

# MEMORIA PROYECTO: CASINO VIRTUAL

---

## BRIEFING

### OBJETIVOS

El objetivo principal de este proyecto es desarrollar un programa completo que simule un entorno básico de casino en la terminal, donde el jugador puede participar en tres juegos clásicos: Ruleta, Carta Alta y Blackjack. Este casino virtual permite gestionar un sistema de saldo para el jugador, realizar apuestas y obtener ganancias o pérdidas dependiendo del resultado del juego.

El programa está pensado para ser modular, intuitivo y visualmente atractivo, a pesar de estar basado en consola, gracias al uso del módulo `rich`, que permite una experiencia mejorada mediante colores, tablas y elementos visuales. Esta herramienta también facilita la comprensión de las instrucciones y el seguimiento de las partidas.

### FUNCIONALIDADES

- **GESTIÓN DE SALDO:** El usuario dispone de funciones para ingresar y retirar saldo, consultar su estado actual y evitar que se quede sin fichas.
- **RUETA:** El jugador apuesta a dos números del 0 al 36. Si acierta uno, gana 1.5 veces su apuesta; si acierta ambos, gana el doble. En caso de no acertar, pierde lo apostado.
- **CARTA ALTA:** Se reparten dos cartas (una al jugador y otra al crupier). Si la carta del jugador es superior, gana el doble de la apuesta; si es inferior, pierde.
- **BLACKJACK:** Se enfrentan jugador y crupier. Cada uno recibe cartas y el objetivo es sumar lo más cerca posible a 21 sin pasarse. Gana el que esté más cerca, o pierde quien se pase.
- **INTERFAZ VISUAL EN TERMINAL:** Toda la interfaz ha sido adaptada con `rich`, lo que mejora la comprensión del juego, embellece el menú y permite resaltar resultados e instrucciones con estilos y colores.

## CONCEPTO

### TÍTULO: CASINO ROYALE

El proyecto surge como una propuesta de simulación de juegos de azar con un enfoque pedagógico. Se pretende que el usuario, además de entretenerse, practique estructuras de decisión, ciclos, entrada y salida

de datos, control de errores, gestión modular de código y principios básicos de orientación a objetos.

Se ha diseñado como un sistema autónomo y completo, estructurado en varios archivos que cooperan entre sí. El código es fácil de mantener y ampliar, permitiendo integrar nuevos juegos o mecánicas adicionales si se desea.

## **DESCRIPCIÓN GENERAL**

El programa se inicia solicitando al usuario que se registre. A partir de ahí se le presenta un menú principal con diferentes opciones: jugar, consultar o modificar su saldo, y acceder a las instrucciones de los juegos. Desde el menú de juegos, puede elegir entre Ruleta, Carta Alta o Blackjack. Cada juego está encapsulado en una clase propia, con sus reglas específicas y control de saldo. Una vez finalizado un juego, el usuario vuelve al menú principal hasta que decida salir.

## **FUNCIONALIDADES DEL PROGRAMA**

### **GESTIÓN DE SALDO**

Se han implementado funciones de depósito, retiro y consulta de saldo. El saldo se guarda en el objeto `Jugador`, y se modifica directamente en cada interacción. Se validan entradas para evitar errores comunes o saldo negativo.

### **JUEGO DE RULETA**

El jugador elige dos números entre 0 y 36. La ruleta simula una animación con números aleatorios antes de detenerse en dos números. Dependiendo del resultado, se calcula el premio (ninguno, 1.5x o 2x) y se actualiza el saldo.

### **JUEGO DE CARTA ALTA**

Se genera una carta para el jugador y otra para el crupier. Si la del jugador es mayor, gana el doble. Es un juego rápido y sencillo que permite partidas ágiles.

### **JUEGO DE BLACKJACK**

Es el juego más elaborado. El jugador puede pedir cartas o plantarse, y el crupier juega automáticamente hasta alcanzar 17 o más. Se calculan los puntos y se decide el resultado según las reglas del blackjack tradicional, pero simplificado.

### **INSTRUCCIONES EN PANTALLA**

Gracias al uso del módulo `rich`, se presenta un panel de instrucciones con formato tabular y estilizado, lo

que facilita al usuario comprender cómo se juega y mejora la estética general del programa.

## JUSTIFICACIÓN DEL PROYECTO

Este proyecto surge de la necesidad de aplicar conocimientos fundamentales de programación de forma práctica y entretenida. El enfoque del casino ofrece un abanico de juegos con mecánicas diferentes pero estructuradas, ideales para entrenar el uso de condicionales, bucles, clases, validación de datos y gestión del flujo del programa.

Además, se ha buscado que el sistema sea lo más visual y amigable posible dentro del entorno de consola. Para ello se ha utilizado ``rich``, que permite mejorar la experiencia del usuario y desarrollar aplicaciones en consola mucho más completas.

La naturaleza modular del proyecto también permite que el código sea mantenible, ampliable y adaptable a otros contextos como sistemas de puntos, ranking de jugadores, o historial de partidas.

## ANÁLISIS DE TECNOLOGÍAS UTILIZADAS

### **PYTHON:**

Lenguaje principal del desarrollo, ideal por su sintaxis clara, su orientación a objetos y su flexibilidad. Se usaron módulos como ``random``, ``os``, ``time`` y ``rich``.

### **RICH:**

Módulo de visualización en consola que permite imprimir texto en color, hacer tablas, paneles y mostrar datos de manera organizada y atractiva para el usuario.

### **VISUAL STUDIO CODE:**

Editor utilizado para escribir, depurar y ejecutar el código. La terminal integrada y el resaltado de sintaxis facilitaron mucho el desarrollo.

### **WORD / DRAW.IO:**

Se utilizaron para estructurar la memoria y planificar el diseño lógico del proyecto antes de codificar.

## CONCLUSIÓN

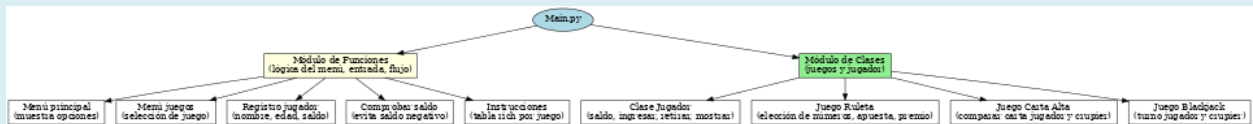
El proyecto de casino ha sido una forma excelente de poner en práctica conocimientos de programación en Python. Combina elementos de azar, estructuras de decisión, clases y diseño modular. La experiencia del usuario ha sido especialmente cuidada mediante ``rich``, y el resultado es un programa robusto, expandible y

visualmente atractivo.

El sistema está preparado para crecer, pudiendo añadir nuevos juegos, funcionalidades o incluso una interfaz gráfica en el futuro. Ha sido un trabajo completo, con sentido didáctico y funcional.

## DIAGRAMA DE ESTRUCTURA DEL PROYECTO

A continuación se muestra un diagrama que representa la estructura general del programa, dividido en módulos principales y funcionalidades clave:



## PSEUDOCÓDIGO GENERAL DEL PROGRAMA

Inicio

↓

Registrar jugador (nombre, edad, saldo inicial)

↓

Mostrar menú principal:

- Jugar
- Ver saldo
- Depositar / Retirar saldo
- Instrucciones
- Salir

↓

Si 'Jugar' elegido:

→ Mostrar menú de juegos:

- Ruleta:
  - \* Pedir dos números y una apuesta
  - \* Girar ruleta (generar dos números aleatorios)
  - \* Comparar con los elegidos y calcular premio
- Carta Alta:
  - \* Pedir una apuesta
  - \* Generar carta jugador y crupier
  - \* Comparar y calcular premio
- Blackjack:
  - \* Pedir una apuesta
  - \* Repartir cartas
  - \* Jugador elige: pedir / plantarse
  - \* Crupier juega hasta sumar 17+

\* Comparar y calcular premio

↓

Actualizar saldo

↓

Volver al menú principal o salir

Fin