

```
<!--Lógica de programación-->
```

Juego tradicional
piedra, papel o
tijera

```
<Por="Edison Paucar"/>
```

```
}
```



Contenidos

01

Introducción

02

Objetivo

03

Diagrama de flujo

04

Cronología

05

Conclusiones

Introducción {

El juego de piedra, papel o tijera es un juego tradicional. En el presente trabajo se va a realizar un código para simular el juego. El trabajo consta de pseudocódigo, un diagrama de flujo y el código el cual está programado en python.

Objetivos {



PIEDRA, PAPEL Y TIJERA



Realizar la codificación del juego tradicional piedra, papel o tijera. con el apoyo del lenguaje de programación python.

Implementar al código los temarios aprendidos durante la materia de lógica de programación.

}

Diagrama de flujo

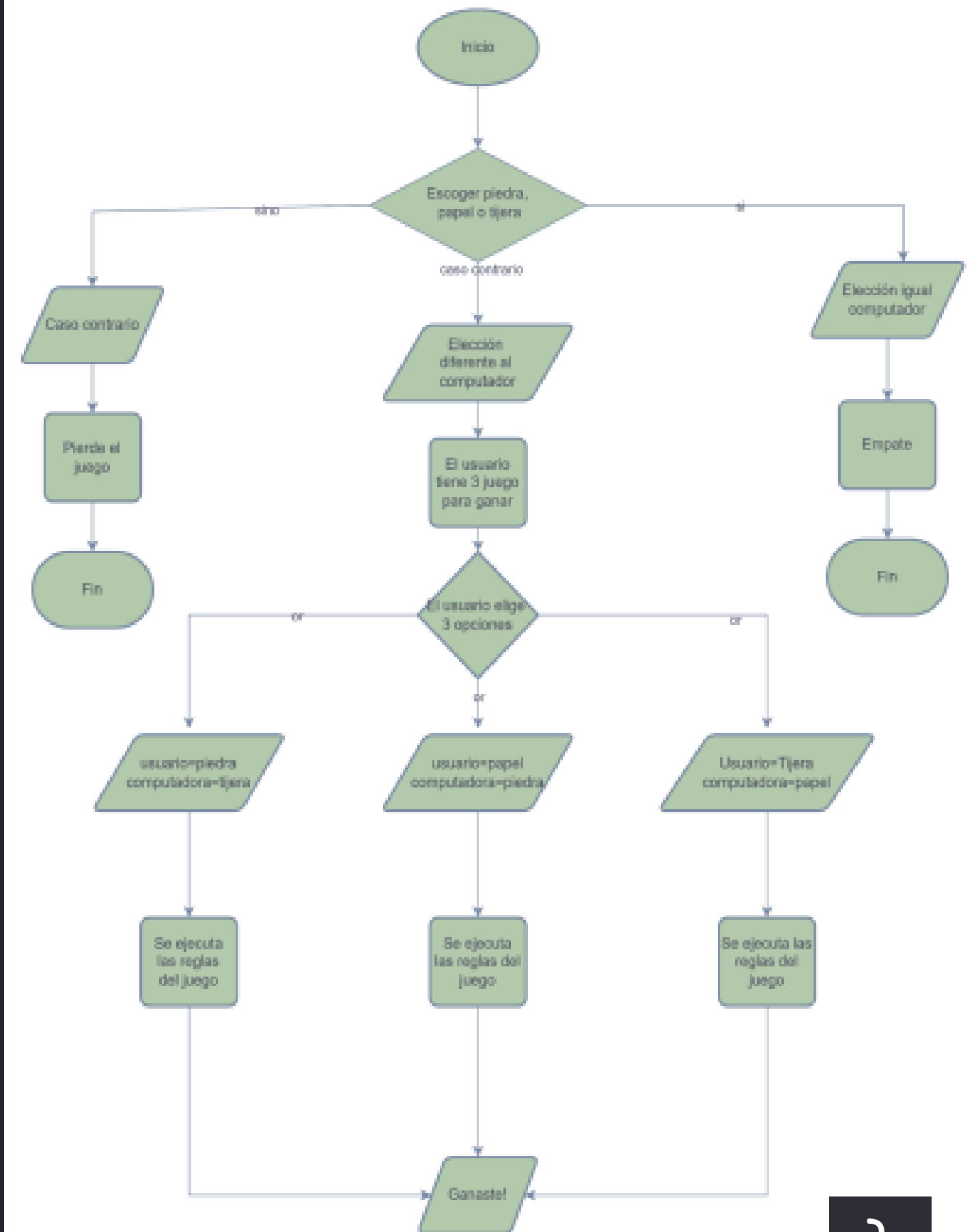
El diagrama de flujo es una representación gráfica del algoritmo del juego. El cual empieza con

Empieza con el ingresos de las opciones del juego (piedra,papel o tijera)
Si la elección del usuario es igual a la de computadora, el juego es un empate.
si la elección es diferente gana o pierde. El usuario gana si:

```
usuario=piedra computadora=tijera
usuario=papel computadora=piedra
usuario=tijera computadora=papel
Caso contario pierde el usuario
```

Muestra fragmento de código de la exportación de datos.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut a enim nec nisl ullamcorper eleifend.



}

Cronología de realización del código

Se implementó, los siguiente en el código, operaciones de comparación. Para determinar si el usuario pierde o gana la partida.

Además, se adicionaron operadores lógicos para determinar si cumple con las reglas del juego.

Se adicionaron condicionales, los cuales dividen en 3 diferentes posibles casos de juego.

Se implementaron bucles, en este caso se utilizó el bucle while.

Por último, se colocaron listas para almacenar las opciones de elección.



- Realizar el pseudocódigo y algoritmo.
- Realizar el código, implementando operaciones básicas.
- Se implementaron condicionales.
- Se adicionaron bucles y listas.

Código en python

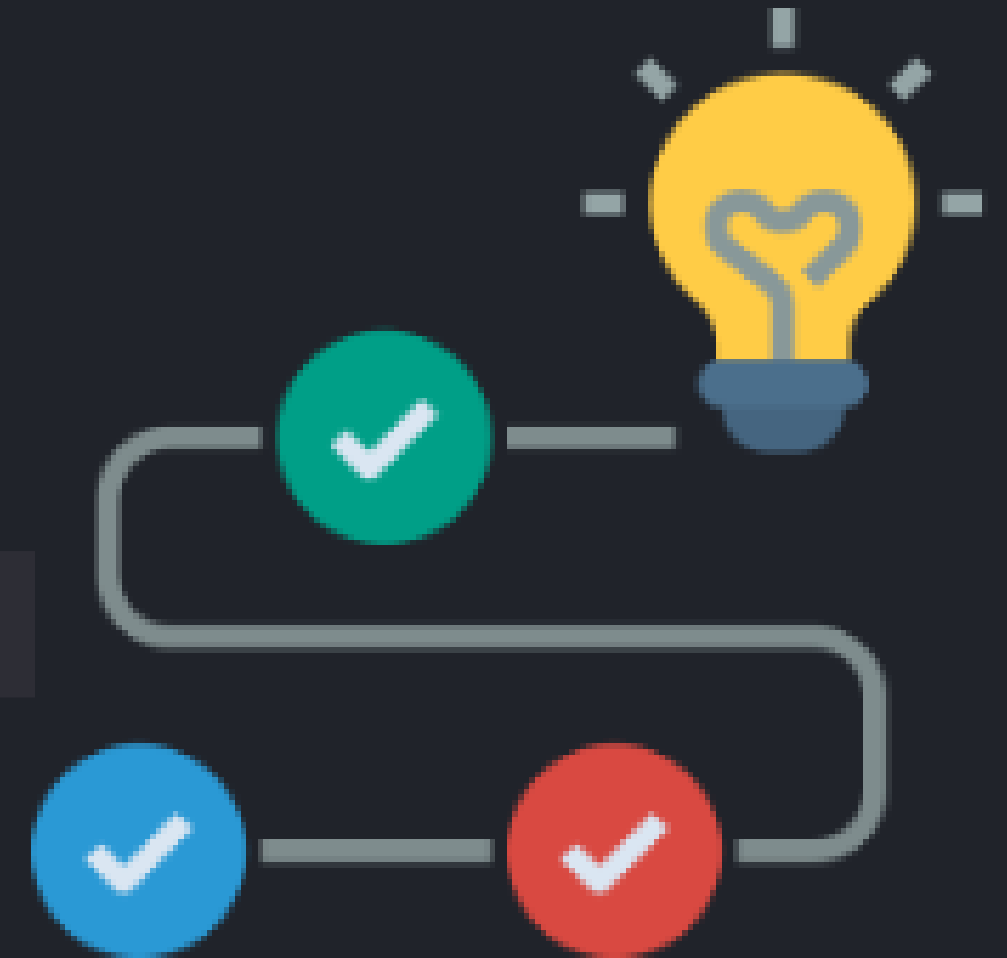
```
# Generar la elección aleatoria de la computadora
eleccion_computadora = random.choice(opciones)
print(f"La computadora eligió: {eleccion_computadora}")

# Determinar el resultado del juego
if eleccion_usuario == eleccion_computadora:
    print(";Es un empate!")
elif (eleccion_usuario == "Piedra" and eleccion_computadora == "Tijera") or \
    (eleccion_usuario == "Papel" and eleccion_computadora == "Piedra") or \
    (eleccion_usuario == "Tijera" and eleccion_computadora == "Papel"):
    print(";Ganaste esta ronda!")
    puntaje_usuario += 1 # Incrementar el puntaje del usuario
else:
    print(";Perdiste esta ronda!")
    puntaje_computadora += 1 # Incrementar el puntaje de la computadora
```

Conclusiones {

Se ejecutó el juego con el aprendizaje adquirido en lógica de programación.

Se realizaron cambios , cada semana, de acuerdo a el aprendizaje adquirido. Esto permitió obtener un programa funcional.



```
<!--Estudio Shonos-->
```

Gracias {

```
<Por="Edison Paucar"/>
```

}