



Powered by
Arizona State University

PROYECTO DESARROLLO SOFTWARE JUEGO PIEDRA, PAPEL O TIJERA



ÍNDICE

1. Descripción del Juego.
2. Como se juega: Usuario VS Computadora y Multijugador.
3. Ambiente IDE de desarrollo.
4. Lenguaje de Programación
5. Librerías utilizadas.
6. Instrucciones de Instalación.
7. Código del Juego.
8. Demostración en pantalla del juego.

DESCRIPCIÓN DEL JUEGO

- Es un juego de manos en el que participan al menos 2 jugadores.
- Cada jugador elige 1 de 3 opciones: piedra, papel o tijera.
- El objetivo es vencer al oponente.



Reglas del Juego :



COMO SE JUEGA: MODO DE JUEGO USUARIO VS COMPUTADORA

1

Inicio del Juego

El usuario accede al menú principal de la aplicación:

- Home
- Configuración
- Perfil de Usuario
- Historial y Estadísticas
- Iniciar Juego
- Salir del Juego



2

La aplicación consultara el modo de juego que desea jugar

- 1: Usuario vs Computadora
- 2: Multijugador.

3

Si selecciona Usuario vs Computadora.

4

Se muestran las reglas generales del juego y en específico para la modalidad elegida.

5

Ingresar nombre del jugador.

6

El usuario decide la modalidad de juego:

- partidas definidas
- partidas libres.

7

Iniciar la Partida



8

Usuario elige :
piedra, papel o tijera.



9

La computadora elegira su opción al azar.



10

La computadora realiza el algoritmo del juego para determinar el resultado.

11

Se mostrará el resultado de la partida:

- Empate
- Ganaste
- Perdiste

12

Se consulta al usuario si desea seguir jugando:

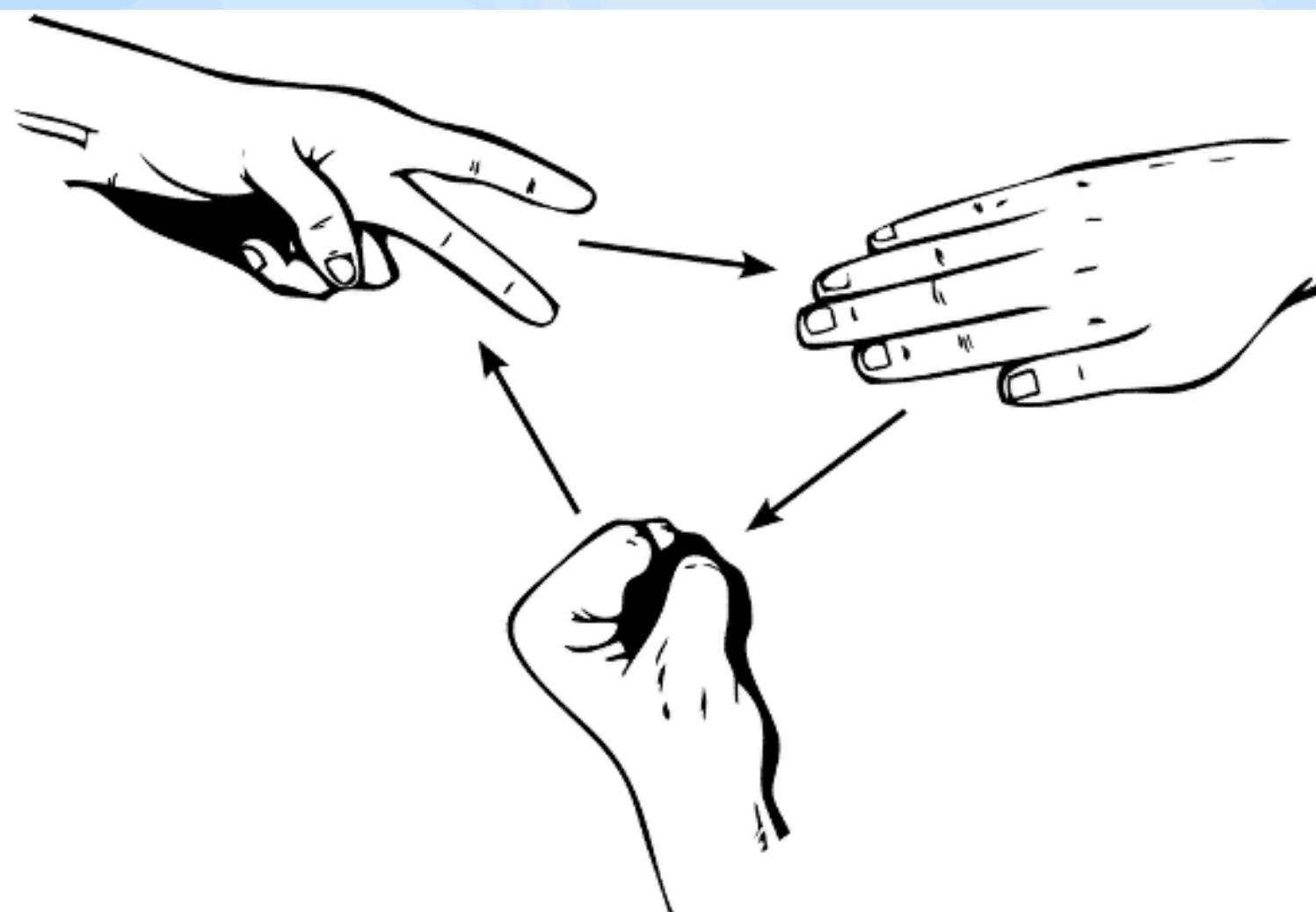


SI: Juega una nueva partida bajo la misma modalidad.

NO: Se muestra el registro histórico y estadística de la partida.

13

Finalmente se muestra el menú principal y el usuario decide si quiere volver a jugar o salir del juego.



COMO SE JUEGA:

MODO DE JUEGO

MULTIJUGADOR

USUARIO VS USUARIO

1

Inicio del Juego

El usuario accede al menú de la aplicación:

- Home
- Configuración
- Perfil de Usuario
- Historial y Estadísticas
- Iniciar Juego
- Salir del Juego



2

La aplicación consultará el modo de juego que desea jugar

- 1: Usuario vs Computadora
- 2: Multijugador.

3

Si selecciona Multijugador Usuario VS Usuario.

4

Se muestran las reglas generales del juego y en específico para la modalidad elegida.

5

Ingresar nombre del jugador 1 (jugador principal).

6

Ingresar nombre del jugador 2.



7

El jugador principal (J1) decide la modalidad de juego:

- partidas definidas
- partidas libres.



8

Iniciar la Partida.



9

Usuario (J1) elige : piedra, papel o tijera.

10

Usuario (J2) elige : piedra, papel o tijera.

11

La computadora realiza el algoritmo del juego para determinar el resultado.

12

Se mostrará el resultado de la partida:

- Empate
- Ganaste
- Perdiste

13

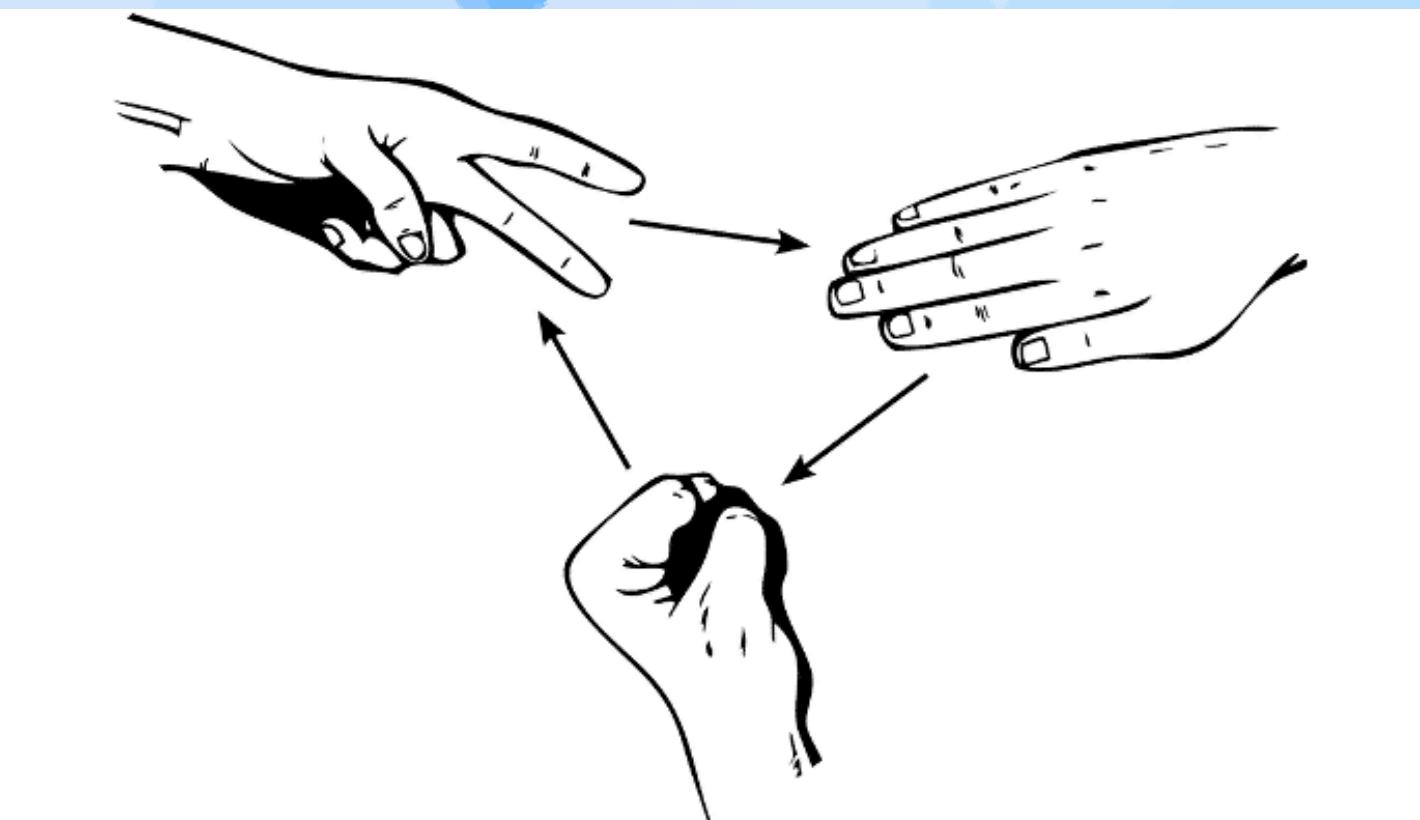
Se consulta al usuario principal (J1) si desea seguir jugando:

SI: Juega una nueva partida bajo la misma modalidad.

NO: Se muestra el registro histórico y estadística de la partida.

14

Finalmente se muestra el menú principal y el usuario decide si quiere volver a jugar o salir del juego.



AMBIENTE IDE DE DESARROLLO

Spyder version 6.0.3 (Standalone)

LENGUAJE DE PROGRAMACIÓN

Python versión 3.11.10 64 bit | Qt 5.15.8 |

PyQt5 5.15.9



python™

LIBRERIAS UTILIZADAS

Random

Se utiliza para generar números aleatorios.

En este juego, se usa `random.choice(opciones)`.

La computadora elige una opción al azar:
piedra, papel o tijera.

Get Pass

Se utiliza en el modo de juego multijugador.

No se muestra en pantalla la opción (piedra, papel o tijera) seleccionada por cada jugador.



INSTRUCCIONES DE INSTALACIÓN

1

Clona este repositorio: git clone <https://github.com/KevinMejia10/Juego-Piedra-Papel-o-Tijera.git>

2

Descarga y abre el archivo "Juego.py" con Python.

3

Sigue las instrucciones en pantalla para jugar.

Rock Paper Scissors

Select your ammo:



Who will win the first game?

Your score: 0

Computer's score: 0

reset score

By Elle J.
Github | Twitter

Esta función configura el juego, inicializando:

- Modo de juego
- El histórico de juegos.
- Estadísticas de los jugadores y la computadora.

CÓDIGO DEL JUEGO

Inicializa las variables y estructuras de datos necesarias para el juego.

VARIABLES

modo_juego (int): 0 indica que el juego no ha comenzado.

historico_juegos (list): Lista para almacenar el resultado de cada juego.

estadisticas (dict): Diccionario para almacenar las estadísticas de cada jugador y la computadora, incluyendo juegos ganados, perdidos y empatados.

MÓDULOS

random: Se utiliza para generar elecciones aleatorias para la computadora.

getpass: Se utiliza para ocultar la entrada del jugador en el modo multijugador

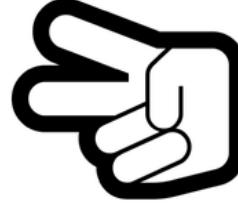
```
import random # Se utiliza para generar elecciones aleatorias para la computadora.  
import getpass # Importa el módulo getpass para ocultar la entrada  
  
modo_juego = 0  
  
# Variables para almacenar el histórico y las estadísticas del juego  
historico_juegos = []  
estadisticas = {  
    "jugador1": {"ganadas": 0, "perdidas": 0, "empatadas": 0},  
    "jugador2": {"ganadas": 0, "perdidas": 0, "empatadas": 0},  
    "computadora": {"ganadas": 0, "perdidas": 0, "empatadas": 0},  
}
```



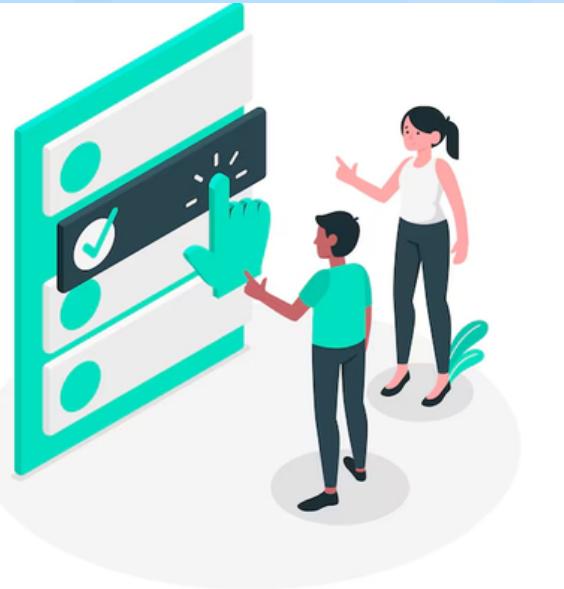
PIEDRA



PAPEL



TIJERA



FUNCIÓN MANEJAR:OPCIÓN(OPCIÓN)

Maneja la opción seleccionada por el usuario en el menú principal.

Esta función recibe la opción elegida por el usuario y ejecuta la acción correspondiente.

Args

opcion (int): El número de la opción seleccionada por el usuario.

Returns

bool: True si la opción es válida y no es la de salida (opción 6), False si la opción es inválida.

Si la opción es la de salida (6), también retorna True, pero el programa principal debería terminar.

Funciones llamadas

- **historico_juegos (list):** Lista que almacena el histórico de los juegos. Se reinicia al iniciar un nuevo juego.
- **estadisticas (dict):** Diccionario que almacena las estadísticas de los jugadores y la computadora. Se reinicia al iniciar un nuevo juego.

- **mostrar_menu_modos_juego():** Muestra el menú para seleccionar el modo de juego.
- **mostrar_historico_estadisticas():** Muestra el histórico y las estadísticas del juego.

Las opciones disponibles son:

- 1 Home.
- 2 Configuración del juego.
- 3 Perfil de usuario.
- 4 Iniciar juego: Inicia el juego, reinicia las estadísticas y el histórico, y muestra el menú de modos de juego.
- 5 Histórico y estadísticas.
- 6 Salir del juego.

Cualquier otra opción se considera inválida y se muestra un mensaje de error.

FUNCIÓN MOSTRAR_MENU_MODOS_JUEGO():

Muestra el menú de modos de juego y maneja la selección del usuario.

Esta función presenta al usuario las opciones de modos de juego disponibles.

Variables globales utilizadas

- **modo_juego (int)**: Almacena el modo de juego seleccionado por el usuario.



Funciones llamadas

- **mostrar_reglas_usuario_vs_computador_a()**: Muestra las reglas del modo Usuario vs Computadora.
- **jugar_usuario_vs_computadora()**: Inicia el juego en modo Usuario vs Computadora.
- **mostrar_reglas_multijugador()**: Muestra las reglas del modo Multijugador.
- **jugar_multijugador()**: Inicia el juego en modo Multijugador.
- **main()**: Regresa al menú principal del juego.

Las opciones disponibles son:

- 1 Usuario vs Computadora.
- 2 Multijugador (Usuario vs Usuario).
- 3 Regresar al menú principal.

El usuario debe ingresar el número correspondiente al modo de juego deseado.

Se manejan errores de entrada (valores no numéricos y opciones inválidas) y se vuelve a pedir la entrada hasta que sea válida

FUNCIÓN MOSTRAR_REGLAS_USUARIO_VS_COMPUTADORA():

Muestra las reglas del juego en la modalidad Usuario vs Computadora.

Esta función imprime en la consola las reglas generales del juego, así como las reglas específicas para la modalidad Usuario vs Computadora.

Reglas modalidad Usuario vs Computadora:

- El usuario elige una de las tres opciones (piedra, papel o tijera).
- La computadora elige aleatoriamente una de las tres opciones.
- Se comparan las elecciones y se determina el resultado según las reglas generales.



Reglas Generales:

- La piedra aplasta la tijera (la piedra gana).
- La tijera corta el papel (la tijera gana).
- El papel envuelve la piedra (el papel gana).
- Si ambos jugadores eligen la misma opción, es un empate.

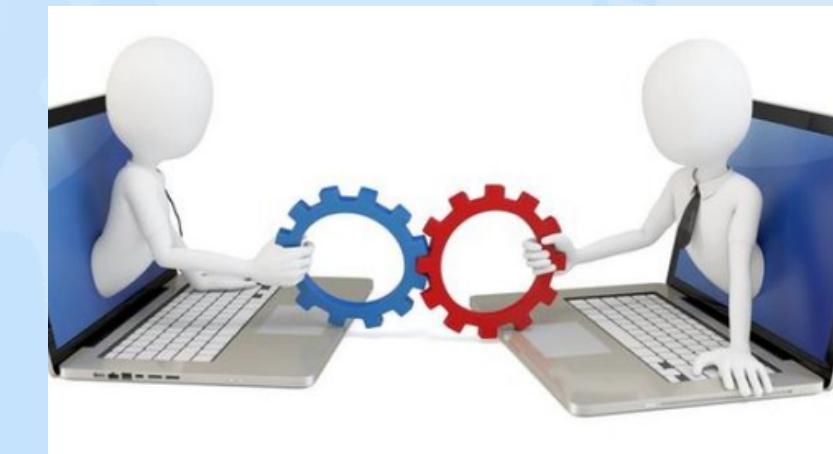
FUNCIÓN MOSTRAR_REGLAS_MULTIJUGADOR():

Muestra las reglas del juego en la modalidad Multijugador (Usuario vs Usuario).

Esta función imprime en la consola las reglas generales del juego, así como las reglas específicas para la modalidad Multijugador (Usuario vs Usuario).

Reglas modalidad Multijugador (Usuario vs Usuario):

- Dos jugadores se enfrentan.
- Ambos jugadores eligen una de las tres opciones (piedra, papel o tijera).
- Se comparan las elecciones y se determina el resultado según las reglas generales.



FUNCIÓN JUGAR_USUARIO_VS_COMPUTADORA():



Implementa el juego en la modalidad Usuario vs Computadora.

Esta función permite al usuario jugar contra la computadora.

El usuario puede elegir jugar un número definido de partidas o jugar partidas libres hasta que decida no seguir jugando.

Se registran las estadísticas de cada partida y se muestra un resumen al final.

Variables globales utilizadas

- **estadisticas (dict)**: Diccionario que almacena las estadísticas del juego. Se modifica dentro de esta función.
- **historico_juegos (list)**: Lista que almacena el histórico de las partidas jugadas.
- **main (function)**: Función que representa el menú principal del juego. Se llama al finalizar la partida.
- **mostrar_historico_estadisticas (function)**: Función que muestra el histórico y las estadísticas. Se llama al finalizar la partida.
- **random (module)**: Módulo utilizado para la elección aleatoria de la computadora.

FUNCIÓN JUGAR_MULTIJUGADOR():

Implementa el juego en la modalidad Multijugador (Usuario vs Usuario).

Esta función permite que dos jugadores se enfrenten en el juego.

El jugador principal (J1) elige jugar un número definido de partidas o jugar partidas libres hasta que decida no seguir jugando.

Las opciones (piedra, papel, tijera) elegidas por cada jugador se ocultan usando “getpass” para mayor privacidad.

Se registran las estadísticas de cada partida y se muestra un resumen al final.



Variables globales utilizadas

- **estadisticas (dict)**: Diccionario que almacena las estadísticas del juego. Se modifica dentro de esta función.
- **historico_juegos (list)**: Lista que almacena el histórico de las partidas jugadas. Se modifica dentro de esta función.
- **main (function)**: Función que representa el menú principal del juego. Se llama al finalizar la partida.
- **mostrar_historico_estadisticas (function)**: Función que muestra el histórico y las estadísticas. Se llama al finalizar la partida.
- **getpass (module)**: Módulo utilizado para ocultar la entrada de los jugadores.

FUNCIÓN MOSTRAR_HISTÓRICO_ESTADÍSTICAS():

Muestra el histórico y las estadísticas del juego.

Esta función imprime en la consola el histórico de las partidas jugadas, mostrando:

- Los nombres de los jugadores.
- Sus elecciones.
- Resultado de cada partida.

También muestra las estadísticas generales del juego, incluyendo el total de partidas jugadas y las estadísticas individuales de cada jugador (ganadas, perdidas y empatadas).

Si no hay juegos registrados en el histórico, se muestra un mensaje indicando que no hay estadísticas recientes.

Si el jugador reinicia el juego, el histórico y estadísticas de la partida anterior se borran.



Variables globales utilizadas

- **historico_juegos (list):** Lista que contiene diccionarios, donde cada diccionario representa una partida y contiene información sobre los jugadores, sus opciones y el resultado.
- **estadisticas (dict):** Diccionario que contiene las estadísticas de cada jugador. Las claves del diccionario son los nombres de los jugadores y los valores son diccionarios con las estadísticas (ganadas, perdidas, empatadas).
- **modo_juego (int):** Variable que indica el modo de juego (1 para Usuario vs Computadora, 2 para Multijugador). Se usa para mostrar las estadísticas correspondientes a cada modo.
- **main (function):** Función que representa el menú principal del juego. Se llama automáticamente al final de esta función para regresar al menú.

FUNCIÓN MAIN():

Función principal que ejecuta el bucle del menú de la aplicación del juego.

Esta función presenta un menú principal al usuario.

El programa solicita al usuario que ingrese el número de la opción deseada.

Se maneja errores de entrada en caso de que el usuario ingrese un valor no numérico.

La función manejar_opcion() es la encargada de procesar la opción elegida por el usuario.

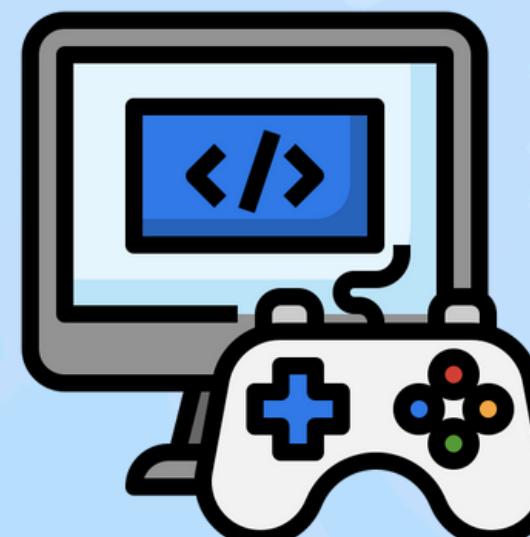
El bucle continúa hasta que el usuario selecciona la opción 6 (Salir del juego).

Variables globales utilizadas

- **salir (bool)**: Variable que controla la terminación del bucle. Se inicializa en `False` y se establece en `True` cuando el usuario elige la opción de salir.
- **opcion (int)**: Variable que almacena el número de la opción ingresada por el usuario.

Funciones llamadas

- **manejar_opcion(opcion)**: Función que procesa la opción elegida por el usuario.



A group of five diverse children (three girls and two boys) are sitting around a table, looking at a laptop screen together. They appear to be engaged in a collaborative activity or game. The background features a colorful, abstract design with various icons and data visualizations, suggesting a digital or educational theme.

iMuchas
gracias!