

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

FACULTAD DE INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN

LÓGICA DE PROGRAMACIÓN 2-ECC-1B

DESARROLLO Y PROGRAMACIÓN DEL JUEGO PIEDRA, PAPEL O TIJERA: LÓGICA, INTERFAZ Y ESTADÍSTICAS PERSISTENTES

AUTHOR: Christian David Muñoz Tonguino

INSTRUCTOR: Lilian Marlene Aman Ramos

DATE: October, 2025

ABSTRACT El presente informe detalla el desarrollo de una aplicación interactiva del clásico juego Piedra, Papel o Tijera, implementada en el lenguaje de programación Python, utilizando la librería Tkinter para la interfaz gráfica. El proyecto fue concebido como una manera de aplicar conocimientos fundamentales de programación estructurada, lógica condicional, control de flujo, almacenamiento de datos y diseño de interfaces de usuario. A través de la construcción de este juego, se lograron integrar aspectos tanto técnicos como prácticos, permitiendo al usuario elegir entre diferentes modos de juego, visualizar estadísticas persistentes y experimentar una experiencia interactiva completa. El documento presenta la motivación del proyecto, su análisis funcional y técnico, los resultados obtenidos, así como las limitaciones e implicaciones encontradas a lo largo del proceso de desarrollo. Adicionalmente, se anexa un diagrama de flujo, un video explicativo y un documento de análisis complementario para una comprensión más integral del trabajo.

INTRODUCCIÓN

El desarrollo de software orientado a la práctica permite a los estudiantes y programadores no solo adquirir destrezas técnicas, sino también fomentar la resolución de problemas, la lógica algorítmica y el pensamiento estructurado. En ese contexto, se eligió desarrollar una versión computarizada del juego tradicional Piedra, Papel o Tijera, por tratarse de una dinámica simple, pero con múltiples posibilidades de implementación.

Este juego es ampliamente conocido en la cultura popular y posee reglas claras y fáciles de programar. Sin embargo, el proyecto fue más allá de replicar sus reglas básicas: se propuso crear una interfaz visual interactiva, incorporar diferentes modos de juego (contra la CPU y entre dos jugadores), agregar tipos de partidas (una sola ronda o mejor de tres) y registrar estadísticas persistentes a lo largo de las sesiones.

El presente informe expone de manera estructurada todo el proceso de desarrollo, desde la concepción inicial de la idea, pasando por la planificación y análisis lógico, hasta la ejecución técnica y evaluación de resultados. También se abordan los problemas encontrados y las soluciones adoptadas, así como una reflexión final sobre la experiencia de crear este proyecto.

MOTIVACIÓN Y ELECCIÓN DEL PROYECTO

La elección de desarrollar el juego Piedra, Papel o Tijera surge de la necesidad de implementar un proyecto que fuera manejable, didáctico y funcional. Este tipo de juegos permite centrar la atención en la lógica del programa, en lugar de depender de estructuras complejas de datos o cálculos matemáticos avanzados. Además, su familiaridad garantiza que cualquier usuario pueda interactuar fácilmente con la aplicación, lo cual es esencial para probar la usabilidad y robustez del sistema.

Desde el punto de vista académico, el proyecto también representa una excelente oportunidad para aplicar los conocimientos adquiridos en diversas áreas:

- Estructuras condicionales y de control.
- Funciones y modularidad del código.
- Interfaces gráficas con Tkinter.
- Lectura y escritura de archivos (JSON).
- Control de estado entre pantallas de una aplicación.

El objetivo general fue crear una aplicación interactiva, funcional y visualmente amigable, capaz de ejecutarse sin dependencias externas y con un enfoque centrado en la experiencia del usuario.

ANÁLISIS FUNCIONAL Y DEL CÓDIGO

El desarrollo del juego se dividió en varias etapas y módulos funcionales claramente definidos:

1) Variables Globales

Se utilizaron variables globales para mantener el estado del juego: puntajes, elecciones de los jugadores, número de rondas jugadas, tipo de partida, etc. Estas variables son necesarias para que el flujo del juego se mantenga coherente entre funciones y pantallas.

2) Manejo de Estadísticas

Se creó un archivo estadisticas.json que almacena el número de victorias de cada jugador (Jugador 1, Jugador 2, CPU) y los empates. Las funciones cargar_estadisticas(), guardar_estadisticas() y actualizar_estadisticas() se encargan de mantener estos datos actualizados cada vez que se completa una partida.

3) Lógica del Juego

La función determinar_ganador(j1, j2) encapsula la lógica fundamental del juego, utilizando condicionales para establecer quién gana entre dos elecciones posibles. A partir de ella, se gestiona la lógica completa del flujo de una partida, teniendo en cuenta si se juega contra la CPU o entre dos jugadores.

4) Interfaz Gráfica con Tkinter

Se implementó una interfaz gráfica compuesta por distintas "pantallas":

- Pantalla de inicio: Presenta el nombre del juego y el botón de iniciar.
- Menú de modo de juego: Permite seleccionar el tipo de enfrentamiento (contra CPU o 2 jugadores) y la modalidad (una jugada o mejor de 3).

- Pantalla de juego: Muestra los botones para seleccionar Piedra, Papel o Tijera, controlando el turno de cada jugador.
- Pantalla de resultados: Muestra el resultado final del enfrentamiento, junto con las estadísticas generales y botones para reiniciar o salir.

Se hizo uso de tk.Label, tk.Button, tk.Radiobutton y tk.Frame para construir la interfaz, y ventana.winfo_children() para limpiar y actualizar dinámicamente cada pantalla.

5) Control de Turnos

Un punto clave fue implementar correctamente el sistema de turnos para el modo 2 jugadores. Se utilizó una variable jugador1_eleccion para almacenar temporalmente la jugada del primer jugador, y luego se espera la selección del segundo antes de determinar el resultado.

RESULTADOS OBTENIDOS

Al finalizar el desarrollo, se obtuvieron los siguientes resultados:

- Se creó una aplicación funcional, clara y amigable para el usuario.
- El programa ofrece una experiencia fluida, sin errores críticos durante su ejecución.
- Se implementó exitosamente el almacenamiento de estadísticas persistentes entre sesiones.
- El juego permite reiniciar sin necesidad de cerrar y volver a abrir el programa.
- El flujo del programa fue representado gráficamente en un diagrama de flujo en PDF.
- Se grabó un video explicativo detallando cada parte del código y el diagrama, facilitando su comprensión para terceros.
- Se elaboró este informe y un documento complementario en Word que detalla el proceso completo del proyecto.

IMPLICACIONES Y LIMITACIONES

Durante el desarrollo del proyecto surgieron diversas implicaciones técnicas y limitaciones prácticas:

Implicaciones:

- Fue necesario estructurar correctamente el código para evitar errores al cambiar entre pantallas.
- Se buscó una experiencia de usuario intuitiva, lo cual implicó varias pruebas de usabilidad.
- Se manejaron correctamente los archivos JSON para asegurar que las estadísticas se actualizarán sin corromper el archivo.

Limitaciones:

- Tkinter es limitado en diseño visual moderno. Aunque funcional, no permite crear interfaces con diseños avanzados o responsivos.
- No se implementó validación de errores más compleja (por ejemplo, cierres inesperados del programa o fallos de lectura del archivo de estadísticas).
- No se consideraron aún características como un historial de partidas o rankings por usuario.
- La lógica de turnos en 2 jugadores podría mejorarse usando pantallas separadas o entrada oculta para mayor equidad.

CONCLUSIONES

El desarrollo de este juego permitió aplicar de forma integrada numerosos conceptos fundamentales de la programación, desde estructuras condicionales y manejo de funciones hasta diseño de interfaces y persistencia de datos.

Se alcanzaron con éxito todos los objetivos propuestos inicialmente. El resultado es un programa completo, funcional, y fácil de entender y ejecutar, lo que lo convierte no solo en una herramienta lúdica, sino también en un recurso educativo ideal para quienes deseen aprender los fundamentos de la programación con Python.

Además, el proyecto sirvió como plataforma para reforzar habilidades de planificación, documentación técnica y presentación, elementos clave en el trabajo profesional de un desarrollador.

REFLEXIONES FINALES

La realización de este proyecto representó un desafío que, si bien no fue extremadamente complejo desde el punto de vista técnico, sí demandó una organización disciplinada, pruebas constantes y una comprensión sólida del flujo lógico de un programa.

Reflexionando sobre el proceso, puedo afirmar que:

- Convertir una idea simple en una aplicación completa requiere atención al detalle, paciencia y capacidad de resolución de problemas.
- El uso de herramientas visuales como Tkinter, aunque limitadas, puede ser muy útil para comprender la interacción entre código y usuario.
- Documentar y exponer el proyecto, tanto en forma escrita como mediante un video, ayudó a consolidar los conocimientos adquiridos.
- Los errores cometidos durante el desarrollo fueron valiosas oportunidades de aprendizaje.

Finalmente, considero que este proyecto ha sido un paso importante en mi formación como programador, y me motiva a seguir explorando proyectos más ambiciosos en el futuro, integrando más lógica, diseño, y funcionalidades avanzadas.

Link de acceso a los videos explicativos del diagrama de flujo y del código funcional del juego Piedra, Papel o Tijera:

https://mailinternacionaledu-my.sharepoint.com/personal/chmunozto_uide_edu_ec/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fchmunozto%5Fuide%5Fedu%5Fec%2FDocuments%2FL%C3%B3gica%20de%20programaci%C3%B3n%20%2D%20Evaluaci%C3%B3n&ga=1

Link de acceso al repositorio en GitHub en donde se encuentra el código funcional y el diagrama del juego Piedra, Papel o Tijera:

<https://github.com/ChrisLeo76908/Juego-piedra-papel-o-tijera.-.git>