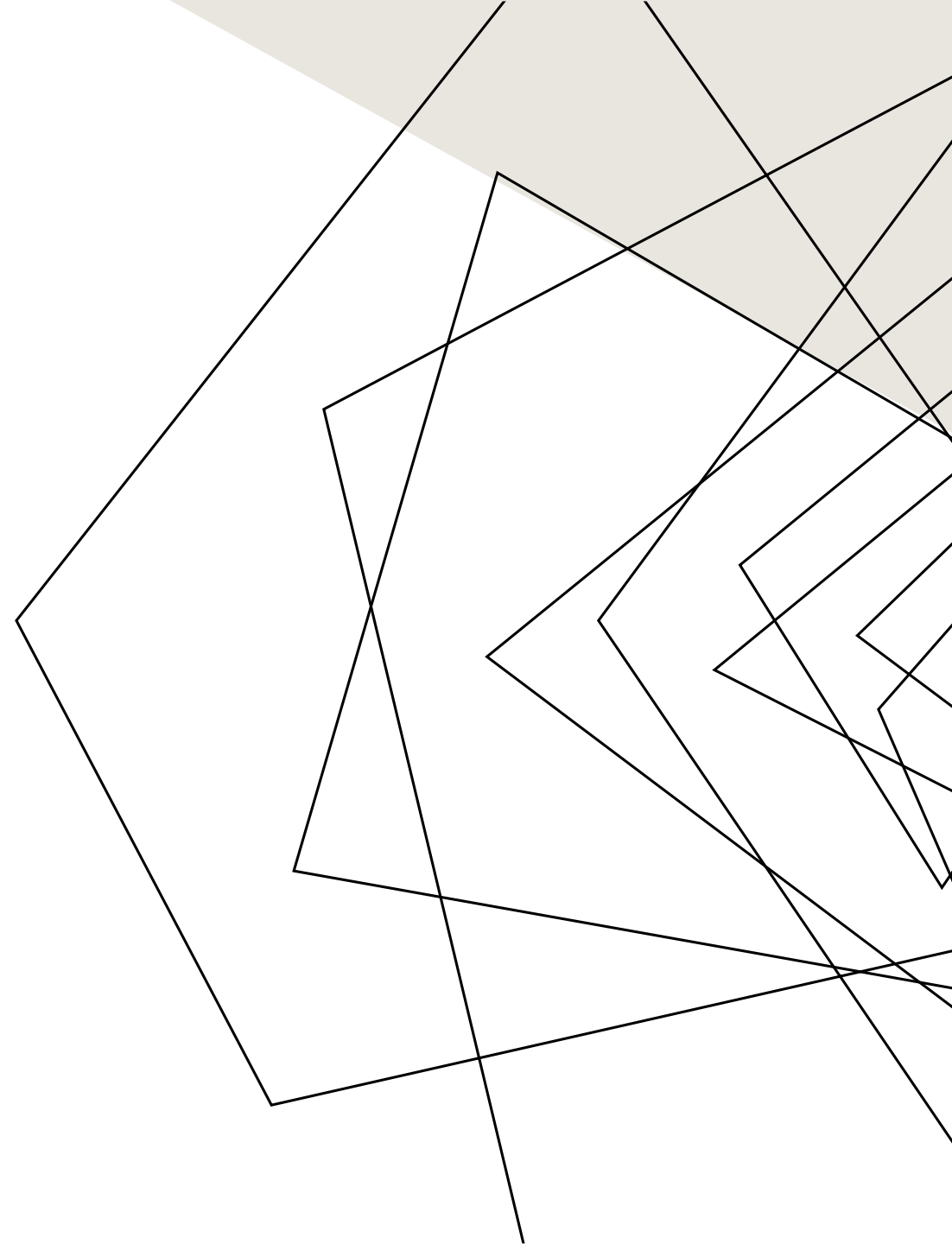




UNIVERSIDAD  
INTERNACIONAL DEL  
ECUADOR

AUTOR: CAMILO NICOLAS HERRERA  
CABEZAS.  
MATERIA: LÓGICA DE  
PROGRAMACIÓN.  
DOCENTE: MÓNICA PATRICIA  
SALAZAR TAPIA.

TEMA: PROYECTO INTEGRADOR



# CRONOGRAMA

<input type="checkbox"/>	Tarea		Estado ⓘ	Fecha ⓘ	Cronograma ⓘ	Depende de ⓘ	+
<input type="checkbox"/>	▼ SEMANA 1 - 2	+	Listo ✓	ene. 12	✓ ene. 12 - 19	REALIZADO	

<input type="checkbox"/>	Subelemento		Estado ⓘ	Fecha ⓘ	Cronograma ⓘ	Depende de ⓘ
<input type="checkbox"/>	Aprendizaje Práctico 1	+	Listo !	ene. 12	✓ -	
<input type="checkbox"/>	Aprendizaje Autónomo 1 E...	+	Listo !	ene. 19	✓ -	
<input type="checkbox"/>	▼ SEMANA 2 2	+	Listo !	ene. 19	! ene. 19 - 23	REALIZADO

<input type="checkbox"/>	Subelemento		Estado ⓘ	Fecha ⓘ	Cronograma ⓘ	Depende de ⓘ
<input type="checkbox"/>	Aprendizaje Práctico 2	+	Listo !	ene. 19	! ene. 19 - fe...	
<input type="checkbox"/>	Aprendizaje Autónomo 2 E...	+	Listo !	feb. 2	! ene. 19 - fe...	

# CRONOGRAMA

<input type="checkbox"/>	▼ SEMANA 3 2		Listo		ene.26	! ene. 26 - fe...	
<input type="checkbox"/>	Subelemento		Estado		Fecha	Cronograma	Depende de
<input type="checkbox"/>	Aprendizaje Práctico 3		Listo		ene.26	✓ feb. 26	
<input type="checkbox"/>	Aprendizaje Autónomo 2 E...		Listo		feb.2	✓ feb. 2 - 26	
<input type="checkbox"/>	▼ SEMANA 4 2		Listo		feb.5	! feb. 5 - 13	
<input type="checkbox"/>	Subelemento		Estado		Fecha	Cronograma	Depende de
<input type="checkbox"/>	Aprendizaje Práctico 4		Listo		ene.27	✓ -	
<input type="checkbox"/>	Aprendizaje Autónomo 4 E...		Listo		feb.23	✓ -	

# CRONOGRAMA

<input type="checkbox"/>	▼ SEMANA 5 2		Listo		feb.13	! feb. 13 - 18	
--------------------------	--------------	--	-------	--	--------	----------------	--

<input type="checkbox"/>	Subelemento		Estado	Fecha	Cronograma	Depende de	
<input type="checkbox"/>	Aprendizaje Práctico 5		Listo	feb.9	✓ -		
<input type="checkbox"/>	Evaluación de Aprendizaje:...		Listo	feb.9	✓ -		
<input type="checkbox"/>	▼ SEMANA 6 2		Listo		feb.16	! feb. 16 - 21	

<input type="checkbox"/>	Subelemento		Estado	Fecha	Cronograma	Depende de	
<input type="checkbox"/>	Aprendizaje Práctico 6		Listo	feb.16	! feb. 16 - 20		
<input type="checkbox"/>	Evaluación de Aprendizaje:...		Listo	feb.16	! feb. 16 - 20		

# CRONOGRAMA

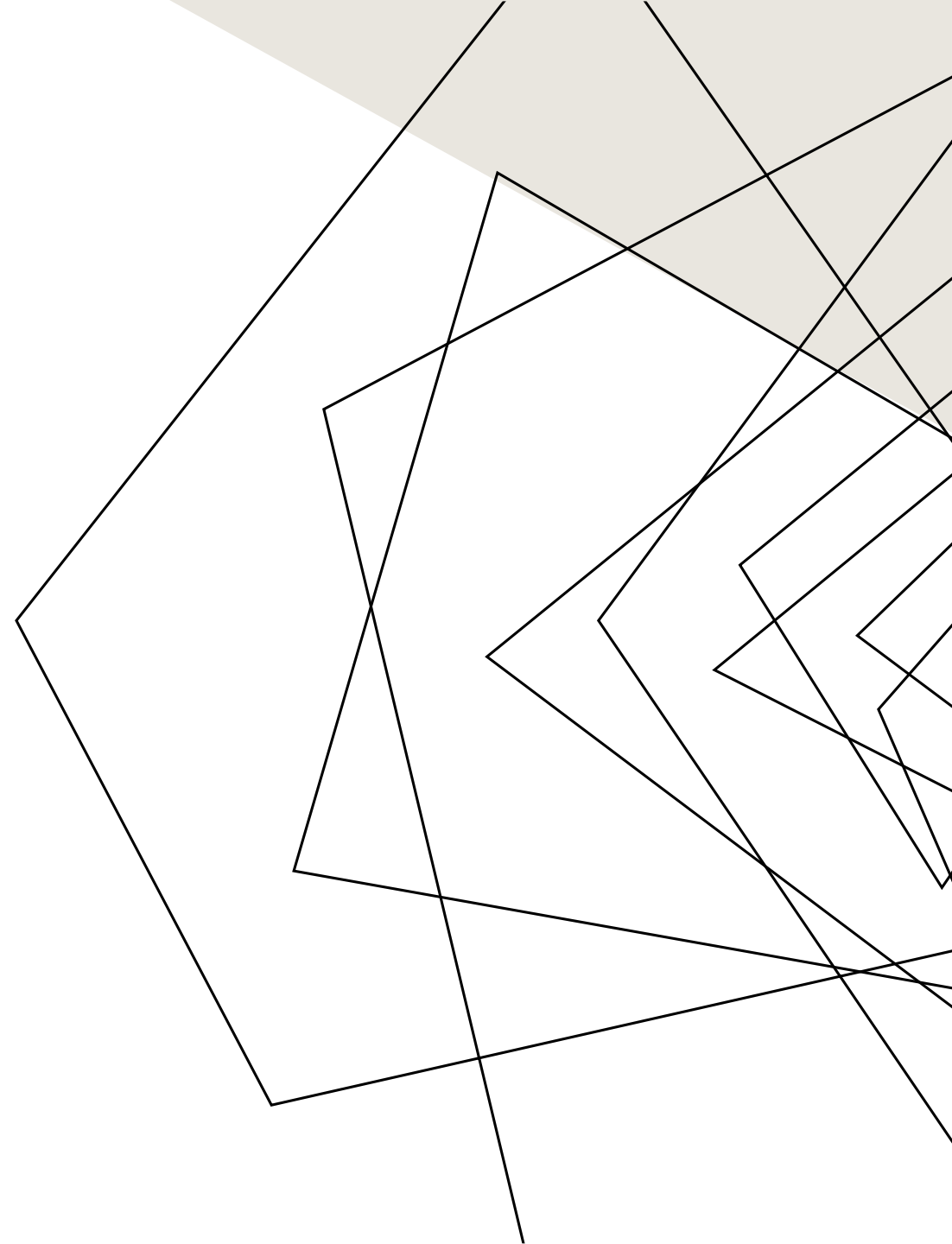
<input type="checkbox"/>	▼ SEMANA 7 2		Listo		feb. 23	feb. 21 - 23	
<input type="checkbox"/>	Subelemento		Estado		Fecha	Cronograma	Depende de
<input type="checkbox"/>	Encuesta de Acceso a Prác...		Listo		feb. 23	feb. 23 - 26	
<input type="checkbox"/>	Evaluación de Aprendizaje:...		Listo		feb. 23	feb. 23 - 26	
<input type="checkbox"/>	▼ SEMANA 8 2		Listo		feb. 23	feb. 21 - 23	
<input type="checkbox"/>	Subelemento		Estado		Fecha	Cronograma	Depende de
<input type="checkbox"/>	Aprendizaje Práctico 8		Listo		feb. 26	feb. 23 - 26	
<input type="checkbox"/>	Evaluación de Aprendizaje:...		Listo		feb. 26	feb. 23 - 26	

# INTRODUCCIÓN

En la era digital en la que vivimos, la programación se ha convertido en una habilidad esencial para desarrollar nuestras capacidades lógicas y creativas en el ámbito del desarrollo informático. Es por eso que aprender a programar no solo nos permite comprender mejor la tecnología que nos rodea, sino también innovar y crear soluciones a problemas complejos. El presente proyecto se centra en la programación del juego del ahorcado utilizando el lenguaje de Python, una excelente oportunidad para poner en práctica estos conocimientos y habilidades.

## PLANTEAMIENTO DEL PROBLEMA

El problema principal para resolver en el proyecto integrador se enfocará a cómo implementar una versión funcional del juego del ahorcado en Python. A su vez el aprendizaje de este.



## TEMAS 1 Y 2

- **Planteamiento del problema y al Entorno de la Programación:**  
Elaborar y/o diseñar el juego del ahorcado. Utilizando el lenguaje de programación Python.
- **Pasos de resolución de problemas: 6 PASOS**
  - I. Identificar el Problema. Python-Juego del Ahorcado
  - II. Comprender el Problema. {
    - Fundamentos de la Programación
    - Python
  - III. Identificar soluciones alternativas. {
    - Aprendizaje y práctica
  - IV. Seleccionar la mejor solución. {
    - Python
    - Conceptos básicos
    - Fundamentos
    - Funciones
    - Listas y Diccionarios
    - Entrada y Salida
    - Experimentación
  - V. Listar los pasos de la solución seleccionados.
  - VI. Evaluar/Probar la solución.
- **Diagrama de uso y de Arquitectura:**  
<https://github.com/CamiloHerrera454/PROYECTO-FINAL.git>
- **Análisis.**

## TEMA 3 Y 4

- **Diagrama de Flujo:**
- <https://github.com/CamiloHerrera454/PROYECTO-FINAL.git>
- **Ambiente de desarrollo:** Herramienta de Visual Studio Code, Python.
- **Condicionales:**
  - I. **If:** Evalúa una condición, si es verdadera se ejecuta el bloque de código dentro del if.
  - II. **Elif:** Se utiliza para verificar múltiples condiciones.
  - III. **Else:** Se utiliza en una cadena de condicionales para especificar un bloque de código que se ejecutará si ninguna de las condiciones anteriores es verdadera.
- **Bucles:**
  - I. **For:** Se utiliza para iterar sobre una secuencia (como una lista, tupla, diccionario, conjunto o cadena).
  - II. **While:** Se utiliza para repetir un bloque de código mientras una condición sea verdadera.

- **Operadores:**

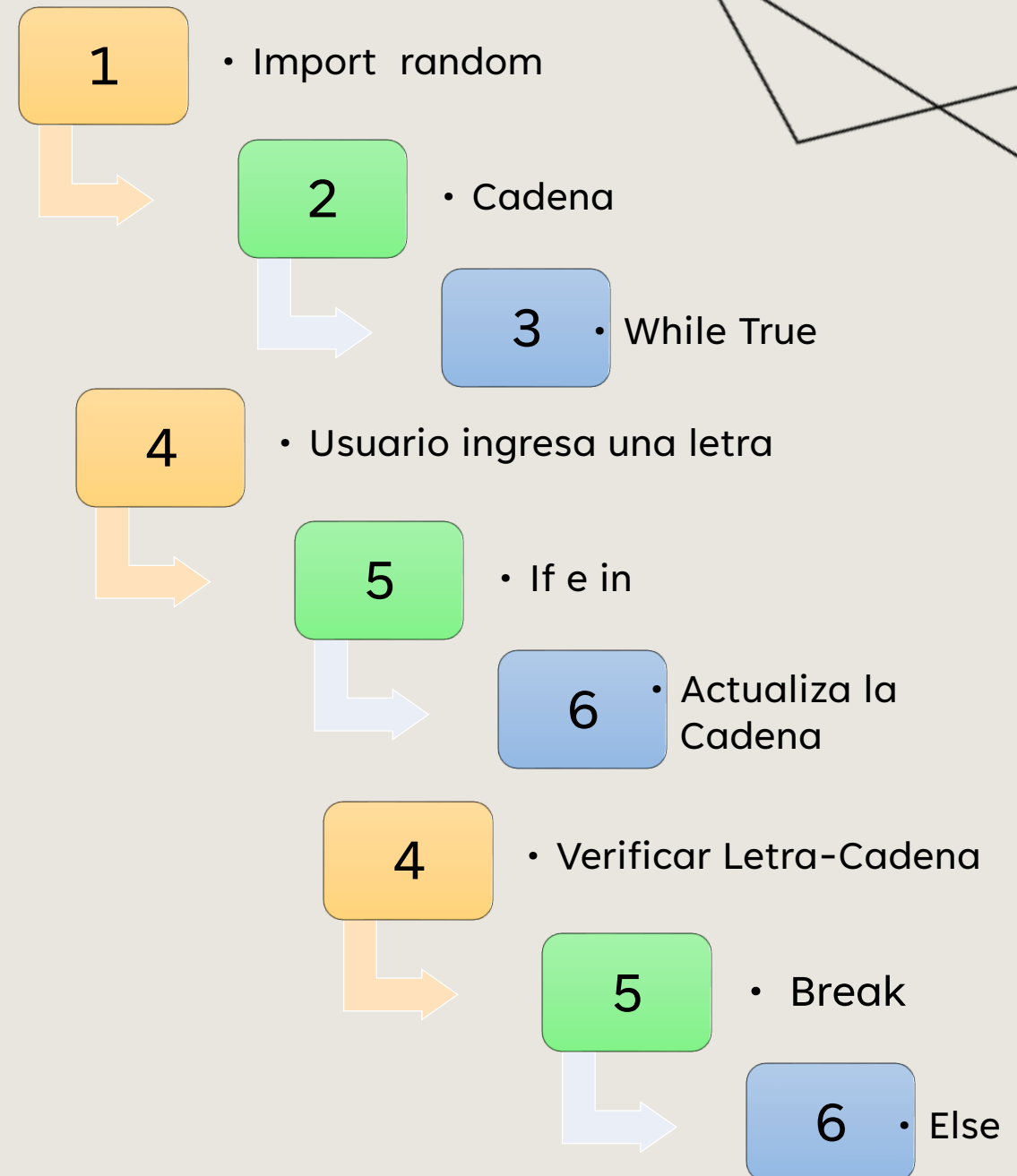


- **Análisis:**



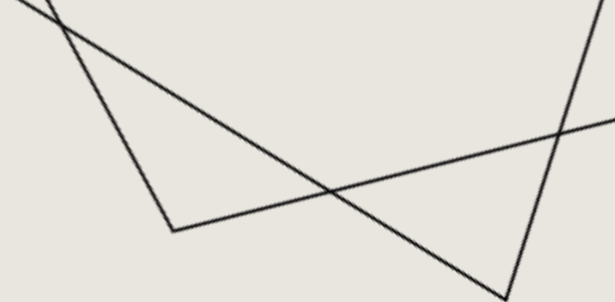
- **Estructuras de Datos y Funciones:**

- I. **Tuplas:** Conforman un tipo de estructura de datos que se utilizan para almacenar una colección ordenada e inmutable de elementos.
  - II. **Listas:** Conforman una lista es una colección ordenada y mutable de elementos, lo que significa que puedes agregar, eliminar y modificar sus elementos.
  - III. **Diccionarios:** Conforman una estructura de datos que permite almacenar pares de claves y valores.
- **Ejecución de Funciones:** Una función es un bloque de código que realiza una operación. Las funciones pueden definir opcionalmente parámetros de entrada que permiten a la persona que llama pasar argumentos a la función. Las funciones también pueden devolver valores de salida



```
1  import random #IMPORTANTE
2
3  palabras = ['pelota', 'futbol', 'computadora', 'celular', 'cuaderno', 'libro', 'dormitorio', 'cielo', 'estrella', 'estadio']
4  secreta = random.choice(palabras) #ALEATORIO
5  cadena = "-" * len(secreta) #GENERA CADENA DE GUIONES DE ACUERDO A LA PALABRA
6  print("Bienvenidos al Juego del Ahorcado")
7  intentos = 0
```

- **Importar la librería random:**
- Esto te permitirá utilizar la función `random.choice()` para seleccionar una palabra al azar de la lista, dicha lista contiene las palabras que se usarán en el juego.
- Esta línea elige una palabra al azar de la lista `palabras`.
- Cadena: Crear una cadena de guiones de la misma longitud que la palabra secreta:



```
✓ while True: #BUCLES-BREAK
    print(cadena)
    letra = input("Ingresa una letra: ") #EL USUARIO INGRESA UNA LETRA
```

- El bucle while True: se usa para que el juego continúe ejecutándose hasta que el jugador gane o pierda. Este tipo de bucle es conocido como un bucle infinito, ya que se repetirá indefinidamente a menos que se interrumpa con una instrucción break.palabra secreta:

```
if letra in secreta: #VERIFICA SI LA LETRA ESTA EN LA PALABRA SECRETA
    for i in range(len(secreta)):
        if secreta[i] == letra: #SI LA LETRA EN LA POSICION I ESTA EN LA LETRA QUE INGRESE EL USUARIO
            cadena = cadena[:i] + letra + cadena[i+1:] #ACTUALIZA LA CADENA
    if cadena == secreta: #VERIFICA SI LA CADENA COINCIDE CON LA PALABRA SECRETA
        print(f"Felicidades has ganado el juego, la palabra secreta era {secreta}")
        break #ROMPE EL BUCLE INFINITO
```

- Explicación: Si la letra ingresada está en la palabra secreta, el código actualiza la cadena de guiones con la letra en la posición correcta.
- Si la cadena de guiones coincide con la palabra secreta, el jugador ha ganado el juego y se rompe el bucle.
- Si la letra no está en la palabra secreta, se incrementa el contador de intentos y se dibuja una parte del ahorcado.
- Si el contador de intentos llega a 6, el jugador ha perdido el juego y se muestra la palabra secreta.

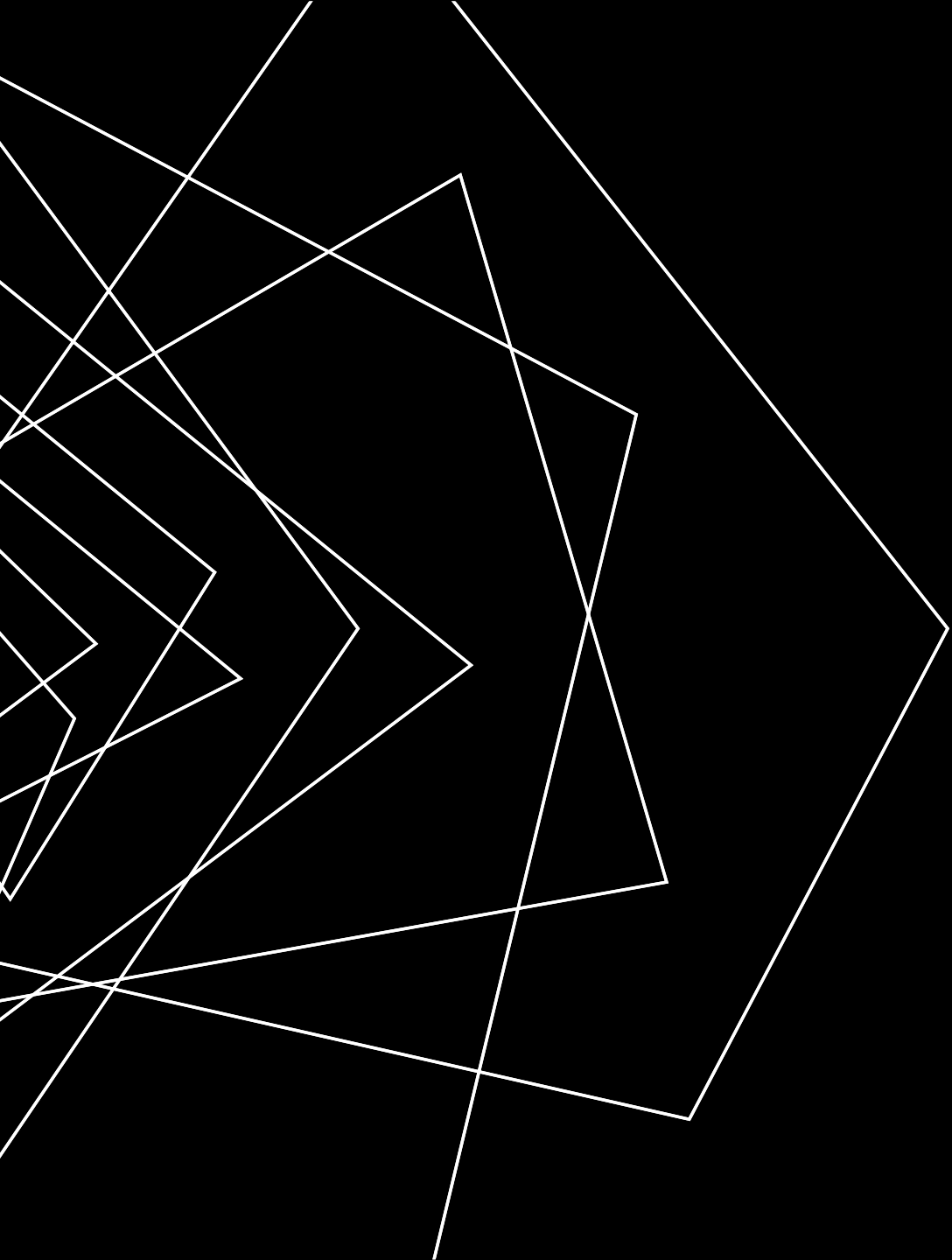
```
else:
    intentos += 1
    if intentos == 1:
        print(" 0")
    elif intentos == 2:
        print(" 0")
        print(" /")
    elif intentos == 3:
        print(" 0")
        print(" /|")
    elif intentos == 4:
        print(" 0")
        print(" /|\\")
    elif intentos == 5:
        print(" 0")
        print(" /|\\")
        print(" /")
    elif intentos == 6:
        print(" 0")
        print(" /|\\")
        print(" / \\")
        print(f"Has perdido el juego, la palabra secreta era {secreta}")
        break
```

- Cada vez que el usuario ingresa una letra incorrecta, se incrementa el contador de intentos en 1. Esto lleva un registro de cuántos intentos fallidos ha tenido el jugador.
- Dependiendo del número de intentos fallidos, se va dibujando el ahorcado paso a paso.
- Cuando el jugador ha fallado 6 veces y ha perdido el juego. Se imprime un mensaje indicando que ha perdido y revela la palabra secreta. Además, la instrucción `break` termina el bucle, deteniendo el juego.



## CONCLUSIÓN

Aprender a programar no solo nos permite comprender mejor la tecnología que nos rodea, sino también innovar y crear soluciones a problemas complejos. Este proyecto se centra en la programación del juego del ahorcado utilizando el lenguaje de Python, brindando una excelente oportunidad para poner en práctica estos conocimientos y habilidades. El objetivo principal del proyecto es elaborar y diseñar el juego del ahorcado utilizando Python, lo cual implica aplicar conceptos básicos y avanzados de programación.



# GRACIAS

Camilo Nicolas  
Herrera Cabezas  
25/02/2025