BIOINGENIERÍA

Algoritmos y Estructuras de Datos

Trabajo práctico N°1:
Aplicaciones de TADs: Problema 2

Autora:

Dinamarca Daiana

24/10/2025

Problema 2

El objetivo de este ejercicio fue implementar el **juego de cartas** conocido como **Guerra**, utilizando la estructura de datos desarrollada en el Problema 1 (Lista Doblemente Enlazada) como base para representar los mazos de cada jugador. La consigna requería diseñar una clase que modelara el comportamiento del mazo, las operaciones de extracción y agregado de cartas, y otra clase para el desarrollo del juego completo, incluyendo la comparación de cartas y la resolución de empates.

El proyecto se organizó en dos clases principales:

- Mazo, encargada de almacenar y manipular las cartas mediante una lista doblemente enlazada.
- JuegoGuerra, que gestiona la lógica del juego, los turnos, las comparaciones entre cartas y la definición del ganador.

Dentro de Mazo, se implementaron métodos como agregar_carta_abajo(), sacar_carta_arriba(), mezclar(), esta_vacio() y cantidad_cartas(). Inicialmente, la implementación de sacar_carta_arriba() no contemplaba parámetros adicionales, lo que produjo un error durante las pruebas del módulo JuegoGuerra, ya que la llamada se realizaba con el argumento mostrar=True. Este inconveniente fue resuelto modificando la definición del método para aceptar el parámetro opcional mostrar=False y, en caso de ser True, imprimir la carta extraída.

Durante el desarrollo se presentaron algunas dificultades relacionadas con la estructura de carpetas e importaciones entre proyectos. En particular, el proyecto P2_JuegoCartasGuerra dependía del módulo ListaDobleEnlazada del Problema 1, lo que generó errores del tipo ModuleNotFoundError: No module named 'P1_ListaDobleEnlazada' al ejecutar los tests.

Se probaron distintas estrategias para resolver este problema: ejecutar los tests desde distintos niveles de la terminal, ajustar los archivos __init__.py para que Python reconociera las carpetas como paquetes. Finalmente, se optó por copiar el archivo ListaDobleEnlazada.py dentro de la carpeta modulos del Problema 2, garantizando así que todas las dependencias estuvieran contenidas en un único directorio funcional. Esto permitió que las pruebas unitarias se ejecutaran correctamente tanto desde la terminal de Visual Studio Code como mediante el botón “Run” del entorno.

Una vez solucionados los problemas de importación, se verificó el correcto funcionamiento de los métodos y la lógica del juego mediante las pruebas unitarias provistas (test_mazo.py y test_juego_guerra.py). Las tres pruebas principales evaluaron partidas en las que gana el jugador 1, el jugador 2 y en las que se produce un empate.

Tras la corrección del método sacar_carta_arriba, todas las pruebas se ejecutaron con éxito, confirmando la correcta interacción entre los objetos Mazo y JuegoGuerra y la validez de la simulación completa del juego.

Conclusión

El desarrollo del Problema 2 permitió integrar la estructura implementada en el Problema 1 dentro de un contexto más complejo, demostrando su funcionalidad y versatilidad. A lo largo del proceso se abordaron desafíos vinculados al manejo de módulos, rutas de importación y consistencia de métodos, los cuales fueron resueltos de manera progresiva hasta lograr un proyecto estable y completamente funcional.

Los resultados obtenidos en las pruebas unitarias confirman que tanto la gestión de los mazos

como la dinámica del juego se comportan según lo esperado, cumpliendo con los requisitos de la consigna y asegurando la reutilización efectiva de la Lista Doblemente Enlazada desarrollada previamente.