

Ejercicio 2 Taller 2 Python

- a) 3: Loteria, ComisionJuegoEspectaculos y Apostador
- b) Lo que esté bajo este condicional será ejecutado solo si se está ejecutando el archivo directamente, pero será omitido si se importa para ser utilizado en otro script
- c) En este caso no ocurre nada si se elimina la instrucción `if __name__ == "__main__":`, ya que este es el único archivo que contiene instrucciones donde se declaran e inicializan objetos, así que es el único archivo que al ejecutarse mostrará algo por consola, por lo que de por sí se debe ejecutar directamente
- d) Se crean 2 objetos de la clase Apostador
- e) Son: `apostador1` y `apostador2`
- f) Hace referencia a `apostador1` y `apostador2` al invocar al método `play`, ya que en este sucede la creación de un objeto `Lotería` que recibe como parámetros un `value` y un objeto `Apostador`, siendo este último al que se refiere al utilizar `self`, el objeto mediante el cual se invoca el método `play`
- g) Se crean dos objetos `Lotería`, uno cuando `apostador1` invoca al método `play` otro cuando `apostador 2` hace lo mismo
- h) Imprimiría: 300 Necesitas poner más dinero en tu wallet 300
- i) Imprimiría: 400, hay una posibilidad de que imprima "Has ganado 720.0 720.0" o "Has perdido lo que apostaste"
- j) Únicamente `apostador`
- k) `Probability` y `value`
- l) `@classmethod`, `cls`, `cls`
- m) `Loteria.changeProbability(numero)`
- n) En este caso no afectaría a la ejecución del programa, pero es recomendado usar `cls`
- o) 4 contando el último añadido
- p) No, si `apostador1` o `apostador2` ganan es decidido de manera aleatoria
- q) No se consideraría correcto. Si bien Python permite hacer este cambio ya que no existen las constantes, al ser declarada como constante por el programador, esta debe ser tratada como tal por toda la duración del programa
- r) Los dos retornan un número de tipo `float`
- s) Hace referencia al objeto `Lotería` creado cuando el objeto `apstador1` o `apostador 2` invoca el método `play`. No se puede omitir, ya que se está creando un objeto de clase `ComisionJuegoEspectaculos` que recibe como parámetro un objeto `Lotería`, que lógicamente debe ser el mismo objeto que invocó al método `playGame` responsable de la creación del objeto de clase `ComisionJuegoEspectaculo`
- t) `Value` es pasado por valor ya que es un tipo primitivo representando la cantidad de dinero que se desea apostar. `Self` se pasa por referencia ya

que se pasa un objeto de clase Apostador, el cual es un tipo de dato compuesto