

Problema 3

En el problema 3 se implementó el TAD mazo como clase de Python con las distintas funciones, aprovechando la clase ListaDoblementeEnlazada creada previamente. A continuación se detalla la función y su correspondiente metodología:

- `poner_carta_arriba(self, carta)`: Se utiliza la función `agregar_al_inicio(self, dato)` de listas doblemente enlazadas con una carta ingresada a modo de dato.
- `sacar_carta_arriba(self, mostrar=False)`: Se utiliza la función `extraer(self, posicion)`, con la posicion fijada en cero, es decir la carta inicial. El parámetro “mostrar” debe ser “False” inicialmente, y se mostrará la carta extraída en caso de que se indique como “True” durante el juego.
- `poner_carta_abajo(self, carta)`: Se utiliza la función `agregar_al_final(self, dato)` de listas doblemente enlazadas con una carta ingresada a modo de dato.
- `__len__(self)`: De igual manera que en el problema 2, se sobrecarga el método para poder implementarse sobre el mazo.

Además, se creó la clase `DequeEmptyError`. La misma se despliega en caso de que se intente sacar una carta de un mazo sin elementos.