

Informe de la implementación de un Mazo para “La Guerra de Cartas”

UNIVERSIDAD NACIONAL DE ENTRE RÍOS

Facultad de Ingeniería

Carrera: T.U.P.E.D

Materia: Fundamentos de Algoritmos y Estructuras de Datos

Estudiante: Franco Tomiozzo

Profesor: Dr. Javier E. Diaz Zamboni

Bioing. Jordán F. Insfrán

Esp. Bioing. Francisco Rizzato

Comisión: 1

Fecha: Septiembre 2025

Introducción

El informe va a describir y analizar la implementación de la estructura del nuevo programa llamado “Mazo”, desarrollada para completar un juegos de cartas.

El trabajo consistió en diseñar un programa que represente un mazo de cartas y permita realizar operaciones fundamentales para saber que cartas se utilizan y como conlleva el juego solo observando de otro punto de vista.

Los docentes nos proporcionaron las clases de las cartas y la lógica del juego y la tarea es que en mediante de un tests que sería el juego, nuestro programa funcionará correctamente

Desarrollo

La clase (**DequeEmptyError**), está destinada para manejar el error que se produce al intentar extraer una carta de un mazo que está vacío.

Lo que evita fallos inesperados y facilita la depuración de errores, haciendo el programa más seguro.

Cada uno de los elementos del mazo se representa mediante un nodo doblemente enlazado.

De esta manera, se construye una lista doble que permite recorrer el mazo y modificarlo con eficiencia.

La clase “Mazo”

Esta clase gestiona la baraja de cartas.

Sus principales funciones son:

- Una Inicialización (**__init__**): Crea un mazo vacío, sin cartas, y define referencias iniciales a su cabeza, cola y tamaño.
- Poner la carta arriba (**poner_carta_arriba**): Agrega una nueva carta en la parte superior del mazo.
- Poner la carta abajo (**poner_carta_abajo**): Agrega una carta en la parte inferior del mazo.
- Sacar la carta de arriba (**sacar_carta_arriba**): Extrae la carta que se encuentra en la parte superior. Si se indica, la carta puede mostrarse como visible. En caso de que no existan cartas, se lanza la excepción personalizada.
- Consultar el tamaño de cartas (**__len__**): Devuelve el número de cartas contenidas en el mazo.
- Representación en texto (**__str__**): Permite ver el contenido del mazo en formato de lista, facilitando la comprensión del estado interno de la estructura con la vista de las cartas.
- Verificar si está vacío (**esta_vacio**): Retorna un valor booleano que indica si el mazo hay o no cartas.

Conclusión

La implementación del nuevo programa desarrollado cumple con los objetivos de construir un mazo de cartas confiable, flexible y seguro, solamente utilizando una estructura de lista doblemente enlazada.

El uso de una clase que controla el error dentro del programa da una mejora y seguridad.

En conclusión, el trabajo realizado con una parte fundamental para la integración con el juego completo dados por los profesores, y al soporte de pruebas automatizadas que garantiza su correcto funcionamiento del programa.