



PIEDRA, PAPEL O TIJERA: VISUALIZACIÓN DEL FUTURO TECNOLÓGICO

Propuestas e ideas sobre
aplicaciones para el futuro
tecnológico



PLAY

Introducción

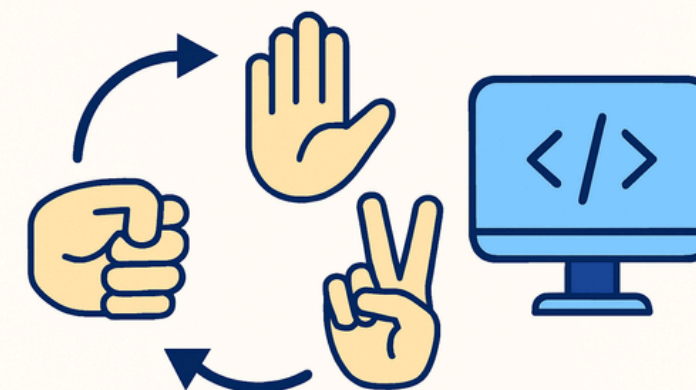
- El proyecto integrador busca aplicar conocimientos de programación en un videojuego sencillo y didáctico.
- Relación con el tema general: Impacto de las nuevas tecnologías en la sociedad.
- El juego Piedra, Papel o Tijera es un ejemplo de cómo la informática permite transformar actividades tradicionales en experiencias digitales interactivas.
- Se conecta con la idea de que los videojuegos son una forma de innovación tecnológica que impacta en educación, entretenimiento y cultura.

Objetivos

- Objetivo general: Desarrollar un videojuego funcional que aplique conceptos de programación y arquitectura de software.
- Objetivos específicos:
- Implementar lógica condicional en Python para simular decisiones.
- Diseñar diagramas funcionales y de arquitectura que representen el flujo del programa.
- Documentar el código con comentarios claros para facilitar su comprensión.
- Presentar resultados y aprendizajes en un entorno académico.
- Reflexionar sobre cómo los videojuegos representan el impacto de la tecnología en la sociedad.

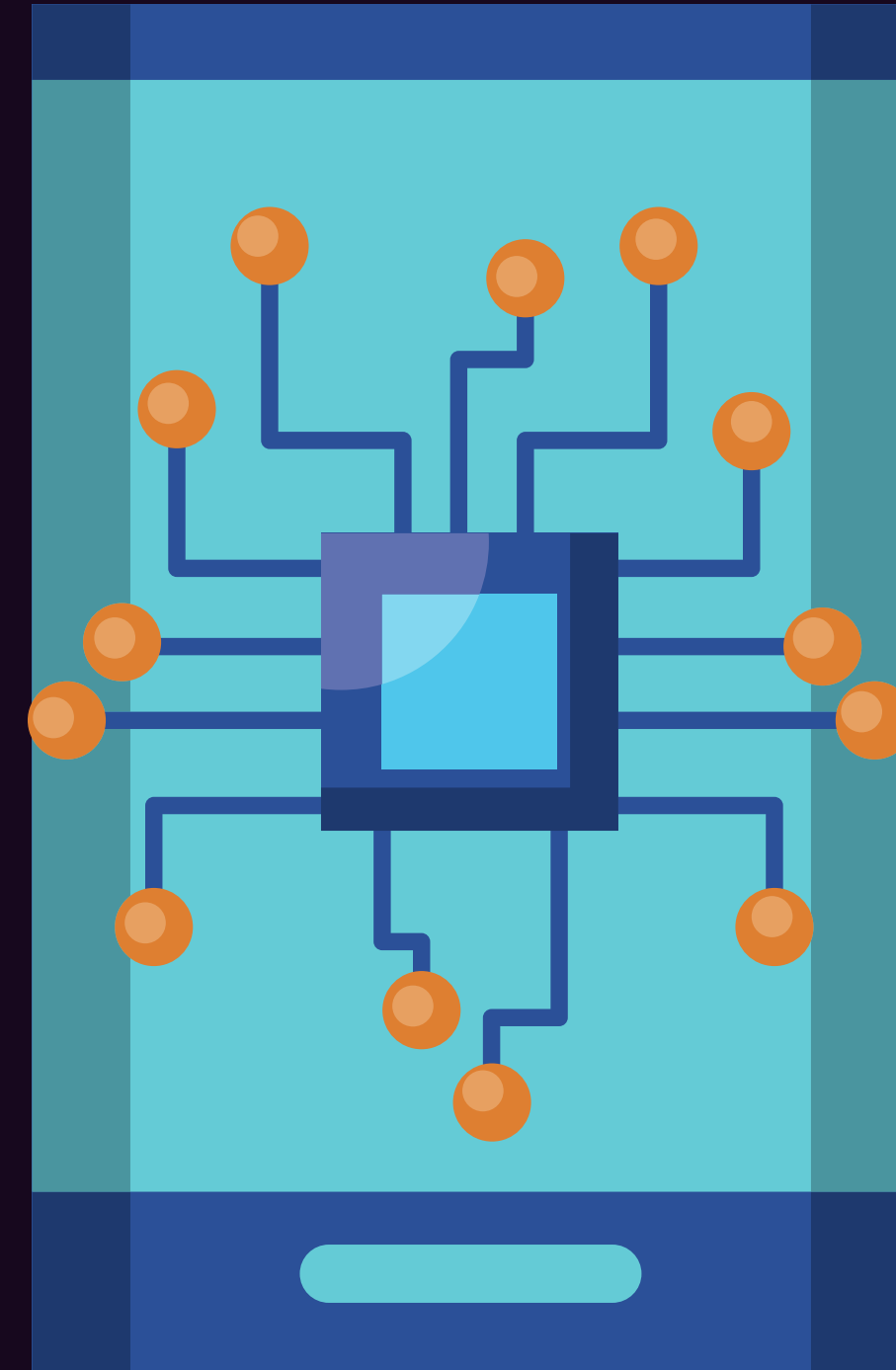


Introducción



Materiales y Herramientas

- Lenguaje: Python
- Librerías: random (para elección aleatoria), tkinter (si se usa interfaz gráfica).
- Entorno: VS Code / PyCharm para desarrollo.
- Herramientas de apoyo:
 - Canva para presentación.
 - GitHub para repositorio y control de versiones.
 - Draw.io / Lucidchart para diagramas.
- Hardware: Computadora personal con sistema operativo compatible.



Desarrollo del Programa



Función principal

Jugar(): controla la lógica del juego.

Entrada del usuario:

Se valida que sea piedra, papel o tijera.

Elección de la computadora:

Se genera aleatoriamente con `random.choice()`.

Comparación de opciones:

Evalúan condiciones para determinar si el jugador gana, empata o pierde.

Ciclo principal:

Permite repetir el juego hasta que el usuario decida salir.

Comentarios en el código:

Explican cada línea para facilitar la comprensión.

Diagrama Funcional

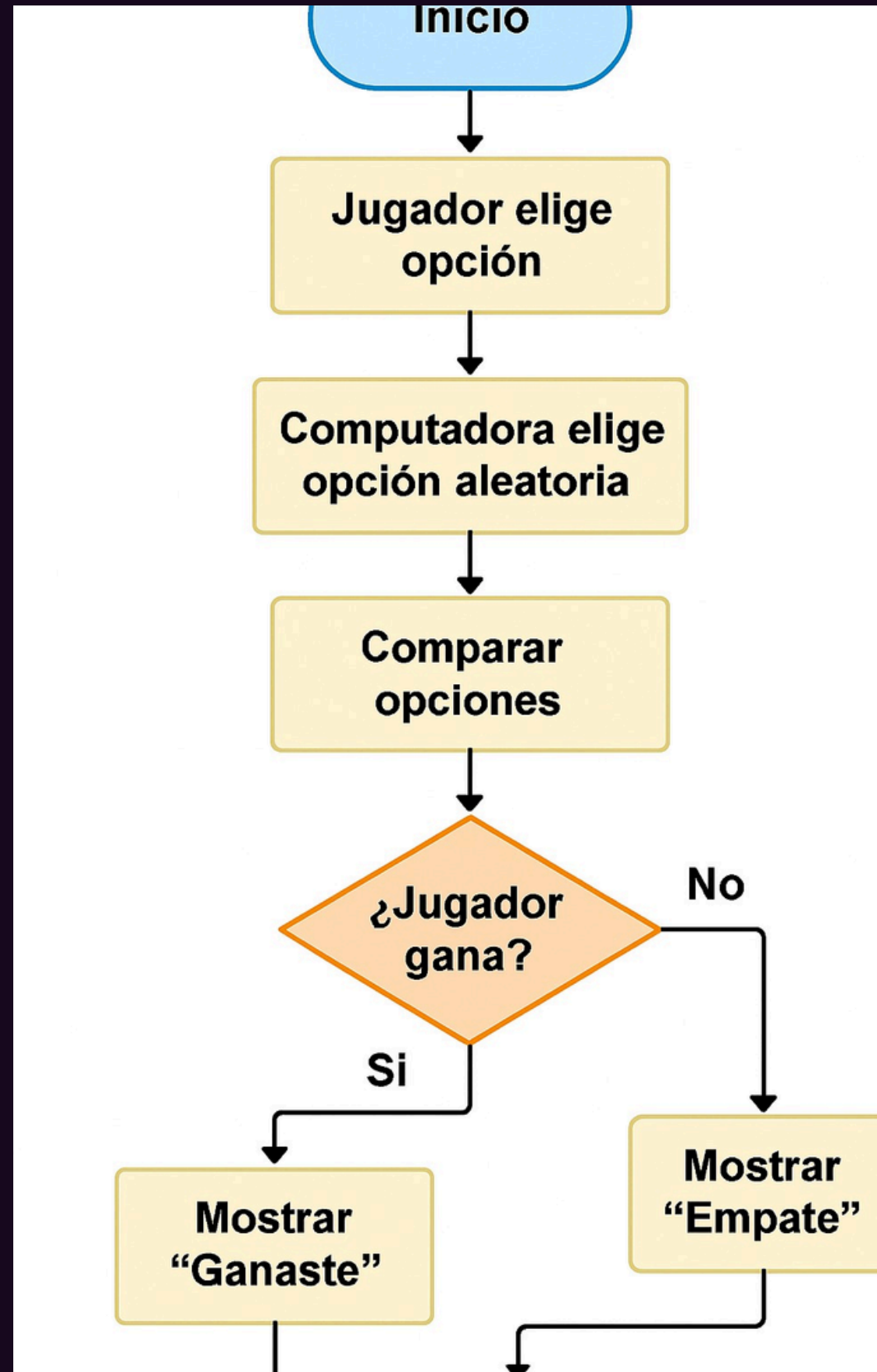
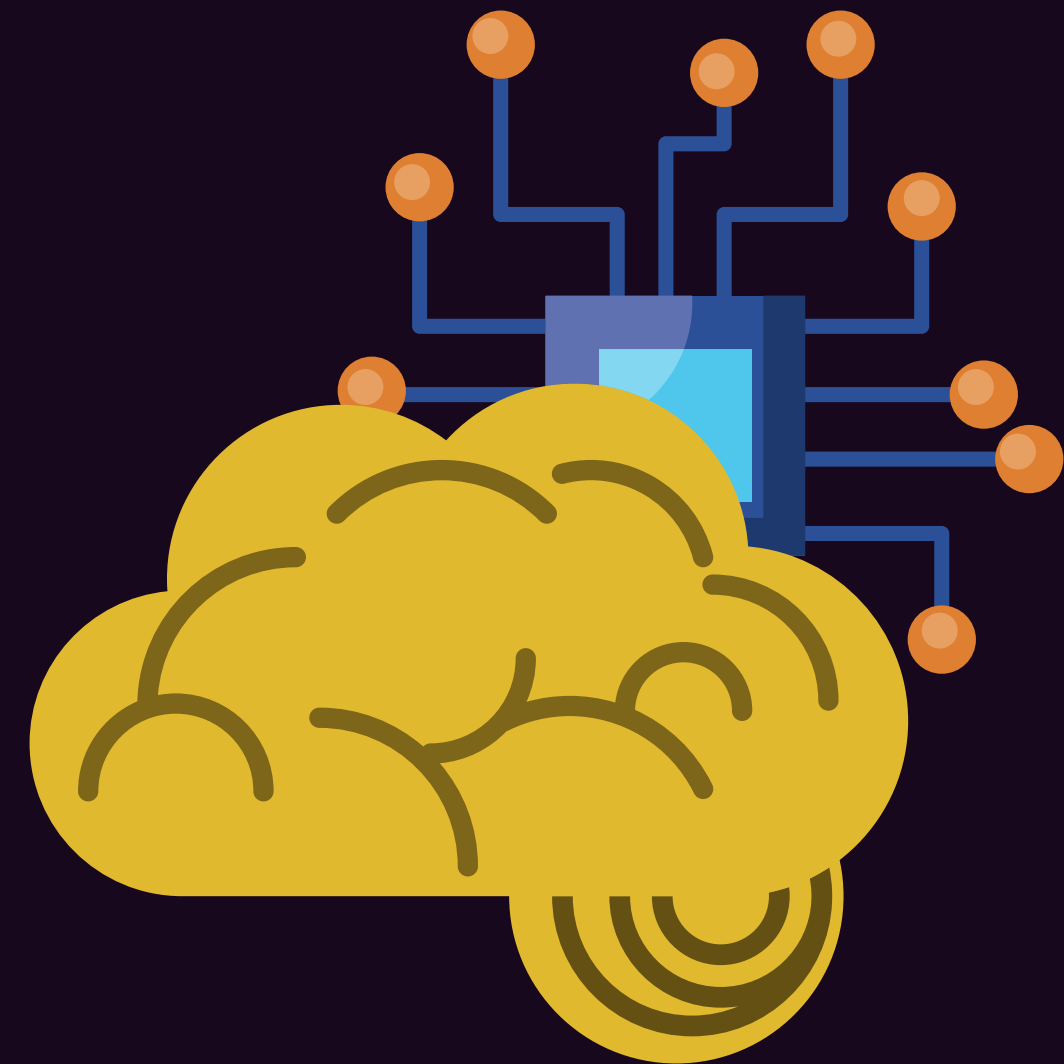
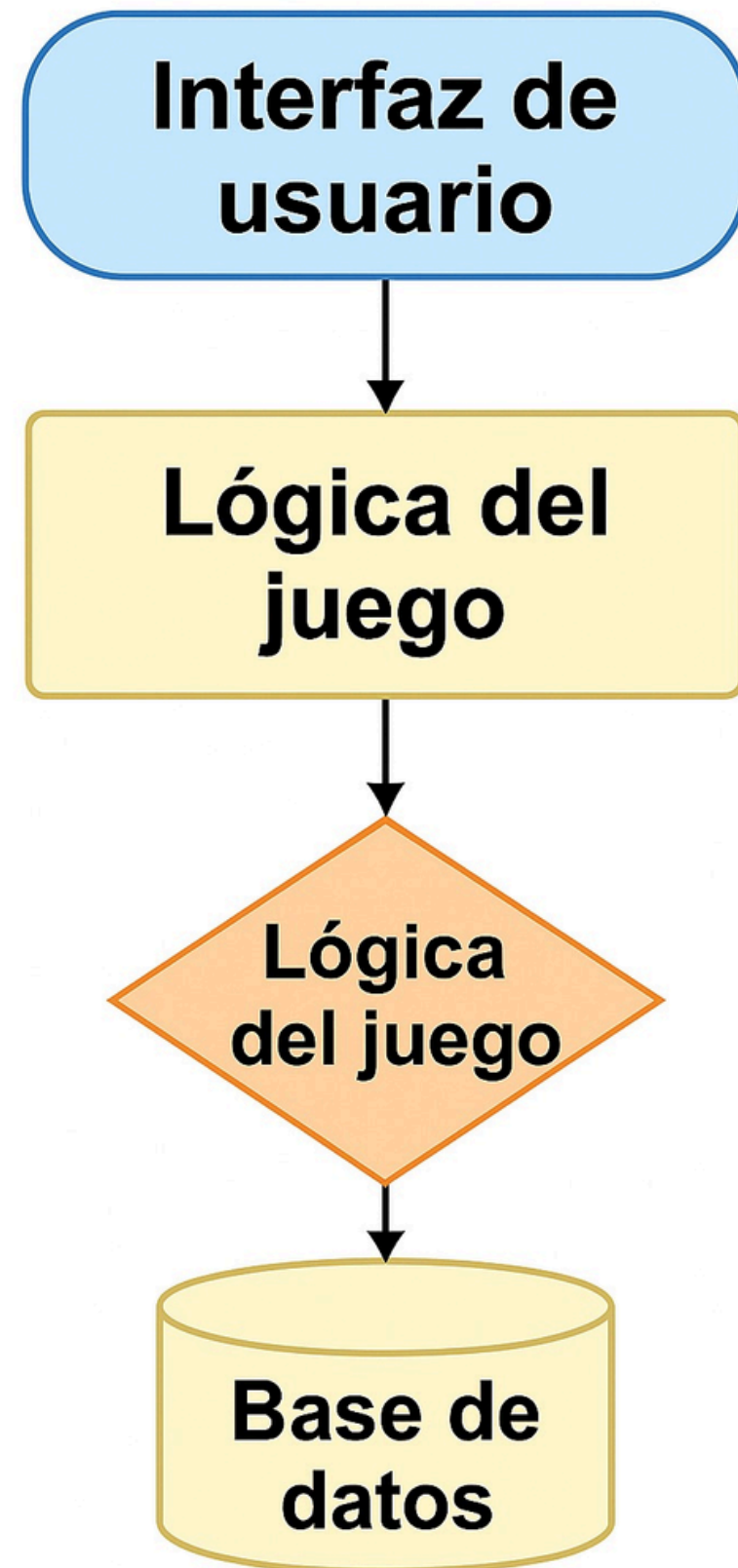
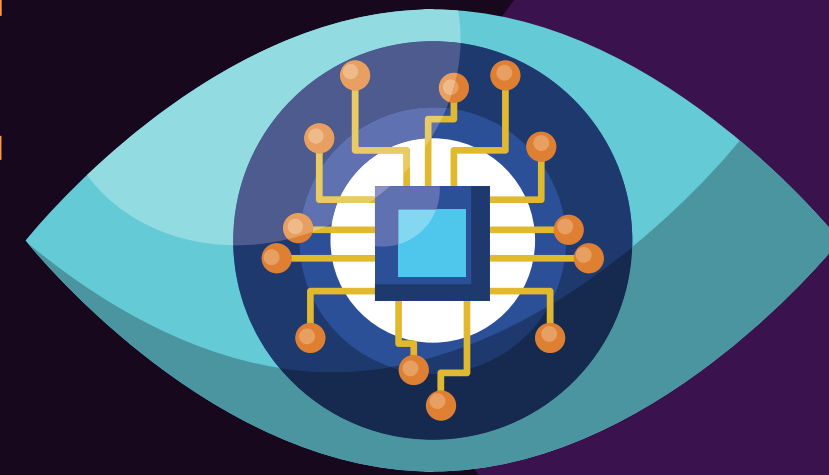


Diagrama de Arquitectura



RESULTADOS DE APRENDIZAJES



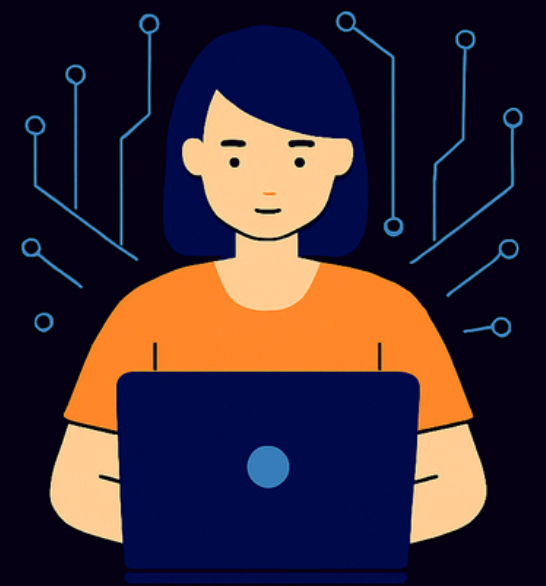
- Se comprendió cómo aplicar estructuras de control (if, elif, else).
- Se aprendió a validar entradas del usuario y manejar errores.
- Se reforzó la importancia de los diagramas para planificar antes de programar.
- Se valoró la documentación como parte esencial del desarrollo de software.
- Se identificó cómo un juego sencillo puede ser un puente para entender conceptos más complejos de programación.

Conclusiones:

- El proyecto demuestra que la programación no solo resuelve problemas técnicos, sino que también puede transformar actividades cotidianas en experiencias digitales.
- Refuerza la importancia de la informática como herramienta transversal en la ingeniería y la sociedad.
- Conecta la práctica académica con la reflexión sobre el impacto de la tecnología en el futuro, mostrando que incluso un juego simple refleja la evolución digital.
- La experiencia fomenta creatividad, pensamiento lógico y habilidades de documentación, esenciales para cualquier ingeniero.

Conclusiones

- La programación permite transformar juegos tradicionales en experiencias digitales e interactivas.
- Los videojuegos son una muestra de cómo la tecnología impacta en el entretenimiento, la educación y la cultura.
- Este proyecto contribuye a la comprensión de conceptos clave en informática y programación.



Demostración

Video mostrando:



- INICIO DEL PROGRAMA.
- ELECCIÓN DEL JUGADOR.

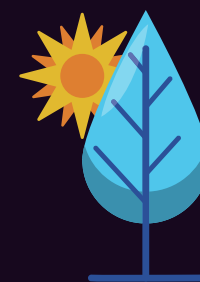
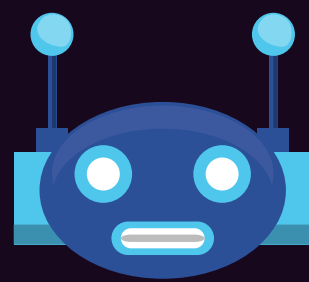
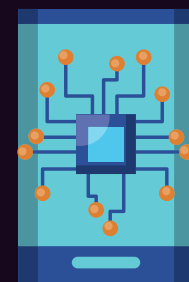
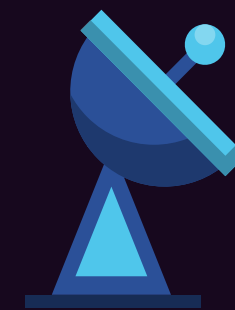
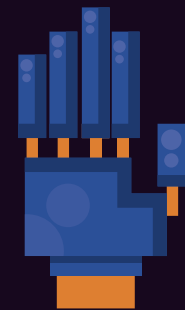
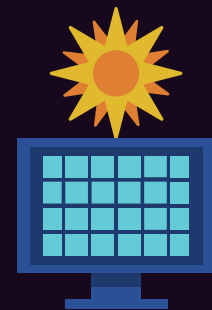
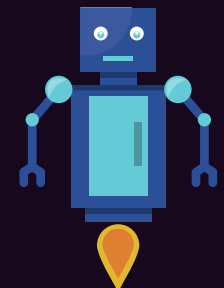
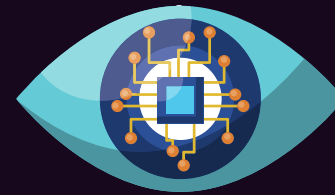
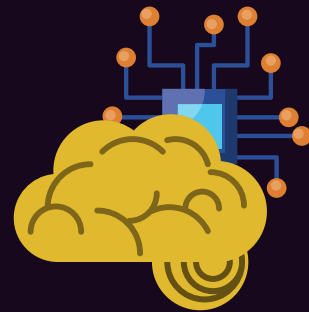
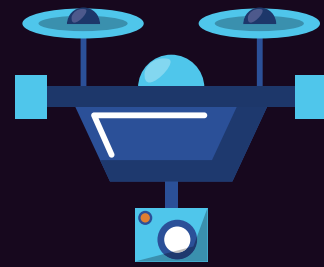
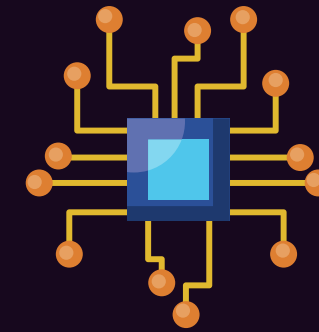
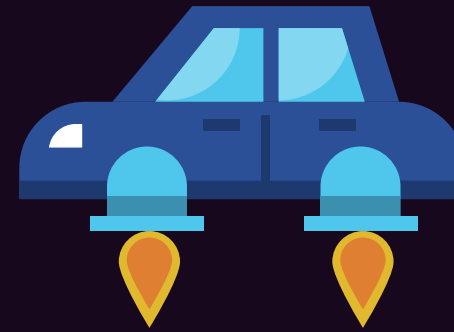
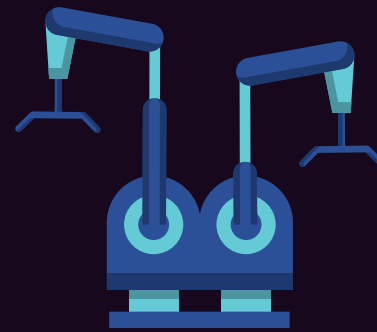
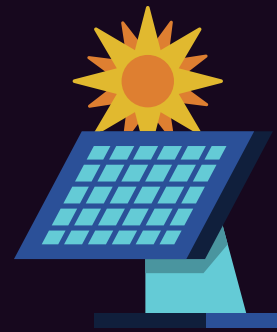


- ELECCIÓN ALEATORIA DE LA COMPUTADORA
- RESULTADO MOSTRADO EN PANTALLA.



OPCIÓN DE REPETIR O FINALIZAR.

Elementos gráficos



Bibliografía:

- Python Software Foundation. (2023). Python documentation. <https://docs.python.org/3/>
- Martínez, J. (2021). Introducción a la programación con Python. Alfaomega.
- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
- McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction (2nd ed.). Microsoft Press.
- Van Rossum, G., & Drake, F. L. (2009). The Python Language Reference Manual. Network Theory Ltd.
- Callister, W. D., & Rethwisch, D. G. (2019). Materials Science and Engineering (10th ed.). Wiley.