

Proyecto 1

Fecha de entrega: Lunes 22 de mayo, 2023

Instrucciones generales

El código de la tarea se debe subir en GitHub, debe crear una carpeta específica para la entrega. Debe estar tanto el código como la documentación en la misma carpeta.

La fecha de entrega es el 22 de mayo a las 23:59. En caso de atraso, se rebajará 10% de la nota final por cada día de atraso hasta llegar a 3. Después de que pasen 3 días de la fecha de entrega, la plataforma no aceptará más entregas.

Parte 1 (50%). Desarrollo de aplicación

Debe crear una aplicación que permita jugar el juego de cartas conocido como *blackjack* o 21. La aplicación debe cumplir con los siguientes requerimientos:

1. Permitir jugar una partida completa de *blackjack* de un jugador contra la computadora. Para la partida se utilizará un único mazo (*deck*) de 52 cartas. Referencia para las reglas del juego: <https://www.casino.org/es/blackjack/como-jugar/>
2. El juego se debe desarrollar en su totalidad en la consola o terminal. En cada turno se deben desplegar lo siguientes datos:
 - a. Mano de la casa o *dealer*: se deben desplegar las cartas que tiene la casa, una escondida y las demás abiertas. Usted puede decidir como representar las cartas, solamente debe utilizar los símbolos *unicode* de cada palo (bastos, diamantes u oros, flores o cruces y corazones).
 - b. Mano del jugador: mismos requerimientos del punto anterior. Todas las cartas abiertas.
 - c. Estado de cada jugador: se debe desplegar si un jugador sigue en juego, si ganó o perdió.
 - d. Estado del juego: Se debe desplegar el estado general del juego: en juego o terminado, por ejemplo.
3. El juego debe tener un menú en el que existan por lo menos las siguientes opciones. Puede crear otras opciones si lo considera necesario:
 - a. Al inicio del juego se debe seleccionar un usuario o crear uno nuevo.
 - b. Nuevo juego: inicia un juego nuevo con el usuario seleccionado.
 - c. Estadísticas: muestra las estadísticas del usuario seleccionado. Estas estadísticas deben ser por lo menos los resultados de los últimos 5 juegos del usuario.
 - d. Salir.
4. Cuando se inicia un nuevo juego, la aplicación debe llevar al usuario por todas las etapas del juego. Esto incluye la repartición inicial de cartas, que el o los usuarios (1 o 2) pidan más cartas o paren y que la casa juegue.

5. La casa, al ser manejada por la computadora, debe seguir una lógica predeterminada, por ejemplo que siempre pida carta si su mano suma menos de 19.
6. Debe implementar algunas de las manos especiales del 21: tres sietes, cinco menores y *blackjack*. Si un jugador tiene alguna de estas manos, gana automáticamente.
7. El juego debe almacenar los datos del usuario y las estadísticas en archivos de texto que serán creados por la aplicación. Usted puede decidir cuántos archivos de texto crear para lograr los resultados especificados en estos requerimientos.
8. El código debe estar organizado en módulos y en clases. Se deben utilizar los conocimientos adquiridos sobre programación orientada a objetos (OOP). Usted debe decidir cuáles clases necesita crear para representar los elementos del juego. Referencia sobre diseño orientado a objetos: https://en.wikipedia.org/wiki/Object-oriented_design

Parte 2 (50%) Documentación interna y externa

La segunda parte de la tarea consiste en crear documentación interna del proyecto (a través de comentarios) y documentación externa en un archivo de *markdown*. Referencia sobre tipos de documentación: <http://www.cs.ecu.edu/karl/3300/spr16/Notes/Documentation/documentation.html#:~:text=External%20documentation%20is%20written%20in,for%20users%20of%20an%20application>.

La documentación interna debe ser clara y concisa, solamente debe describir qué hacen los diferentes bloques de su código. La documentación externa debe explicar con detalle el proceso de diseño así como la implementación. Aquí un documento para leer más al respecto: <https://document360.com/blog/technical-documentation/>

La documentación externa se debe hacer en *markdown*. Es decir, debe incluir en su repositorio de GitHub uno o varios archivos .md. Aquí una referencia al *markdown*: <https://www.markdownguide.org/basic-syntax/#html>

Rúbrica

Elemento	Valor
Parte 1: Implementación	50%
<i>Funcionamiento del juego (desarrollo e interfaz de usuario)</i>	25%
<i>Creación de usuarios y estadísticas (almacenamiento en archivos, lectura y escritura de datos)</i>	15%
<i>Organización del código en funciones, módulos y clases (representación de elementos de la vida real con objetos, creación de módulos e importación correcta de los mismos)</i>	10%
Parte 2: Documentación	50%
<i>Documentación interna (suficientes comentarios, clara y concisa)</i>	25%
<i>Documentación externa (utilización de correcta de markdown, documentación completa del proceso de diseño e implementación)</i>	25%