

**UNIVERSIDAD INTERNACIONAL DEL ECUADOR**

**Facultad:** Ciencias Técnicas

**Carrera:** Ingeniería en Ciberseguridad

**Docente:** Mónica Salazar

**Asignatura:** Lógica de Programación

**Nombre del estudiante:** Dayana Yaselga

**Actividad:** Evaluación en Contacto con el Docente

**Período académico:** octubre 2024 – marzo 2025

**El impacto de las nuevas tecnologías en la sociedad:** visualización del futuro a través de un juego interactivo.

### **Descripción general del problema**

Las nuevas tecnologías han transformado todos los aspectos de la sociedad, incluyendo el entretenimiento y el aprendizaje. En este proyecto se utilizó el clásico juego Piedra, Papel o Tijeras para explorar cómo la programación y el desarrollo de software pueden ofrecer soluciones interactivas.

A través de este proyecto, se puede comprender cómo las tecnologías digitales permiten crear experiencias de forma virtual, fomentando un aprendizaje dinámico, desde la comprensión de conceptos básicos de programación, como la entrada (input), las estructuras condicionales, y la implementación de bucles para la automatización de procesos.

### **Propósito del proyecto**

El propósito principal de este proyecto es visualizar el impacto de las nuevas tecnologías en la sociedad a través del desarrollo de un juego interactivo, como Piedra, Papel o Tijeras, donde se pudo aplicar conocimientos básicos adquiridos en clase, desde el uso de estructuras de entrada (input) hasta la integración de bucles (while) que permitieron que el juego funcione de manera continua y eficiente, también fomento la creatividad y la innovación al integrar principios de programación y diseño en un entorno práctico, al crear un programa funcional en Python, se logró entender los conceptos aprendidos en clases y aplicarlos a un código funcional.

S1	S2	S3	S4	S5	S6	S7	S8	
Unidad 1		Unidad 2		Unidad 3		Unidad 4		
Tema 1	Tema 2	Tema 3	Tema 4	Tema 5	Tema 6	Tema 7	Tema 8	
<b>Actividad Autónoma 1.</b> <b>Selección del proyecto:</b> Se eligió desarrollar el software Piedra, Papel o Tijeras. <b>Investigación inicial:</b> Se investigó los diferentes tipos de diagramas funcionales y arquitectura de aplicaciones para comprender su uso en el desarrollo de software. <b>Análisis del problema:</b> Se identificó la necesidad de crear un programa interactivo, sencillo y funcional, y se plantearon posibles soluciones para su desarrollo. <b>Diseño del software:</b> Se elaboró un diagrama de caso de uso para describir las funcionalidades principales del programa y establecer lo que se espera que haga. Se diseñó un diagrama de arquitectura MVC, mostrando cómo los componentes del programa interactúan a nivel macro, dividiendo las responsabilidades en Modelo, Vista y Controlador.		<b>Actividad Autónoma 2.</b> <b>Inicio del desarrollo y configuración del entorno:</b> Se configuró el entorno de trabajo en Visual Studio Code para comenzar la codificación del programa. <b>Creación del pseudocódigo y diagrama de flujo:</b> Se redactó el pseudocódigo que describe la lógica del juego de Piedra, Papel o Tijeras. Con base en el pseudocódigo, se elaboró un diagrama de flujo que permitió visualizar cómo funcionaría el juego de manera lógica y estructurada. <b>Primera codificación:</b> Se inició la codificación del programa utilizando los conocimientos previos y el apoyo del diagrama de flujo para estructurar la lógica. <b>Uso de GitHub para organización y respaldo:</b> Se creó un repositorio en GitHub para almacenar el código y los documentos relacionados con el proyecto. Se subieron los avances del programa al repositorio.		<b>Actividad Autónoma 3.</b> <b>Desarrollo del programa:</b> En esta etapa, se aplicaron más conocimientos sobre estructuras lógicas, lo que permitió mejorar y optimizar la funcionalidad del programa. <b>Integración de condicionales:</b> Se implementaron estructuras condicionales como if y elif, lo que ayudó a definir el flujo lógico del juego y a gestionar las distintas decisiones y resultados posibles. <b>Uso de estructuras repetitivas:</b> Se incluyó un bucle while, que permitió mantener el juego en ejecución hasta que el usuario decidiera finalizarlo, haciendo la experiencia más dinámica e interactiva. <b>Contador de puntajes:</b> Se añadió un sistema para llevar un registro de puntajes, permitiendo a los jugadores visualizar sus resultados acumulados durante la partida.		<b>Actividad autónoma 4.</b> <b>Foro de dudas y comentarios.</b> La interacción en el foro resultó fundamental para mejorar el programa, ya que los comentarios y sugerencias de los compañeros ayudaron a identificar áreas de mejora y a optimizar el código. Retroalimentación mutua: Revisar los códigos de otros compañeros permitió no solo brindar apoyo, sino también aprender nuevas estrategias y enfoques para el desarrollo de programas similares.		<b>Entrega del software funcional y terminado.</b> <b>Entrega del código funcional:</b> Se logró desarrollar un programa completamente funcional que fue trabajado y mejorado durante las 8 semanas. <b>Publicación en GitHub:</b> El código, junto con todos los avances y documentos del proyecto, fue subido a un repositorio en GitHub para mantener la organización y facilitar el acceso. <b>Incorporación del archivo README:</b> Se incluyó un archivo README detallando la funcionalidad del código, las características principales y el propósito del programa. <b>Presentación del proyecto:</b> Se realizó una presentación completa del trabajo, explicando los procesos seguidos, las decisiones tomadas y el impacto del proyecto.

### Conclusiones:

- La implementación del juego Piedra, Papel o Tijeras permitió aplicar conceptos clave de programación como estructuras condicionales, bucles y organización de código.
- La creación de diagramas funcionales y arquitectura de aplicaciones ayudó a estructurar y visualizar mejor el programa antes de su codificación, resaltando la importancia de un proceso correcto.
- El uso de plataformas como GitHub fue esencial para el control de versiones y la colaboración, durante el desarrollo de software.
- La participación en foros y el intercambio de ideas con compañeros fomentaron una retroalimentación valiosa, mejorando no solo el programa propio sino también el entendimiento de diferentes programas.