TALLER PROGRAMACION AVANZADA: ENTREGA FINAL

Integrantes: Juan Almonte ,Sean Castillo.

Correo Electronico:

juan.almonte@alumnos.ucn.cl
sean.castillo@alumnos.ucn.cl

Rut:22.168.387-0,20.542.721-K

Paralelo:C1.

Profesor: Tomas Reiman.

Fecha:29 de mayo del 2024.

Índice

Introducción:

Durante el desarrollo del proyecto gestor de mazos para Magic the Gathering. Se decidió confeccionar un diagrama de clases para la organización y eficiencia del código, sin embargo, a medida que se confeccionaban los procesos y métodos detallados en el diseño se presentó la problemática, los métodos y lógica propuesta era ineficiente e innecesaria en bastantes aspectos derivando a implementar nuevos métodos y clases al programa.

Diagrama de clases 1

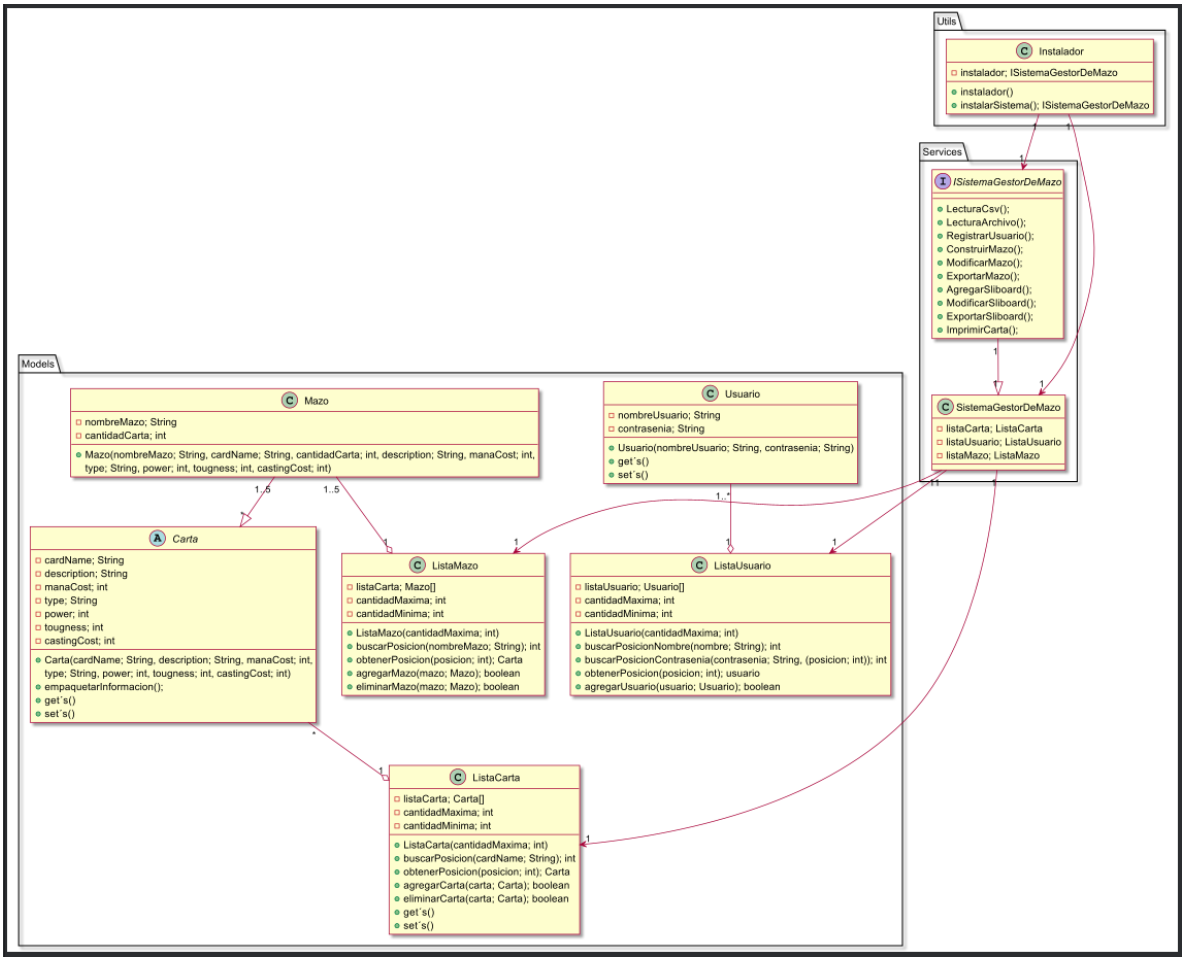

Diagrama de clases 2 (Era bastante grande para adjuntarlo)

Los cambios significativos observables en el programa son:

Adición de clases Tierra, Tierra2 y ListaTierra. Con el fin de separar el tipo de cartas de las cartas normales, ya que, las tierras no poseen restricciones en base a la cantidad. Permitiendo una gestión mucho más eficaz de los datos dentro de la clase ListaMazo.

Adición de métodos:

Los métodos planteados en el anterior diagrama de clases eran insuficientes para la lógica del programa y la resolución de la problemática. Se decidió implementar métodos relacionados con la comprobación, adición, eliminación y modificación de datos para las clases listas.

Consideraciones:

Durante el desarrollo de la lógica del programa se planteó el uso de la abstracción para las clases con el fin de facilitar el almacenamiento de los atributos, sin embargo, demostró ser ineficiente e innecesario para la lógica mayoritaria del programa. Aun así, se optó por mantenerlas a fin de que el código no sufriera percances respectos a las modificaciones implementadas por las mismas.

Clases empleadas en el programa

Clases Empleadas para la lógica del programa	
Carta-ListaCarta	Encargada de almacenar los atributos de las cartas
Tierra-ListaTierra	Encargada de almacenar los atributos de las tierras
Usuario-ListaUsuario	Encargada de almacenar los usuarios ingresados por parámetro
Mazo-ListaMazo	Encargada de gestionar y almacenar los mazos y sidecks generados por el usuario
Sidedeck	Encargada de almacenar los "sidecks" generados por el usuario, "Extiende de mazo"

Clase Carta-ListaCarta:

La clase Carta y ListaCarta, poseen la finalidad de almacenar atributos obtenidos de la lectura de archivos del método "lecturaListasCartas" cual se ahondará próximamente. Cabe añadir que la clase Carta es abstracta por decisiones de lógica anterior, cuales no se consideraron oportunas, por ende, se implementó la clase Carta2 a fin de asignar atributos a la clase y poder operar con ellos.

Clase Tierra-ListaTierra:

La clase Tierra y ListaTierra, poseen la finalidad de almacenar atributos obtenidos de la lectura de archivos del método "lecturaListaTierras" cual se ahondará próximamente. Cabe añadir que la clase tierra es abstracta por decisiones de lógica anterior, cuales no se consideraron oportunas, por ende, se implementó la clase Tierra2 a fin de asignar atributos a la clase y poder operar con ellos.

Clase Usuario-ListaUsuario:

La clase Usuario permite el almacenamiento de datos ingresados por parámetro (Nombre de usuario y contraseña).

ListaUsuario gestiona y entrega las credenciales junto al mazo correspondiente relacionado con el usuario, considerando su existencia dentro de los datos del programa.

Mazo-ListaMazo:

La clase Mazo almacena los datos ingresado por parámetro (Nombre de la carta, Cantidad de la carta y el usuario actual).

ListaMazo gestiona, modifica y entrega datos de los mazos creados por el usuario actual con una capacidad máxima de 60 cartas con el fin de utilizar los datos en los distintos métodos próximos.

Sidedeck:

La clase Sidedeck, como su nombre dice es el mazo de al lado. Extiende de la clase mazo, por ende, cumple con la misma lógica respecto al almacenamiento de datos ingresados por parámetro, sin embargo, su capacidad dentro de la clase ListaMazo es de 15 cartas.

Observaciones:

Durante la confección del respectivo programa surgieron distintos desafíos, retos y contratiempos tales como:

La implementación de un nuevo paradigma de programación en base a criterios anteriores podía llegar a complicar el entendimiento de la lógica base del programa.

Las relaciones entre clases padre eh hijas llegaba a causar problemas, porque la comunicación entre ambas no era la esperada.

Durante la elaboración del taller a causa de la carencia de computadores propios y portables se tuvo que trabajar en distintos equipos llevando a varios problemas con IntelliJ. conllevando a la transcripción de taller en mas de una ocasión a un nuevo archivo.

La gestión de tiempo del tiempo fue un reto respecto a que Juan Almonte cayó enfermo por un tiempo prolongado. Ocasionando que Sean castillo tuviera que dedicar más tiempo a la confección del taller.

Uno de los principales retos durante la confección del taller fue la implementación del SideBoard y métodos generales para la administración de los mazos dentro del programa.

Flujo de programa:

El programa inicia solicitando por pantalla al usuario ingresar sus credenciales, si no existen datos previos no se permitirá la continuación del mismo y se exigirá al usuario registrarse.

Consecuentemente el programa despliega variedad de opciones para que el usuario decida operar:

- Construir mazo
- Ver mis mazos
- Buscar carta
- Salir

Dependiendo de la opción se despliega un submenú.

En el primer caso el usuario puede crear los 4 mazos juntos a un sidedeck añadiendo una carta al mazo o en su defecto modificarlos. Cabe añadir que únicamente se le permitirá al usuario crear mazos si estos son inexistentes. De no ser el caso únicamente se le permitirá modificarlos.

En el segundo caso el usuario puede ver los mazos. Por medio de una impresión por pantalla. Se desplegarán las cartas que lo componen independiente si el mazo está completo o no, sin embargo, si el mazo no existe se desplegara un mensaje señalado la inexistencia de estos.

En el tercer caso el usuario puede buscar una tierra o carta especifica mediante el nombre de la misma.

En el último caso el usuario sale del programa de nuevo al inicio de sesión permitiendo a otro usuario ingresar y modificar o crear sus propios mazos. Cabe añadir que si el usuario creo mazos estos serán exportados junto a las credenciales del usuario en un archivo “txt” permitiendo almacenarlos y al usuario volver a acceder a sus mazos.

Métodos empleados en el programa

Métodos Empleados para la lógica del programa	
registrarUsuario	Permite el registro del usuario, almacenando los datos ingresado por parámetro en Usuario-ListaUsuario.
iniciarSesion	Permite al usuario ingresar al programa, siempre cuando existan sus credenciales dentro del mismo.
lecturaListaCartas	Permite la lectura de un archivo de extensión “txt” y almacena los datos leídos en Carta-ListaCarta.
lecturaListaTierras	Permite la lectura de un archivo de extensión “txt” y almacena los datos leídos en Tierra-ListaTierra.
agregarSliboard	Crea un sidedeck dentro de la lista mazo
eliminarCartaSideck	Elimina una carta ingresada por parámetro en una posición dentro de la clase ListaMazo respecto al sidedeck
imprimirCarta	Realiza la búsqueda con datos ingresados por parámetro e imprime por pantalla una carta en una posición especifica dentro de la clase ListaCarta.
imprimirTierra	Realiza la búsqueda con datos ingresados por parámetro e imprime por pantalla una carta en una posición especifica dentro de la clase ListaTierra.
agregarNuevaCarta	Agrega una nueva carta a un mazo dentro de una posición en la clase ListaMazo.
eliminarCarta	Elimina una carta o tierra respecto al mazo dentro de una posición en la clase ListaMazo.

comprobarListaModificada	Verifica si los mazos fueron modificados dentro del programa.
comprobarCreacionMazo	Verifica la creación del mazo dentro del programa.
seleccionarMazo	Verifica si el mazo existe previamente dentro del programa. Permite la selección de un mazo para distintas opciones.
procesarCarta	Permite trabajar con la carta con el fin de modificar la cantidad dentro del mazo o agregarla.
comprobarCarta	Comprobación de la cantidad máxima posible para la modificación o adición de las cartas a los mazos (1-4). A excepción de tierras.
comprobarCantidadSidedeck	Comprobación de la cantidad máxima posible de cartas dentro del sidedeck. (Max: 15)
comprobarCantidad	Comprobación de la cantidad máxima posible de cartas dentro del mazo. (Max: 60)
mostrarCarta	Imprime por pantalla todas las tierras o cartas dependiendo de la selección del usuario.
mostrarMazos	Imprime por pantalla los datos del mazo o sideck seleccionado por el usuario
guardarRegistroUsuario	Exporta los datos de ListaUsuario y Usuario a un documento de formato “txt”
exportarMazo	Exporta los datos de la ListaMazo y Mazo a un documento de formato “txt”

Conclusión:

El desarrollo del programa fue desafiante, la implementación de las nuevas clases fue necesarias para el cumplimiento de la mayoría de objetivos, los métodos añadidos permitieron una gestión más eficaz y gestión optima respecto a lo solicitado por la problemática.