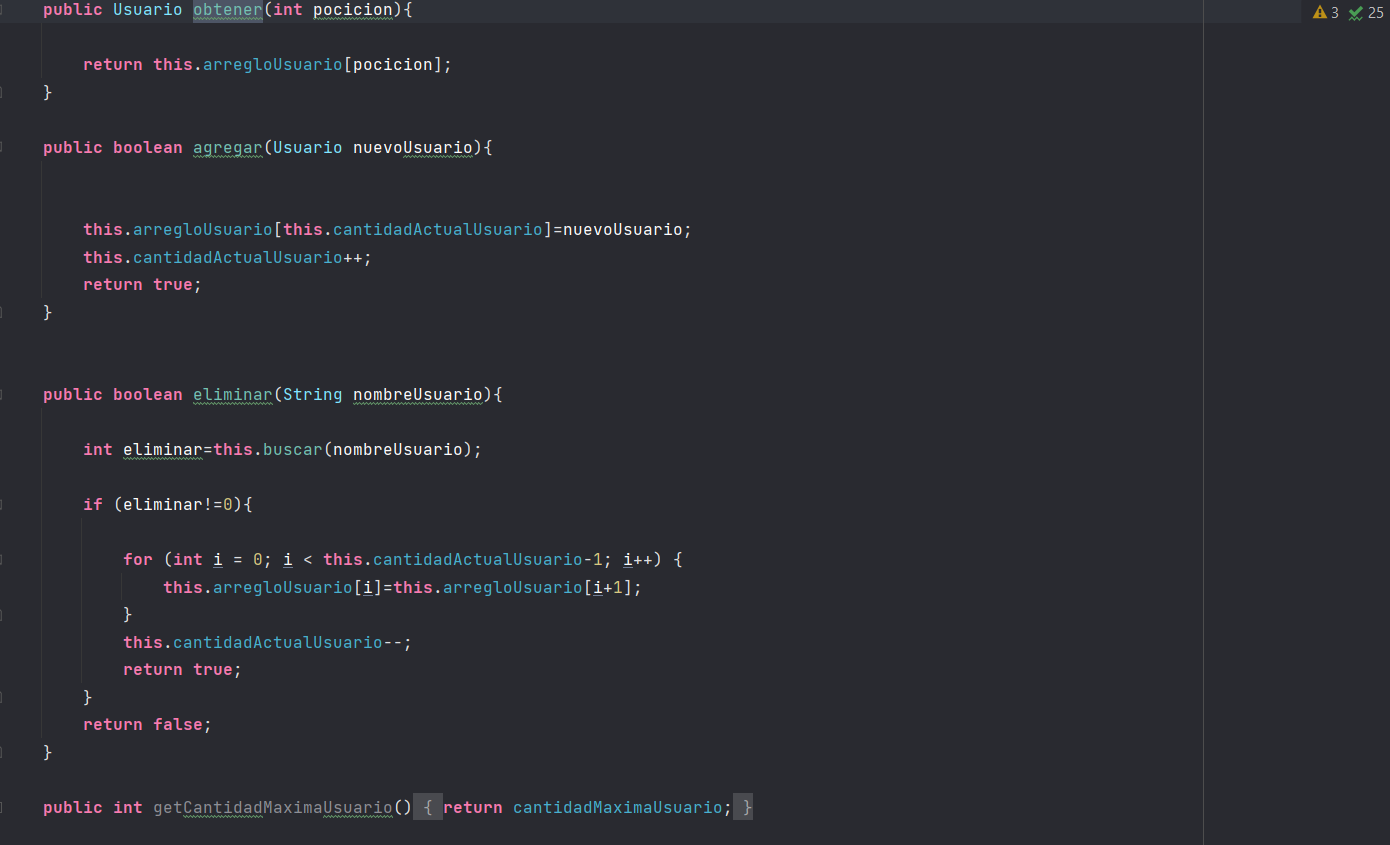


Clase Usuario: Aquí tendremos la clase usuario la cuan tendrá los parámetros como el nombre de usuario y la contraseña las cuales nos servirán para poder almacenarlos después en una lista Usuarios aquí encontrara también sus métodos get y set junto a su constructor.

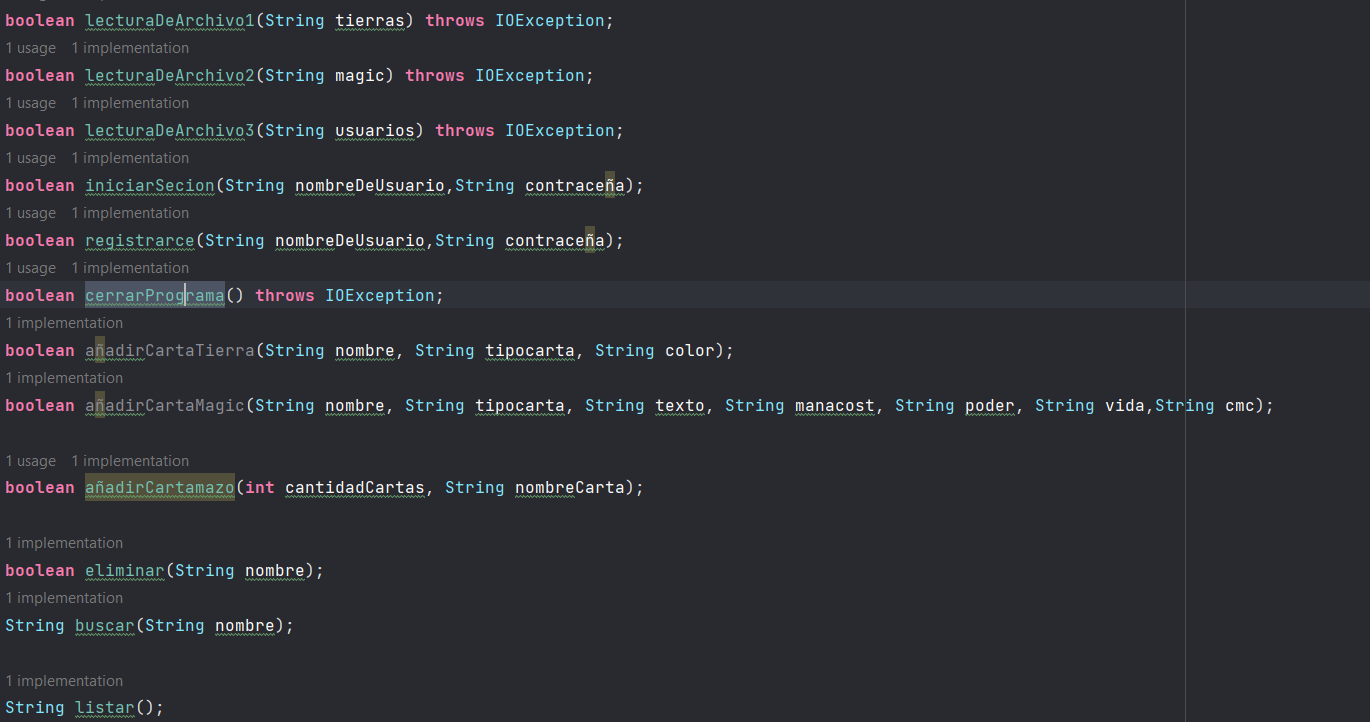


Lista Usuarios: En esta clase definiremos los parámetros de una lista normal como buscar, obtener, eliminar y agregar para utilizarlos en caso de ser necesario también están sus atributos privados como cantidad máxima, cantidad Actual haciendo llamado a la clase usuario para definir el arreglo.



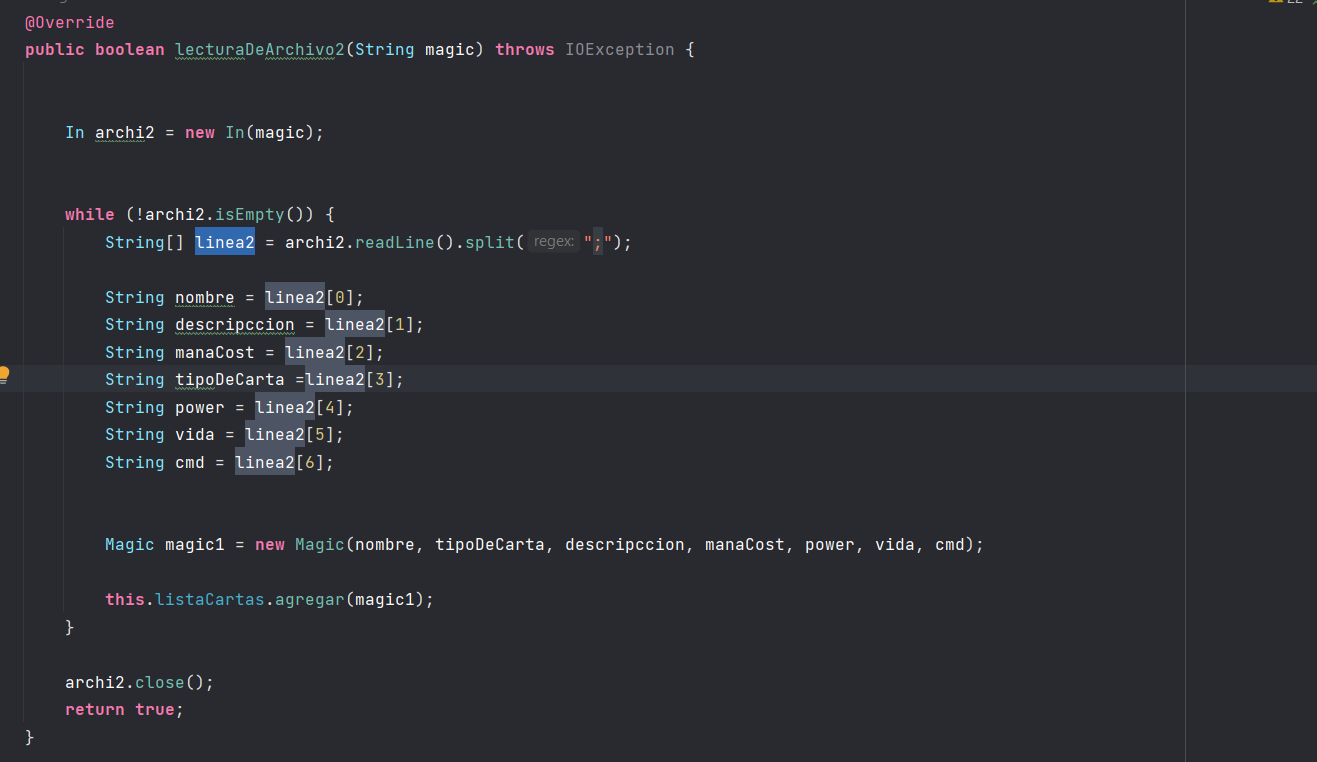


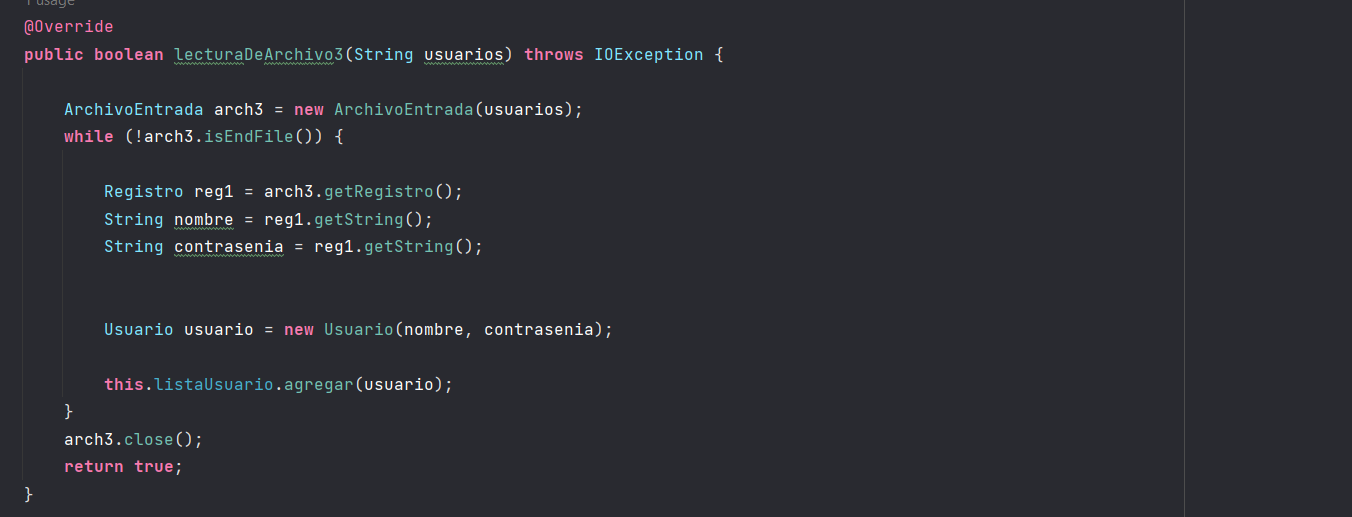
ISistema Magic: aquí tendremos la interface y los subprogramas que se irán extendiendo hacia la clase de Sistema la cual nos permitirá ejecutar todo los procesos que nos solicitan por el taller.



SistemaMagic: aquí encontraremos los métodos de junto a las instrucciones que le podremos dar, los primeros 3 sub programas serán de lectura de archivos. La primera para leer las cartas tierras, la segunda para leer las cartas Magic y la tercera para leer las de la lista Usuarios que son las que se irán guardando para almacenar los nuevos registros de usuarios que irán añadiendo.



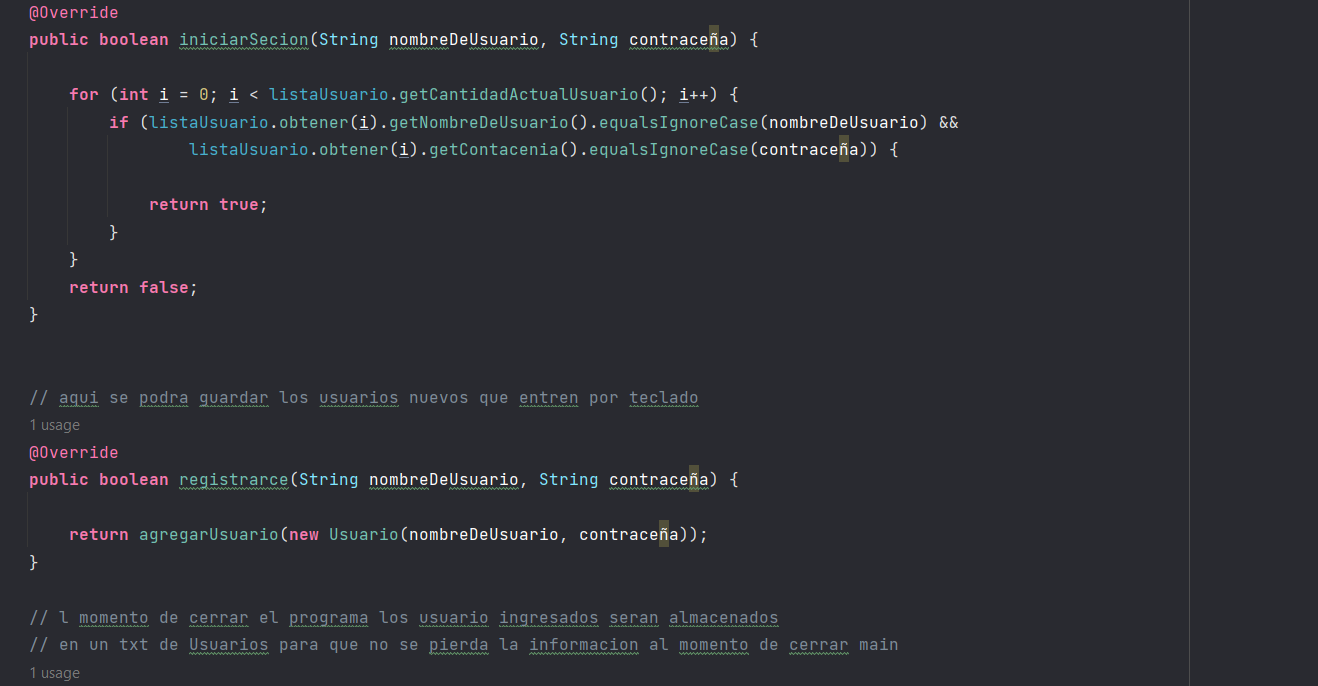


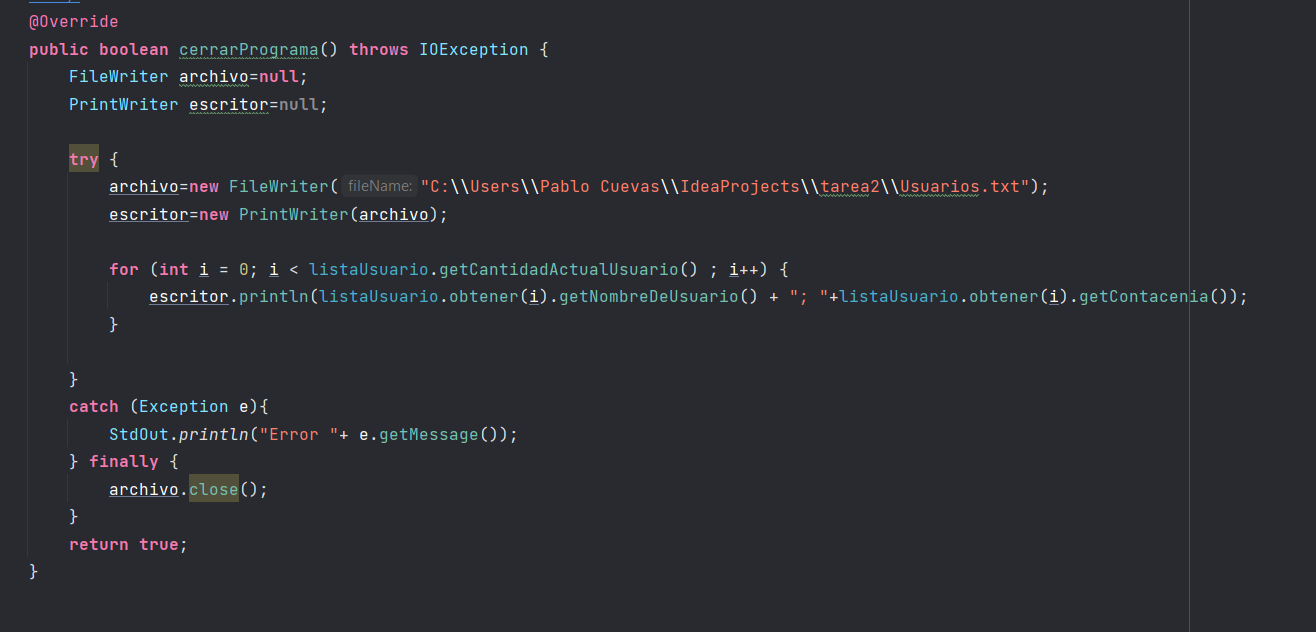


Las imágenes de a continuación se verán los sub programas de iniciar sección el cual va a comparar si en la lista usuario hay en la posición un nombre y contraseña igual a lo que se está pidiendo por pantalla.

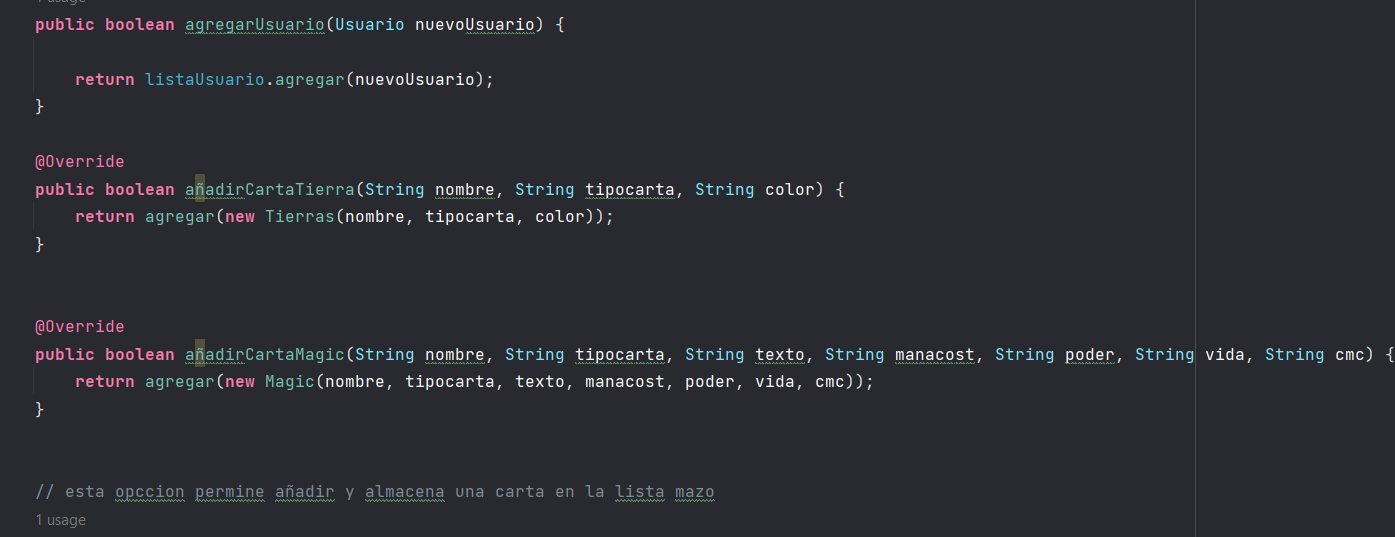
El sub programa de registrarse aquí se apreciará la forma de almacenar a través del lo que se pregunte en el main un nuevo usuario que será ingresado manualmente por el usuario con su nombre Usuario y contraseña.

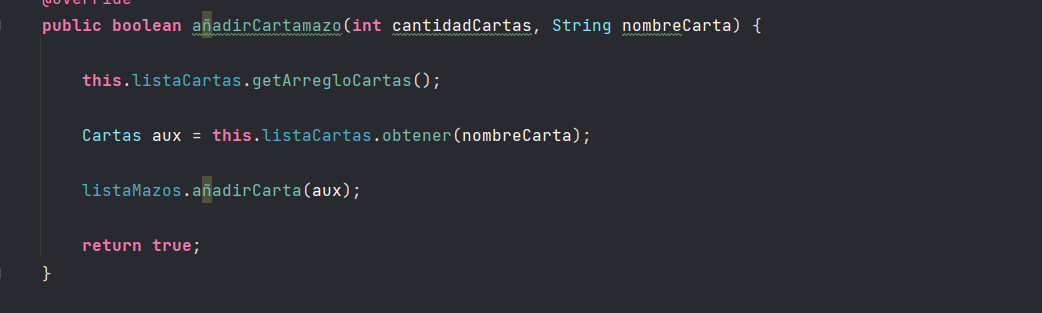
El sub programa de cerrar sesión este hecho para tirar un archivo de salida llamada Usuarios.txt el cual contendrá los Usuarios ingresados anteriormente, para que al momento de reiniciar el programa o cerrar el programa estos estén guardados y no se olviden dentro del ejercicio. Requisito previo tener un Usuarios.txt y que la dirección de guardado sea acorde al computador ejercido.

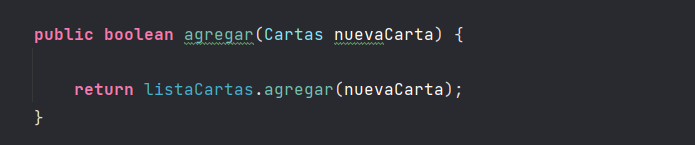




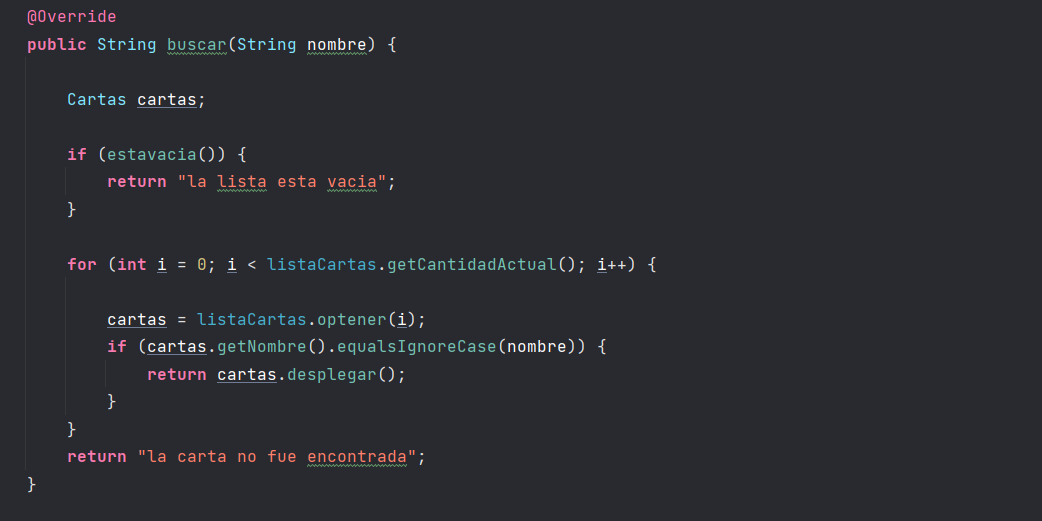
Después encontramos los métodos de añadir que se encargaran de añadir una carta al mazo o añadir mas cartas al propio txt y añadir a los usuarios a las listas .

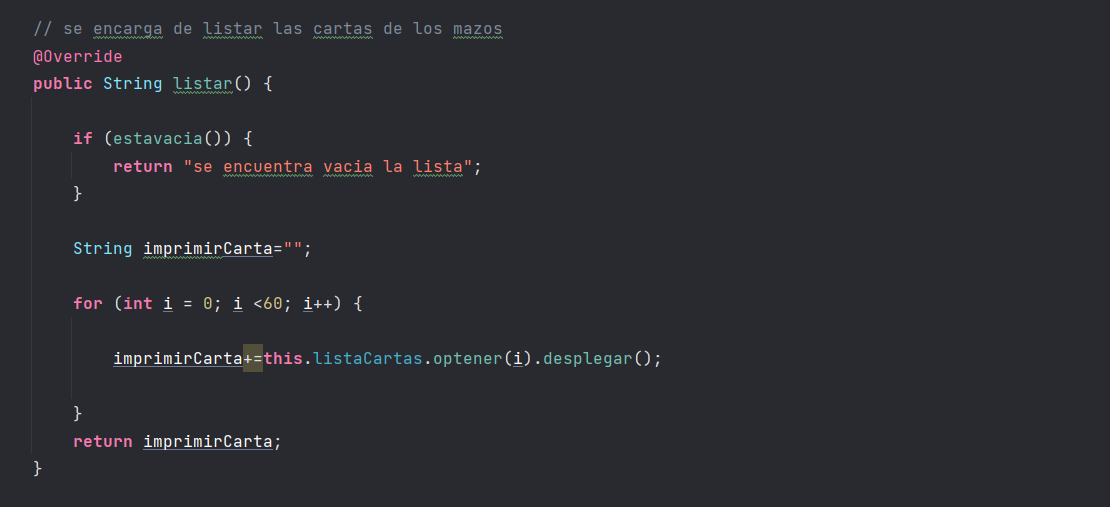






Por último, tendremos los 2 subprogramas de buscar y listar los cuales como su nombre dice se encargaran de buscar la carta deseada en la lista carta y de listar las cartas solicitadas.





Por ultimo tendremos el instalador que se encarga de comunicar todo lo del sistema con el main.