

Somos Lógica

PROYECTO AHORCADO



ÍNDICE

1 Introducción

3 Contexto

5 Definición

7 Next Steps

2 Equipo

4 Planificación - Desarrollo

6 Demo

8 Conclusiones

8 Dudas



INTRODUCCIÓN

Este proyecto recrea el juego del Ahorcado en Python, permitiendo adivinar palabras letra a letra antes de agotar los intentos. Ideal para principiantes, combina lógica de programación básica (condicionales, bucles, listas) con una experiencia interactiva, brindando una práctica completa y divertida de conceptos fundamentales.

EQUIPO SOMOS LÓGICA



Julia Marín Salas

Scrum Master



Alba Ventas Gallego

Scrum team



Elena Rivas Merchán

Scrum team



Brigitte Manrique Báez

Scrum team

Entidad dedicada a fomentar la metodología IT en colegios como herramienta clave para impulsar el desarrollo de habilidades críticas , el fomento de la creatividad y la mejora de las habilidades de los estudiantes para el futuro

CONTEXTO



El colegio público Emilio Lluch, de la provincia de Valencia nos contacta aprovechando la semana de Halloween para desarrollar un proyecto educativo llamado el **Juego del Ahorcado** mediante el cual acercamos a los niños al mundo de la tecnología.

FASES



1. Definición
2. Planificación y desarrollo
3. Diseño
4. Pruebas e implementación {Demo}
5. Next steps
6. Conclusiones



PLANEACIÓN Y DESARROLLO

Investigación y diseño

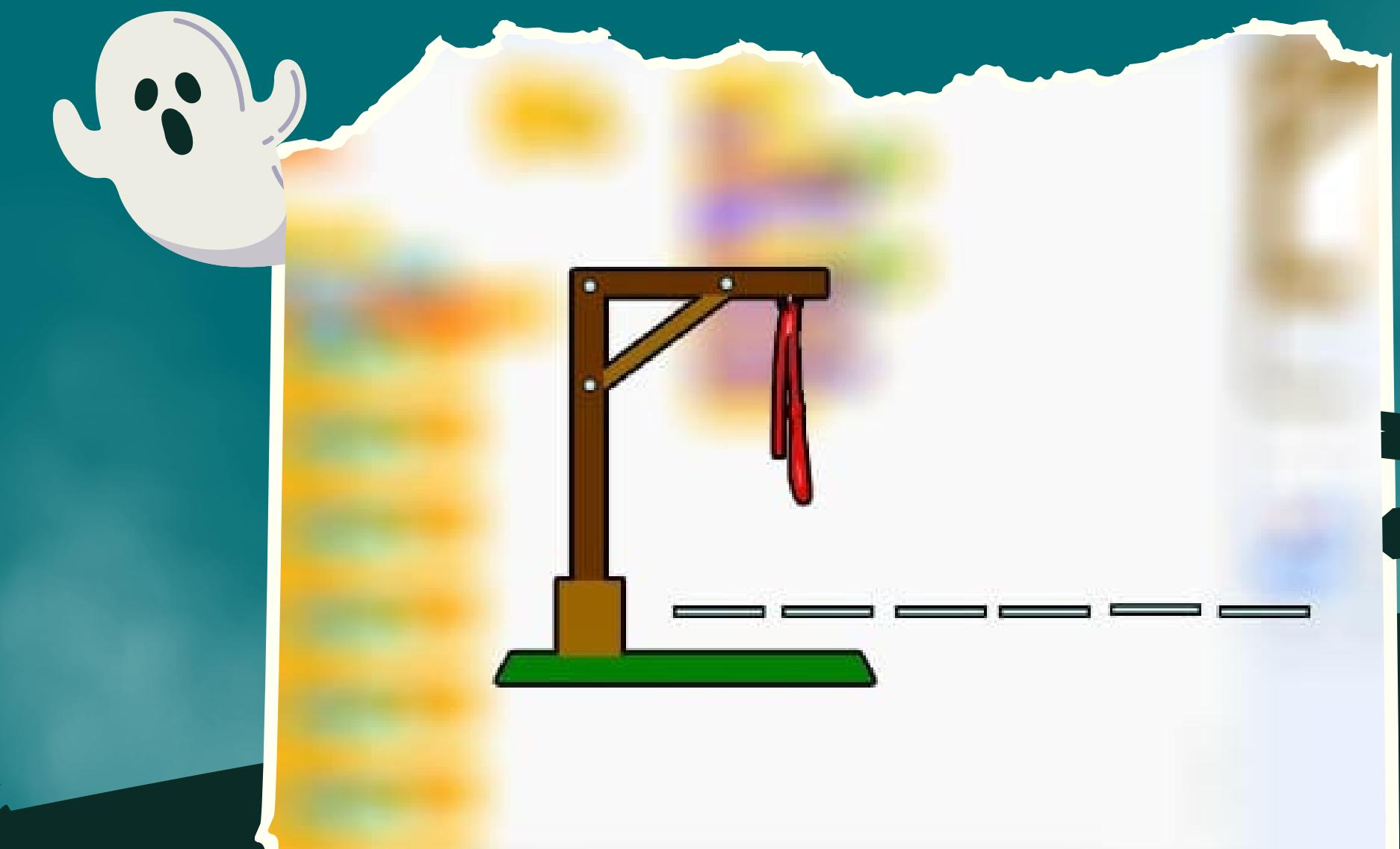
Configuración del entorno

Desarrollo del código

Interacción y coordinación
con el usuario

Pruebas y mantenimiento

Entrega



DEFINICIÓN

¡¡Bienvenidos al Juego del Ahorcado en Python!

Este clásico juego de adivinanza ha sido programado en Python. Adivina la palabra secreta antes de que se complete la figura del ahorcado: cada letra incorrecta te acerca al final. Pon a prueba tu vocabulario y descubre cómo la programación puede crear juegos interactivos y divertidos.

¡Comienza el reto y que gane el mejor!

DEMO

Código del Juego del Ahorcado

Herramientas

- Python
- Visual Studio Code
- Jupyter (opcional)

```
# Definimos el número máximo de intentos
max_intentos = 6
intentos_fallidos = 0
palabra = random.choice(palabras)
palabra_oculta = "_" * len(palabra) # Palabra oculta con guiones bajos
letras_adivinadas = [] # Lista para letras ya adivinadas
while intentos_fallidos < max_intentos:
    print(f'Palabra oculta: {palabra_oculta}')
    mostrar_ahorcado(intentos_fallidos)
    letra = input('Escribe una letra: ').lower()
    # Verificamos si la letra es válida antes de continuar
    if validar_letras(letra, letras_adivinadas):
        print(validar_letras(letra, letras_adivinadas))
    else:
        # Si no adivina, incrementamos los intentos fallidos
        print(f'{letra} no está en la palabra. Prueba de nuevo.')
        intentos_fallidos += 1
        print(f'Intentos fallidos: {intentos_fallidos}/{max_intentos}')

    # Si no es válida, volvemos a pedir la letra
    letras_adivinadas.append(letra)
    # Usamos la función adivinar_posiciones para encontrar la letra en la palabra
    posiciones = adivinar_posiciones(palabra, letra)
    if posiciones:
        print(f'¡La letra "{letra}" está en la palabra!')
        # Actualizamos la palabra oculta con las posiciones encontradas
        palabra_oculta_lista = list(palabra_oculta) # Convertimos a lista para modificar
        for posicion in posiciones:
            palabra_oculta_lista[posicion] = letra # Reemplazamos los guiones bajos
        palabra_oculta = ''.join(palabra_oculta_lista) # Convertimos de nuevo a cadena
        print(f'Palabra oculta actualizada: {palabra_oculta}')
    # Si ya no quedan guiones bajos, el jugador ha ganado
    if "_" not in palabra_oculta:
        print(f'; Felicidades! Has adivinado la palabra: {palabra}')
        break

# Condición si se acaban los intentos
if intentos_fallidos == max_intentos:
    print(f'; Intentos agotados.')
```





Agregar niveles de
dificultad



Elegir categorías



Mejorar interfaz

NEXT STEPS



CONCLUSIONES

Primeros pasos en programación

Fomentar el trabajo en equipo

Desarrollo de lógica del juego

Manejo de herramientas: IT

Oportunidad de aprendizaje:
feedback



```
Introduzca al juego la palabra que se oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 1/6  
Palabra oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 2/6  
Palabra oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 3/6  
Palabra oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 4/6  
Palabra oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 5/6  
Palabra oculta: _____  
  
Por favor introduce una sola letra válida  
* no está en la palabra. Prueba de nuevo.  
Intentos fallidos: 6/6  
Palabra oculta: _____
```



¡JUDAS?



**GRACIAS
POR VUESTRA ATENCIÓN**

