

Juego ruleta electrónica

RouletTech Systems

- **Funcionamiento del Juego**

- **Configuración inicial:**

- Se ingresa un número determinado de jugadores, a cada uno de los cuales se le asigna una cantidad fija de fichas.
- Los jugadores tienen la posibilidad de realizar apuestas en cada ronda, con un máximo de **6 apuestas**:
 - **3 números.**
 - **Color** (rojo o negro).
 - **Par/Impar.**
 - **Docena** (primera, segunda o tercera docena).

- **Proceso de juego:**

- Cada jugador realiza sus apuestas. Una vez que todos los jugadores han apostado, se genera un número aleatorio simulando el giro de la ruleta.
- El sistema verifica los resultados y calcula las ganancias o pérdidas de cada jugador.
- Los jugadores que se queden sin saldo automáticamente abandonan el juego, y aquellos que lo deseen pueden retirarse voluntariamente.
- El juego termina cuando no quedan jugadores activos o cuando se decide finalizarlo manualmente.

- **Informes y seguimiento:**

- Antes de iniciar una nueva ronda, se genera un informe detallado que incluye:
 - El total apostado por cada jugador.
 - El saldo restante.
 - Las ganancias obtenidas en la ronda.
- Este informe permite a los jugadores tomar decisiones sobre si continuar o retirarse del juego.

- **Estructura del Programa**

- El sistema está diseñado con una estructura modular y utiliza las siguientes clases principales:

- **Clase RuletaMenuPrincipal**

- Clase Principal del proyecto "RouletTech Systems".
- Esta clase inicializa el sistema de la ruleta llamando al menú principal para que el usuario pueda interactuar con la aplicación

- **Uso:**

- - Ejecutar el método main para iniciar la aplicación.

- **Clase RuletaMenuPrincipal**

- Representa la ventana principal del juego de ruleta. Esta clase gestiona la interfaz gráfica para interactuar con el usuario, permitiéndole realizar acciones como agregar jugadores, cargar apuestas, ver resultados, iniciar una ronda, y salir del juego. Utiliza una serie de botones para ejecutar las funciones disponibles del juego.

- **Atributos:**

- - frame: El contenedor principal de la ventana de la interfaz gráfica.
- - apuestasTextArea: Área de texto para mostrar las apuestas realizadas.
- - jugadores: Array que almacena los jugadores activos en el juego.
- - jugadorActual: Controla el índice del jugador que está en turno.

- **Métodos:**

- - mostrarVentana: Muestra la ventana principal del menú del juego.
- - gestionarApuestasJugador: Permite gestionar las apuestas de un jugador durante su turno.
- - mostrarReglasDelJuego: Muestra las reglas del juego en una ventana de texto.
- - iniciarNuevaRonda: Reinicia las apuestas y las fichas de los jugadores para una nueva ronda.

- **Clase Persona (Clase Madre)**

- Representa a un jugador en el juego de la ruleta.
- Esta clase contiene los atributos básicos de un jugador como su nombre, la cantidad de fichas disponibles y su estado (activo o no).
- Además, ofrece métodos para registrar a un jugador, darlo de baja, actualizar su cantidad de fichas y mostrar su información.

- **Atributos:**

- - nombre: El nombre del jugador.
- - cantFichas: La cantidad de fichas que posee el jugador.
- - activo: Un valor booleano que indica si el jugador está activo en el juego.

- **Métodos:**

- - registrarJugador: Registra al jugador con un nombre y cantidad de fichas.
- - bajaJugador: Desactiva al jugador, indicándole que ha salido del juego.
- - actualizarFichas: Modifica la cantidad de fichas del jugador, permitiendo sumar o restar fichas.
- - toString: Devuelve una representación textual del jugador con su nombre y cantidad de fichas.
-

- **Clase Jugador (Hereda de Persona)**

- Representa a un jugador en el juego de la ruleta. Esta clase hereda de la clase Persona y gestiona las apuestas realizadas por un jugador en las distintas categorías: números, color, paridad y docena.

- **Atributos:**

- - apuestasNumeros: Array de apuestas a números (máximo 3).
- - apuestaColor: Apuesta realizada en color (rojo/negro).
- - apuestaParImpar: Apuesta realizada en paridad (par/impar).
- - apuestaDocena: Apuesta realizada en docena (1, 2, o 3).
- - montos: Array con el monto apostado en cada categoría (números, color, paridad, docena).
- - numApuestas: Número de apuestas realizadas (máximo 3 números).

- **Métodos:**

- - apostarNumero: Realiza una apuesta a un número específico.
- - apostarColor: Realiza una apuesta al color (rojo/negro).
- - apostarParidad: Realiza una apuesta a la paridad (par/impar).
- - apostarDocenas: Realiza una apuesta a la docena (1, 2 o 3).
- - reiniciarApuestas: Reinicia las apuestas del jugador al comenzar una nueva ronda.
- - toString: Devuelve una representación textual del jugador con sus apuestas y saldo de fichas.

- **Clase GestorDatos**

- Encargada de gestionar los datos del juego, incluyendo la carga de jugadores, la baja de jugadores y el manejo de las apuestas. Esta clase contiene métodos para registrar jugadores, manejar las apuestas a números, colores, paridad y docenas, y validar los datos ingresados por los jugadores.

- **Métodos principales:**

- - cargarJugadores: Carga los jugadores en el juego.
- - darDeBajaJugador: Baja un jugador, desactivándolo para futuras apuestas.
- - cargarJugador: Carga los datos de un jugador, incluyendo nombre y cantidad de fichas.
- - cargarApuestaNumeros: Permite a un jugador realizar una apuesta a números.
- - cargarApuestaColor: Permite a un jugador realizar una apuesta al color (rojo/negro).
- - cargarApuestaParidad: Permite a un jugador realizar una apuesta a paridad (par/impar).
- - cargarApuestaDocena: Permite a un jugador realizar una apuesta a docenas.
- - validarFichas: Valida si el jugador tiene suficiente saldo para realizar una apuesta.
-

- **Clase GestorJuego**

- Encargada de gestionar la lógica del juego de la ruleta. Esta clase maneja el cálculo y la validación de las apuestas realizadas por los jugadores, así como la determinación de los ganadores en cada ronda.

- **Atributos:**
 - - paridad: Array que contiene la paridad (par/impar) para cada número de la ruleta.
 - - docena: Array que contiene la docena (1, 2, 3) para cada número de la ruleta.
 - - colores: Array que contiene el color (rojo, negro, verde) para cada número de la ruleta.
 - - numero: El número que salió en la ruleta en esta ronda.
- **Métodos:**
 - - inicializarParidad: Inicializa el array de paridad con los valores correspondientes a cada número.
 - - inicializarDocena: Inicializa el array de docenas con los valores correspondientes a cada número.
 - - inicializarColores: Inicializa el array de colores con los valores correspondientes a cada número.
 - - comprobarGanadores: Recorre los jugadores y determina si han ganado en alguna de las apuestas.
 - - mostrarResultadoRonda: Muestra los resultados de la ronda en un cuadro de diálogo.
- **Clase GestorDeImpresiones**
 - Encargada de gestionar las impresiones y listados de los jugadores y sus apuestas. Esta clase proporciona métodos para buscar jugadores, ordenar por nombre o fichas, y generar informes sobre las apuestas realizadas y los resultados finales de los jugadores.
- **Métodos principales:**
 - - buscarJugador: Busca un jugador por su nombre en el array de jugadores.
 - - listarApuestasRonda: Muestra todas las apuestas realizadas por los jugadores en una ronda.
 - - mostrarJugadorEncontrado: Muestra la información de un jugador dado su nombre.
 - - ordenarPorFichas: Ordena a los jugadores en el array por la cantidad de fichas en orden descendente.
 - - ordenarPorNombre: Ordena a los jugadores en el array por su nombre en orden alfabético.
 - - mostrarInformeFinal: Muestra el informe final con la cantidad de fichas de cada jugador.

Ejemplo del vector de jugadores

Jugador 1 ("Juan"):

```
Jugador[0] = {"Juan", 120, true, [25, 18, 36], "ROJO", "PAR",  
"COL_1", [5, 10, 2, 2, 2, 1]};
```

Nombre: "Juan".

Tiene 120 fichas disponibles. Está activo.

Apuestas:

Números: 25, 18, 36. Color: "ROJO". Paridad: "PAR". Docena: "COL_1".

Montos apostados:

5 fichas al 25, 10 al 18, 2 al 36. 2 fichas al color "ROJO". 2 fichas a "PAR". 1 ficha a "COL_1".