

## Trabajo Práctico N° 1

### Informe Problema 3

Para resolver este problema implementamos una clase Mazo utilizando como estructura interna una lista doblemente enlazada (LDE), desarrollada en el problema anterior.

El objetivo fue simular el desarrollo del juego, respetando las reglas planteadas: enfrentamientos por turnos, acumulación de cartas por parte del ganador, y resolución de empates mediante guerras sucesivas. Para lo que la clase Mazo debía gestionar correctamente las operaciones básicas y contemplar situaciones límite, como intentar extraer cartas de un mazo vacío.

La clase implementa los siguientes métodos fundamentales:

- `poner_carta_abajo(carta)`: Inserta una carta al final del mazo. Utiliza el método `agregar_al_final` de la LDE.
- `poner_carta_arriba(carta)`: Inserta una carta al principio del mazo. Usa `agregar_al_inicio`.
- `sacar_carta_abajo()`: Extrae y retorna la última carta del mazo. Lanza `DequeEmptyError` si el mazo está vacío.
- `sacar_carta_arriba()`: Extrae la carta superior del mazo (inicio de la lista). También lanza `DequeEmptyError` si no hay cartas disponibles.
- `__len__()`: Retorna la cantidad de cartas. Aprovecha el contador de la LDE para garantizar complejidad  $O(1)$ .
- `__str__()`: Devuelve una representación en cadena del contenido del mazo.

El método `esta_vacia()` también fue utilizado internamente para verificar si el mazo posee cartas antes de realizar operaciones destructivas.

#### Validación y análisis

Los tests provistos por la cátedra se ejecutaron de forma exitosa sin necesidad de modificar sus estructuras, lo que confirma que las firmas de los métodos y su funcionalidad coinciden exactamente con lo requerido.

El comportamiento del juego se reprodujo correctamente, incluyendo:

- Flujo normal de turnos.
- Resolución de guerras consecutivas.
- Manejo de mazos vacíos mediante excepción controlada (`DequeEmptyError`).

No se midieron tiempos de ejecución en este caso, dado que el foco estaba puesto en la correcta funcionalidad del juego más que en el rendimiento.