

PROYECTO INTEGRADO

JUEGO DEL AHORCADO

Pasos para resolver un problema



Problema : Juego del ahorcado

Identificamos el problema

¿Como se va a desarrollar el juego ?

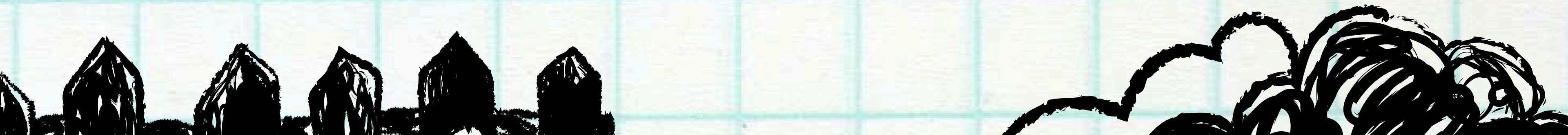
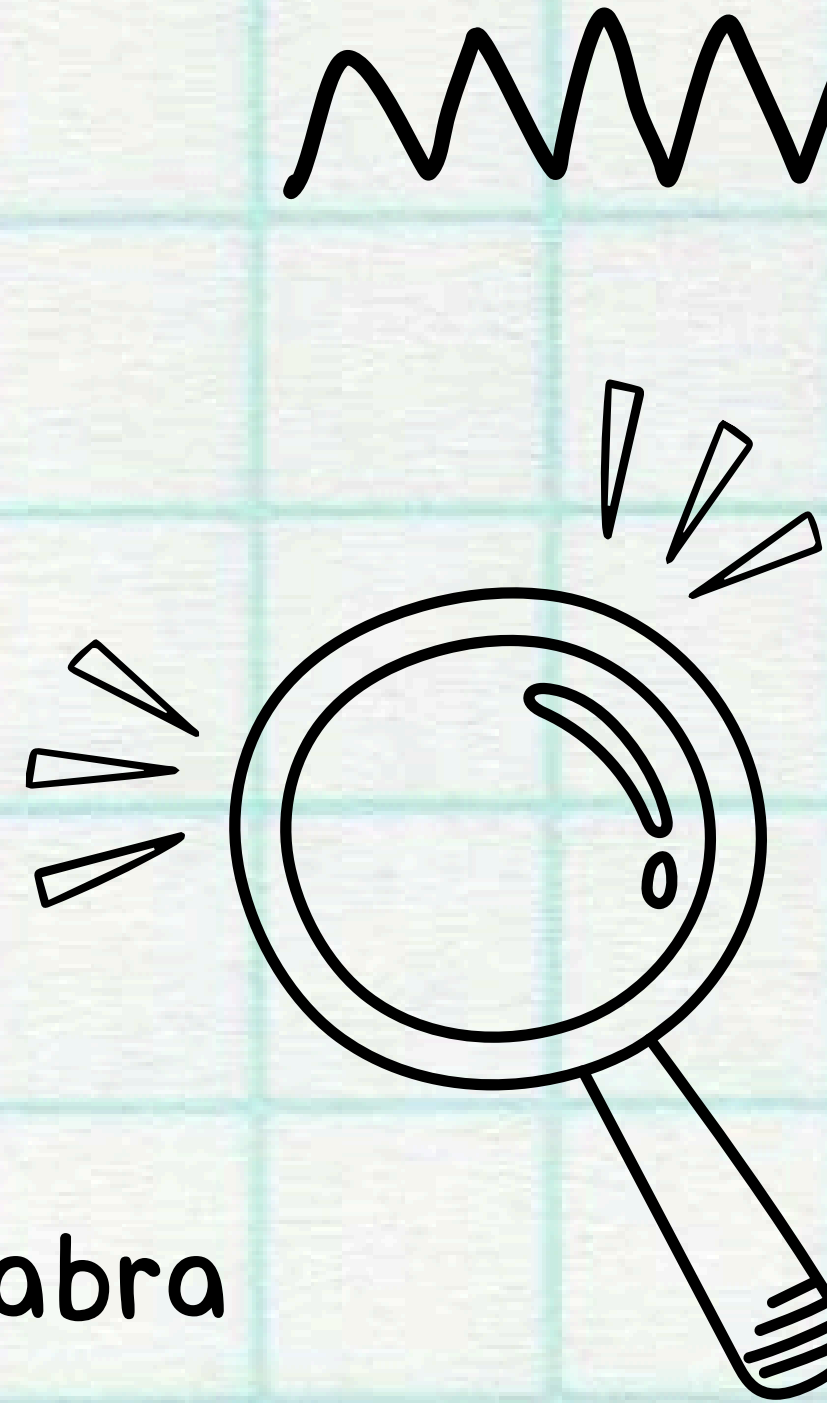
Simple de entender y facil de jugar

¿Cuantos jugadores va tener uno o mas ?

Es de un solo usuario que tiene que completar la palabra

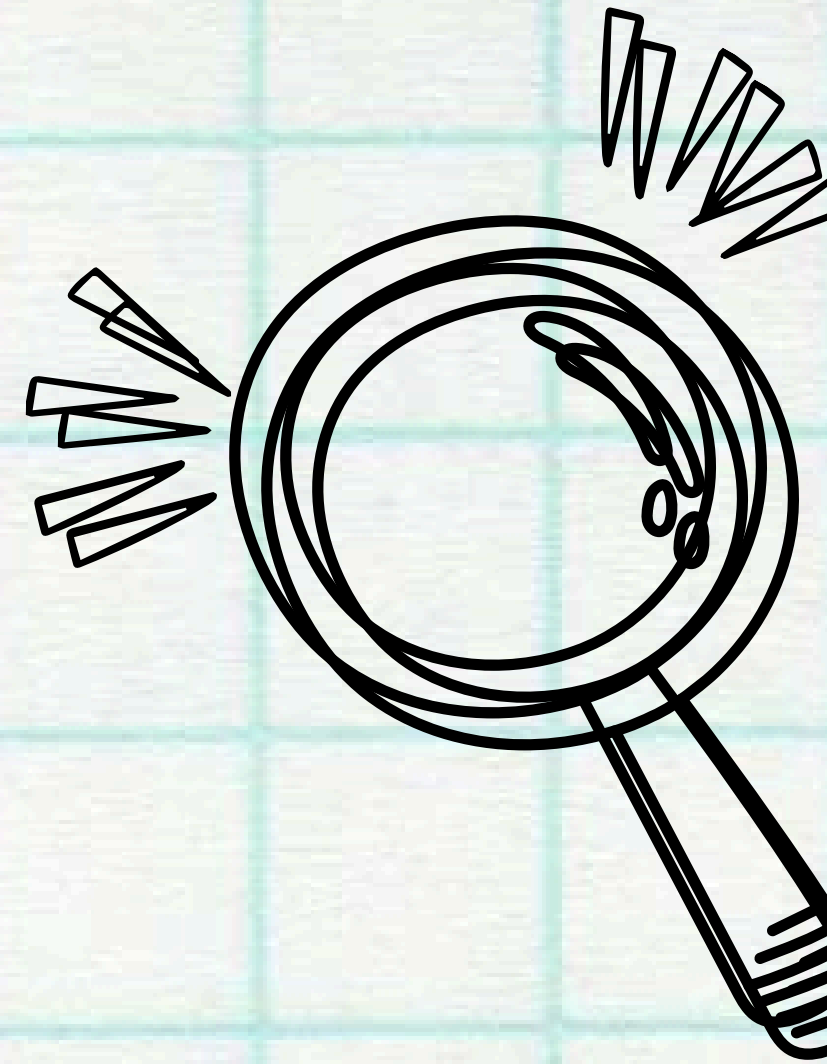
¿Cuantos intentos va a tener el usuario ?

Solo va a tener tres intentos antes de perder



Comprender el problema

Se va desarrollar mediante un código,
posterior a eso crearemos un listado de
funciones que se van a utilizar.

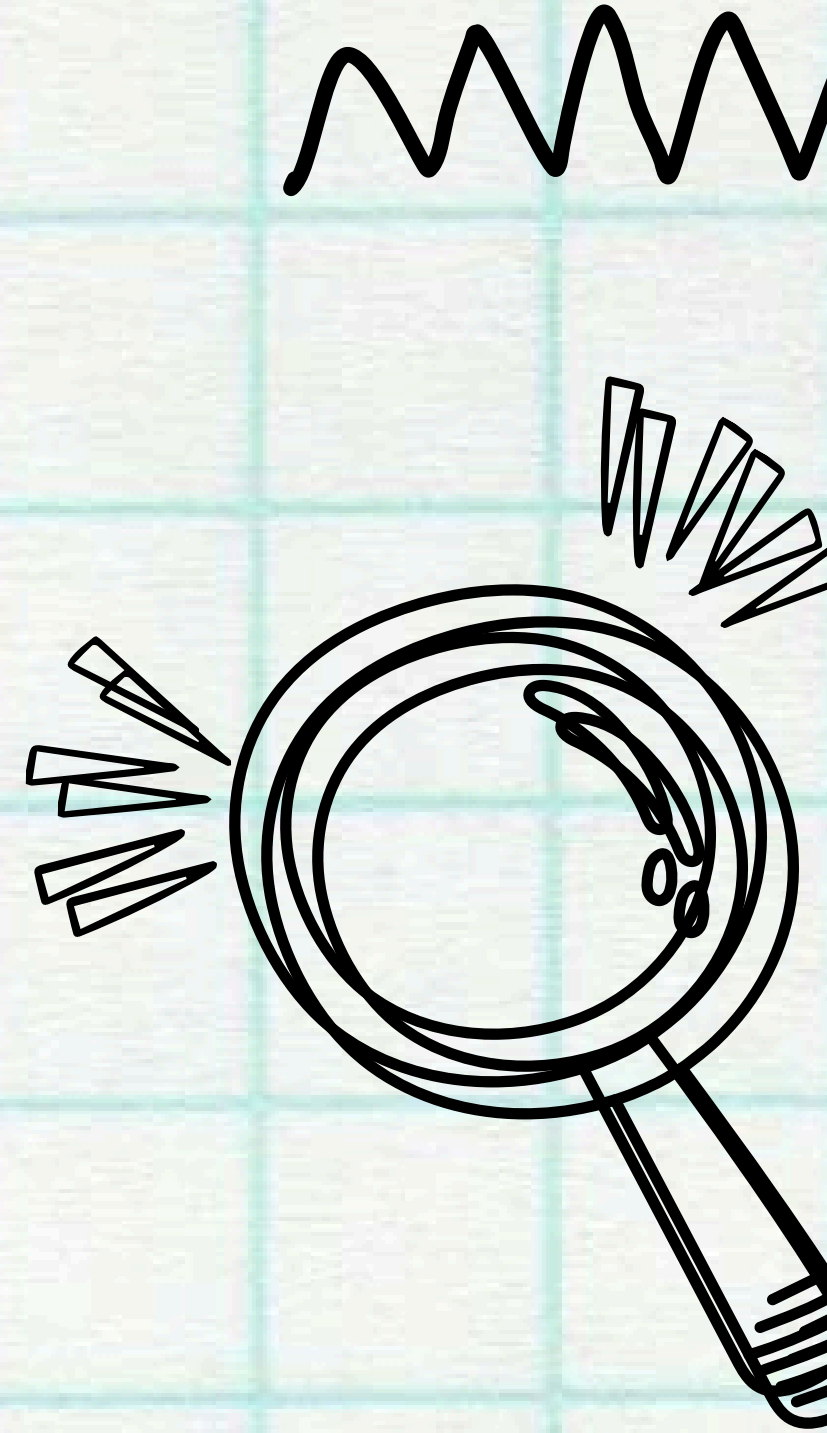


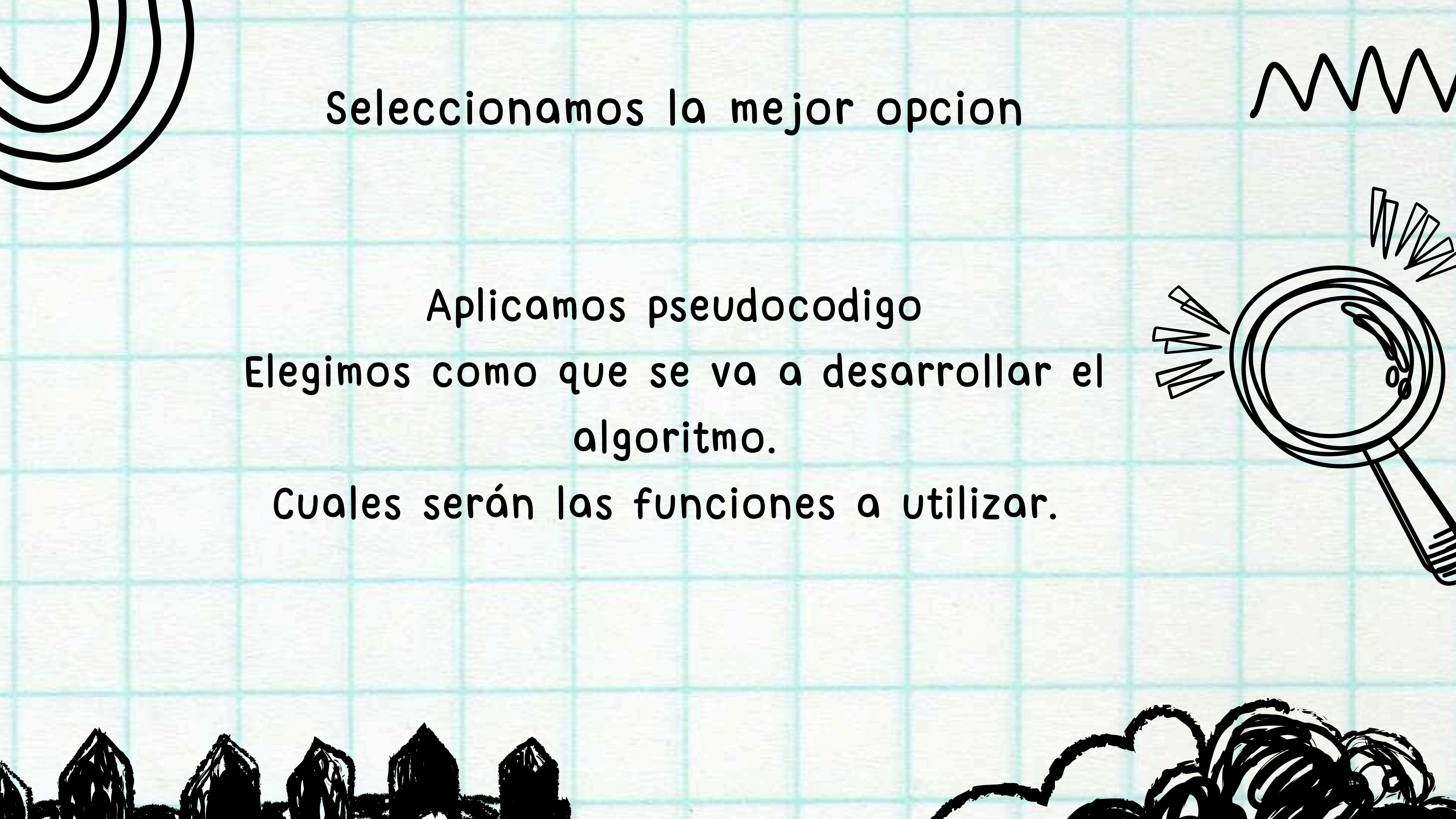


Indicar soluciones alternativas

Aqui aplicaremos pseudocódigo

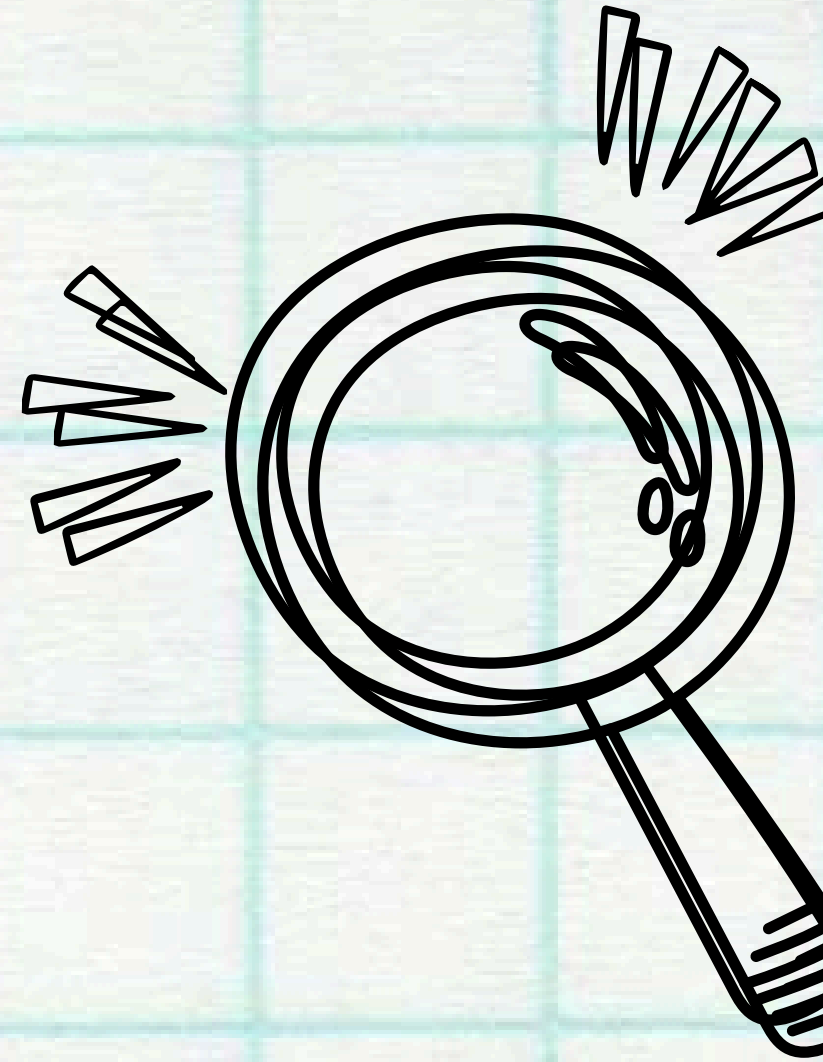
```
5 def obtener_palabra_aleatoria(lista_palabras):  
6     """Selecciona una palabra aleatoria de la lista."""  
7     indice_aleatorio = random.randint(0, len(lista_palabras) - 1)  
8     return lista_palabras[indice_aleatorio]  
9  
10 def mostrar_tablero(palabra_oculta, letras_adivinadas):  
11     """Muestra el tablero del juego."""  
12     display = ""  
13     for letra in palabra_oculta:  
14         if letra in letras_adivinadas:  
15             display += letra
```

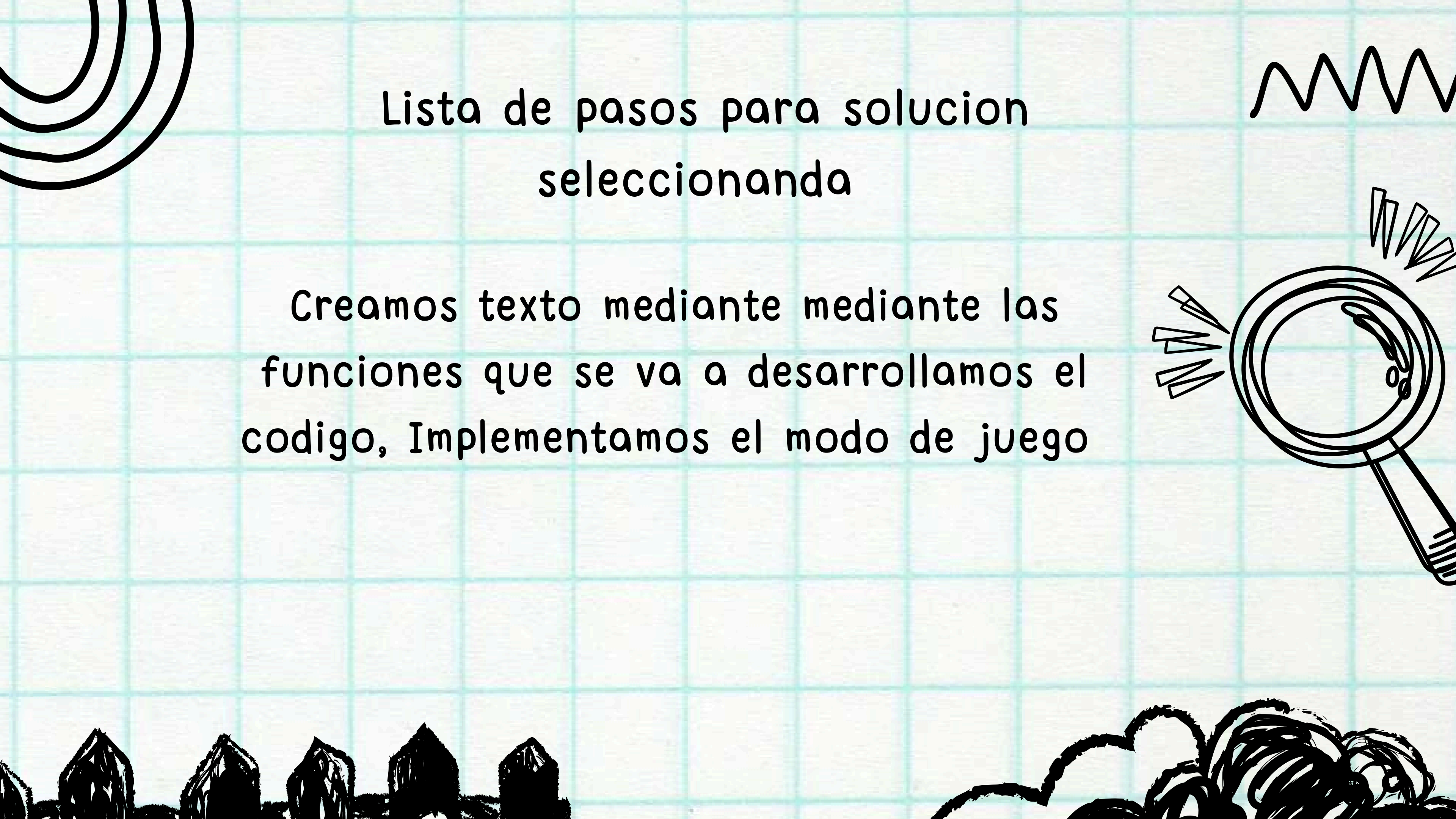




Seleccionamos la mejor opcion

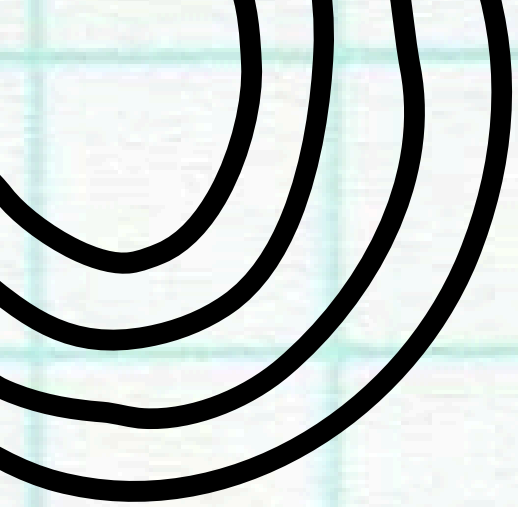
Aplicamos pseudocodigo
Elegimos como que se va a desarrollar el
algoritmo.
Cuales serán las funciones a utilizar.





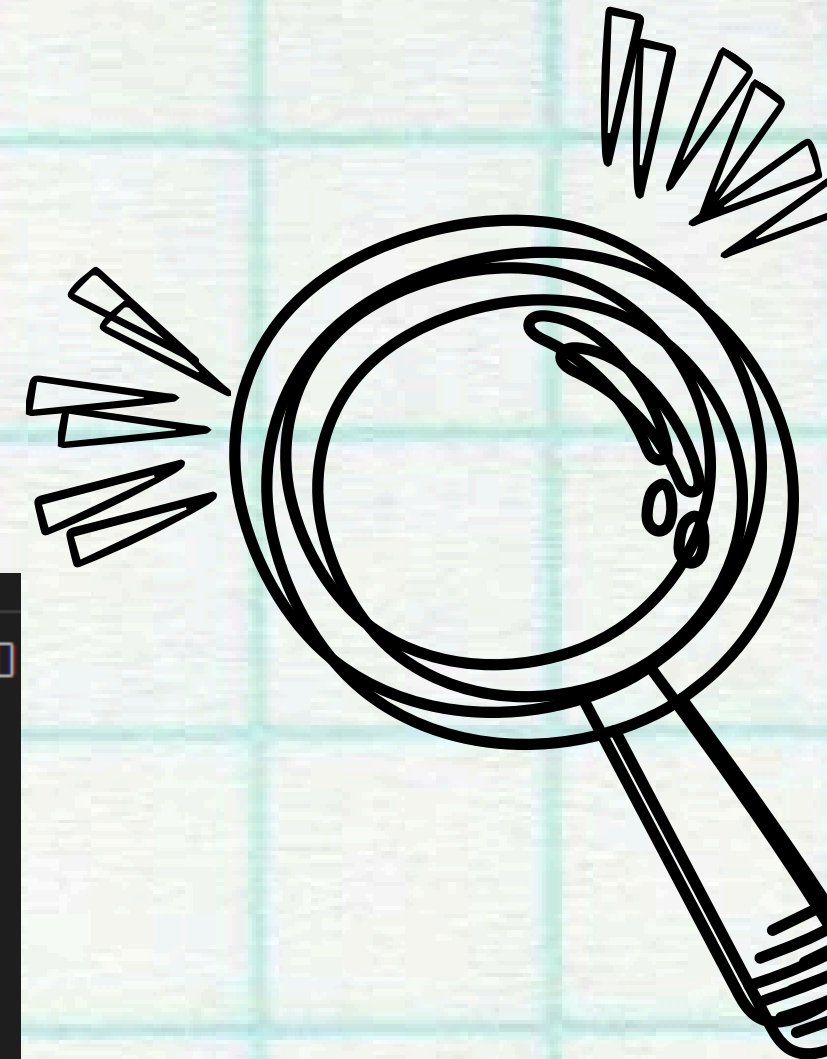
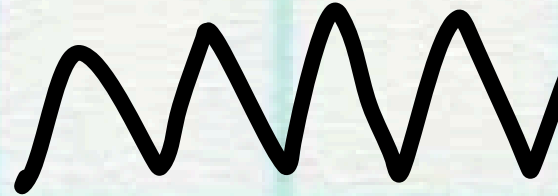
Lista de pasos para solución seleccionanda

Creamos texto mediante mediante las
funciones que se va a desarrollamos el
codigo, Implementamos el modo de juego



Evaluar la solución

Aqui ejecutamos nuestro codigo



```
LA DE DEPURACIÓN  TERMINAL  PUERTOS  Python + v []  
o, r, t, y  
á en la palabra.  
o, p, r, t, y  
á en la palabra.  
do la palabra: python  
/Users/USUARIO/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/USUARIO/OneDrive/Imágenes/Escritorio/numeros enteros.py"
```



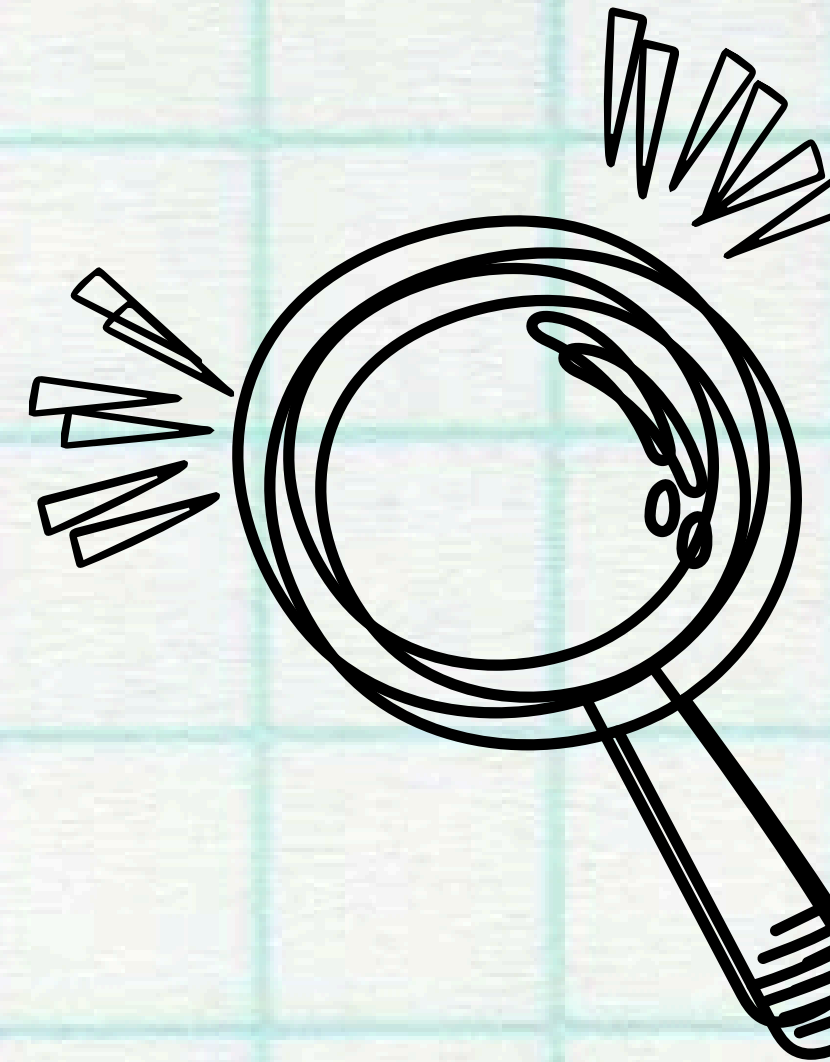
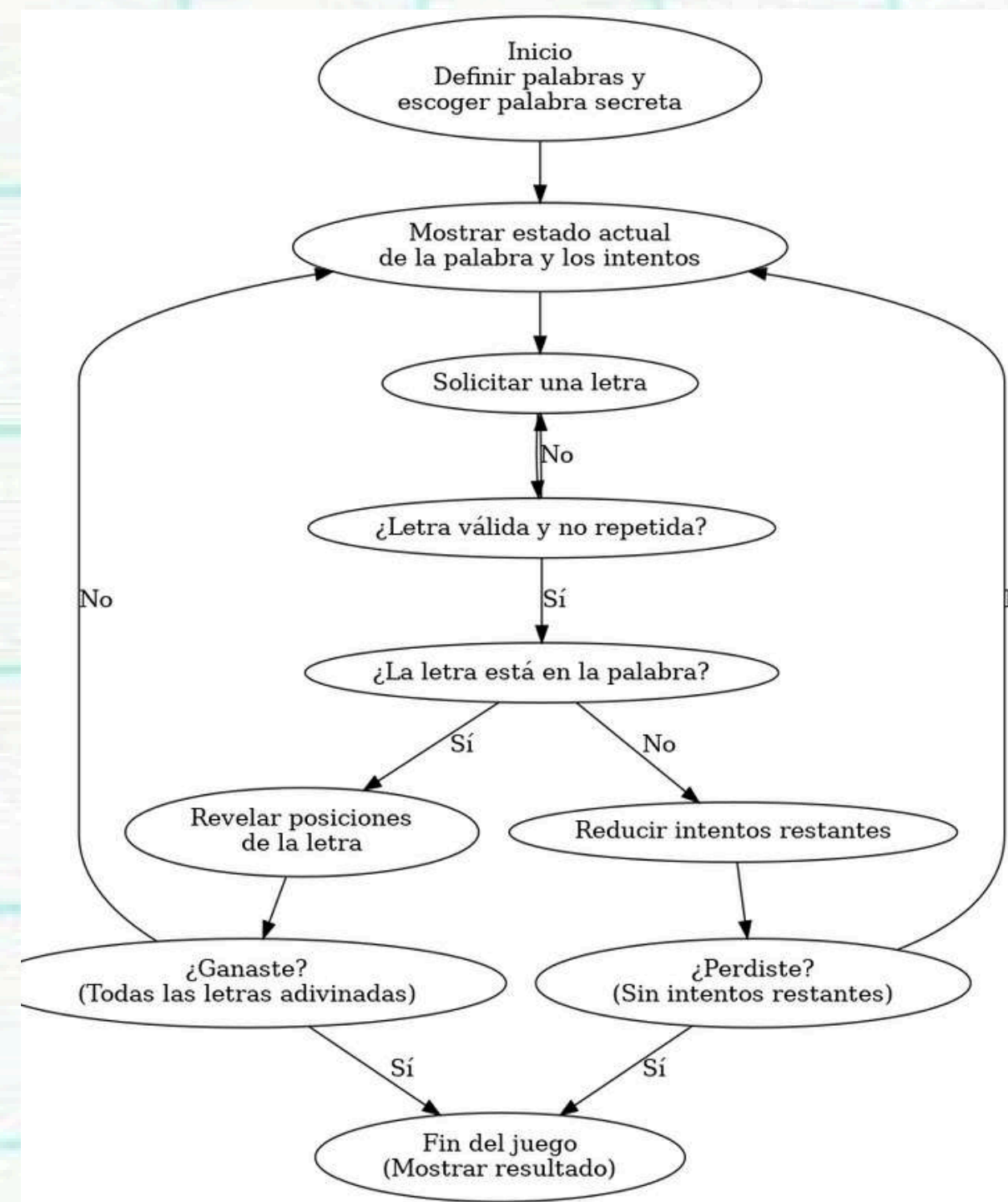
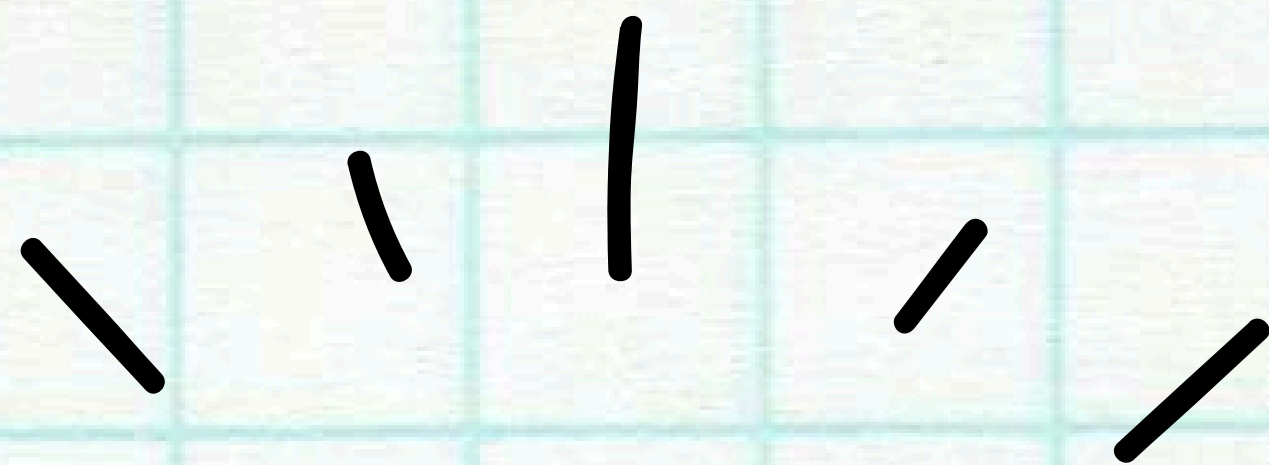
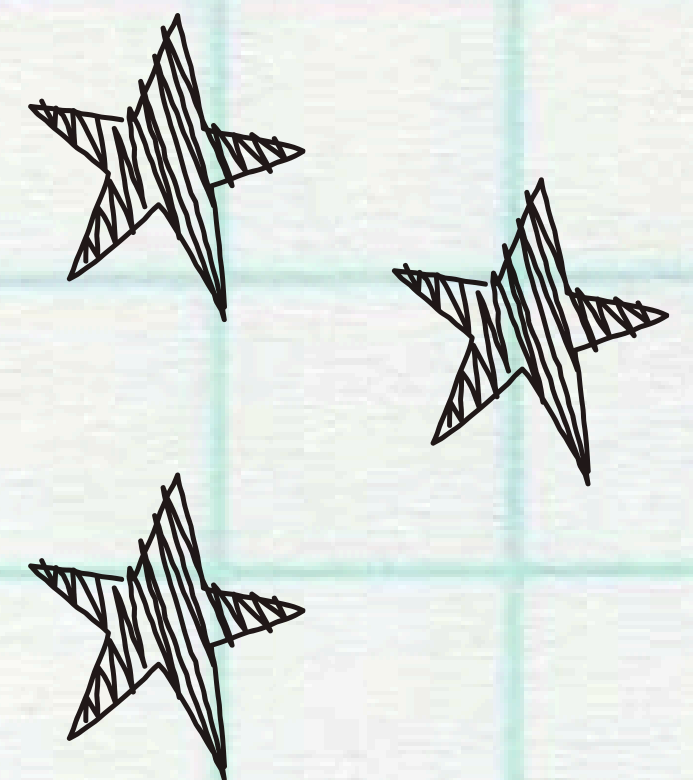
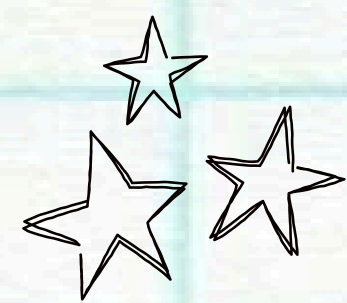
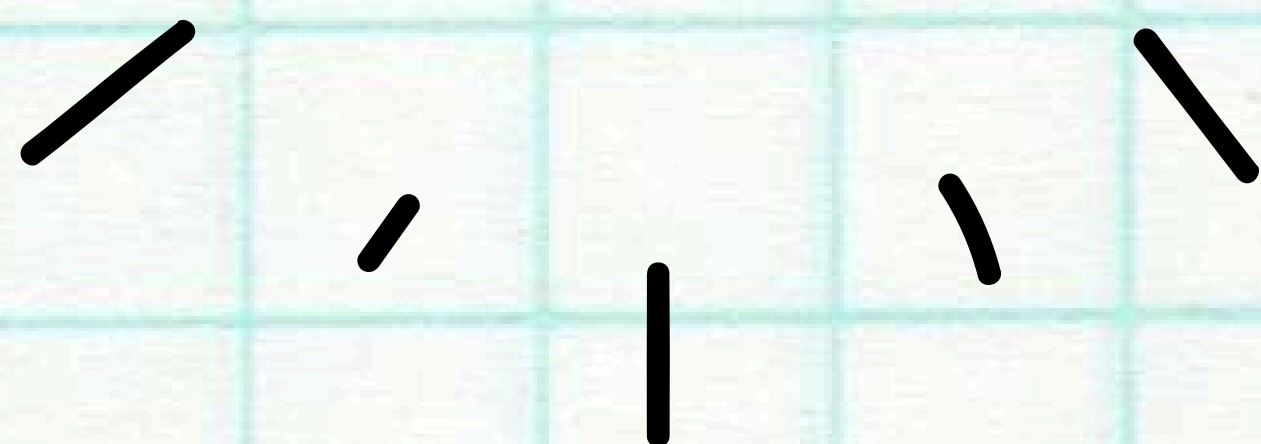


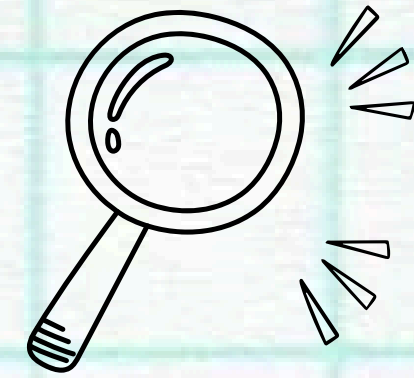
Diagrama de flujo.



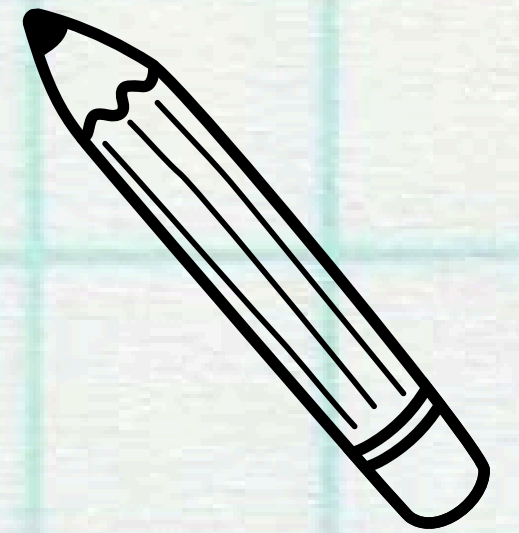


FUNCIONES





CONDICIONAL "if,



El "if" es una función que en el lenguaje de programación significa si,

El "ELIF" es es el contrario de los dos no es ni "SI" ni "NO"

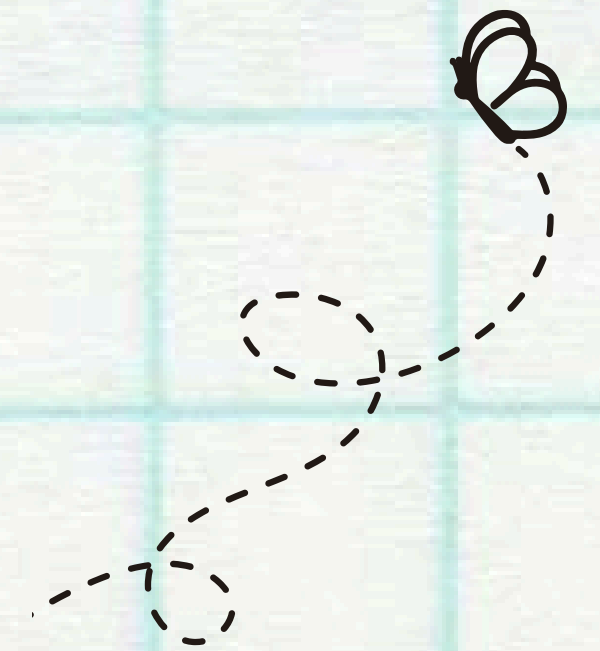
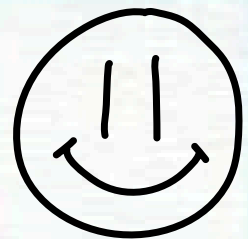
El "else" es lo contrario de "IF" en este caso sería no

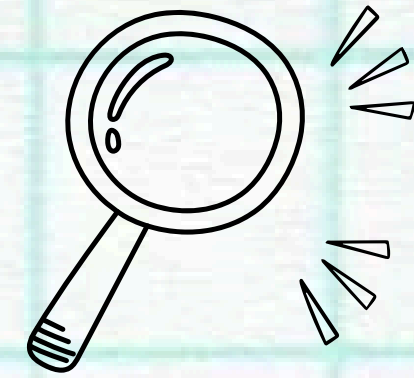
```
continue

if letra in letras_adivinadas:
    print("Ya has intentado esa letra")
    continue

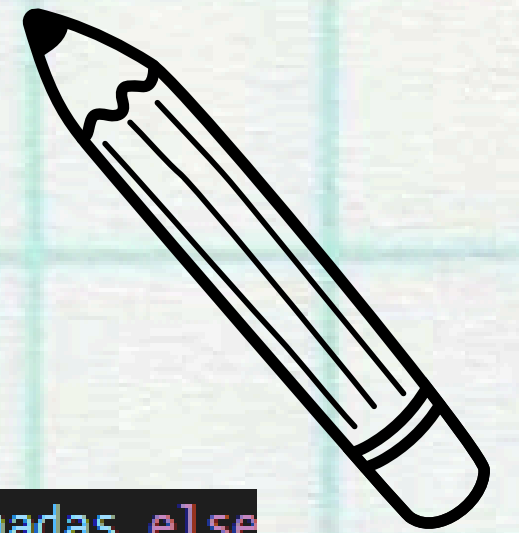
letras_adivinadas.add(letra)

if letra in palabra_secreta:
    print(";Bien hecho! La letra está")
    if set(palabra_secreta).issubset(letras_adivinadas):
        print(f"\n;Felicidades! Has adivinado la palabra")
        break
    else:
        print("La letra no está en la palabra")
        intentos_restantes -= 1
```





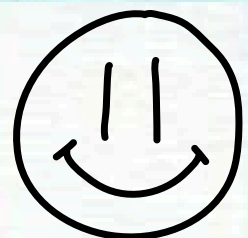
BUCLAS "WHILE"

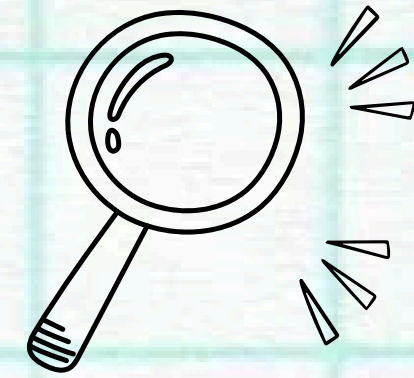


EL "WHILE" es un bucle de una sentencia verdadera, que ir acompañada por un parámetro "DEF" este nombra a una función

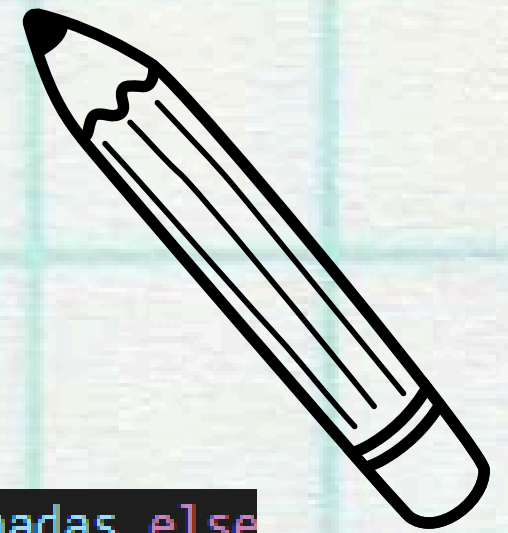
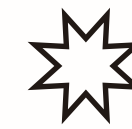


```
return " ".join([letra if letra in letras_adevinadas else  
def jugar_ahorcado():  
    print("Bienvenido al juego del Ahorcado!")  
    palabras = ["python", "programacion", "ahorcado", "desarr  
    palabra_secreta = random.choice(palabras)  
    letras_adevinadas = set()  
    intentos_restantes = 6  
  
    while intentos_restantes > 0:  
        print(f"\nPalabra: {mostrar_palabra(palabra_secreta,  
        print(f"Intentos restantes: {intentos_restantes}")  
        print(f"Letras adivinadas: {'', '.join(sorted(letras_a  
  
        letra = input("Introduce una letra: ").lower()
```



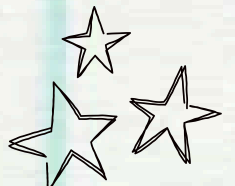
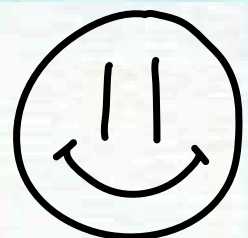


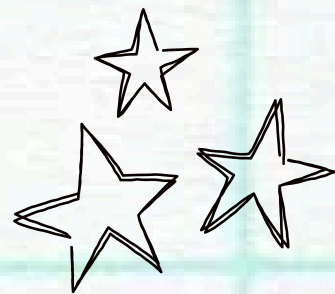
BUCLAS "FOR"



El bucle "FOR" en programación te permite repetir un bloque de código un número de veces y realizar operaciones en cadena

```
return " ".join([letra if letra in letras_adivinadas else  
def jugar_ahorcado():  
    print("Bienvenido al juego del Ahorcado!")  
    palabras = ["python", "programacion", "ahorcado", "desarr  
    palabra_secreta = random.choice(palabras)  
    letras_adivinadas = set()  
    intentos_restantes = 6  
  
    while intentos_restantes > 0:  
        print(f"\nPalabra: {mostrar_palabra(palabra_secreta,  
        print(f"Intentos restantes: {intentos_restantes}")  
        print(f"Letras adivinadas: {'', '.join(sorted(letras_a  
  
        letra = input("Introduce una letra: ").lower()
```





¡GRACIAS!

