

<div>CICLO FORMATIVO DE GRADO SUPERIOR</div> <div>DESARROLLO DE APLICACIONES WEB</div> <div>UT2 – ESTRUCTURAS DE CONTROL</div>	<div>EFA MORATALAZ</div> <div>Profesor: DGC</div>	<div>Asignatura: PROGRAMACIÓN</div> <div>Fecha:</div>	Nota:
Alumno: _____			

## **ACTIVIDAD 2.2 – LA RULETA DE JAVA**

Vamos a crear una aplicación para jugar a la ruleta europea, tal y como se puede jugar en casinos, casas de apuestas, online, etc. Puesto que todos estamos evidentemente en contra de las apuestas y somos conscientes del riesgo que supone este tipo de juegos, vamos a apostar dinero de Java y vamos a simular las tiradas y los premios.

En primer lugar, el usuario deberá introducir la cantidad (en euros) que quiere depositar. El juego acabará sólo si el jugador se queda sin saldo o si él mismo decide abandonar (se le mostrará el saldo con el que ha terminado).

Posteriormente, aparecerá un menú donde el usuario podrá elegir entre las diferentes opciones para jugar:

1. Números pares
2. Números impares
3. Rojos
4. Negros

(Puedes añadir más funcionalidades una vez consigas que estas 4 funcionen correctamente)

Una vez el jugador seleccione la opción que desee, se le pedirá que inserte su apuesta, es decir, las monedas que se quiere jugar. Evidentemente, la apuesta no puede ser mayor que el saldo ni tampoco podrá ser menor que 0 (comprobar esto siempre).

En base a la opción tomada y la apuesta, se simulará la tirada. Se generará un número aleatorio entre 0 y 36 y se comprobará si el jugador ha ganado o ha perdido. Si el jugador acierta, el premio será el doble del importe apostado (apuesta=10 → premio=20). Si el jugador falla, pierde las monedas apostadas. Después, actualiza el saldo del jugador.

Para cada opción del menú se tendrá que implementar un algoritmo que decida si el jugador ha ganado o ha perdido. Es decir, en el caso de elegir números rojos, tendrás que controlar que números son rojos y cuáles son negros y, en función del número que haya salido en la tirada, decidir el premio.

Comprueba todas las casuísticas posibles y todas las situaciones donde el usuario pueda “romper” tu aplicación. Por ejemplo, el menú tiene opciones limitadas y, en caso de introducir otro número, se mostrará un mensaje de error y se pedirá introducir de nuevo la opción.

**Entregar el archivo .java correspondiente con el código del mismo, siguiendo la norma de creación de nombres del curso: UTx\_ACTy\_NombreApellidos, siendo x el nº del tema e y el nº de la actividad. No se corregirá ninguna actividad que no esté subida con este formato.**

ITEMS	SI (1 PUNTO)	NO (0 PUNTOS)
1. ¿Gestiona correctamente el saldo y las apuestas del jugador?		
2. ¿Tiene las 4 funcionalidades básicas y funcionan correctamente?		
3. ¿Se incluye alguna funcionalidad más a la ruleta?		
4. ¿Se utilizan estructuras de repetición para controlar la existencia de la partida?		
5. ¿Se utilizan estructuras de control?		
6. ¿Se utilizan métodos?		
7. ¿La aplicación creada tiene funcionalidad real? ¿Se simula el giro de ruleta? ¿La interacción jugador-máquina es cómoda e intuitiva?		
8. ¿El código está comentado adecuadamente y tiene una estructura clara?		
9. ¿Las variables declaradas tienen nombres descriptivos e identificativos?		
10. ¿Se controlan todo tipo de errores?		