



EXPOSICIÓN: JUEGO DEL AHORCADO EN PYTHON

Roy Jimenez



INTRODUCCIÓN

Objetivo:

Crear un juego donde el jugador adivine una palabra oculta letra por letra, con un límite de intentos, utilizando Python como lenguaje base.

2. ESTRUCTURA DEL PROYECTO

El código se organiza en tres funciones principales y utiliza estructuras de datos clave:

1. `obtener_palabra_aleatoria()`
 - Lista: Usa una lista predefinida de palabras (["gato", "computadora", ...]).
 - Módulo random: Selecciona una palabra aleatoria con `random.choice()`.
2. `mostrar_tablero()`
 - Strings y ciclos: Construye dinámicamente un "tablero" que muestra letras adivinadas y oculta las demás con `[_]`.
 - Ejemplo:
 - Palabra: "gato" → Letras adivinadas: ["a", "o"] → Tablero: `[_]` a `[_]` o.
3. `jugar_ahorcado()`
 - Ciclo while: Controla los intentos del jugador (7 vidas).
 - Input de usuario: Captura letras ingresadas con `input()`.
 - Listas y conjuntos:
 - Almacena letras correctas en una lista (`letras_adivinadas`).
 - Usa `set()` para verificar si todas las letras fueron adivinadas.

3. CONCEPTOS DE PROGRAMACIÓN APLICADOS

- Estructuras de datos:
 - Listas: Para almacenar palabras y letras adivinadas.
 - Strings: Para construir el tablero visual.
 - Conjuntos (set): Para comparar letras eficientemente.
- Funciones: Modularizan el código (ej: una función para cada tarea específica).
- Ciclos y condicionales:
 - while: Gestiona el flujo del juego hasta agotar intentos.
 - if/else: Evalúa si una letra es correcta o no.



CONCLUSIÓN

Este proyecto es un ejemplo claro de cómo aplicar conceptos básicos de Python para crear un juego funcional. Además:

- Educativo: Refuerza lógica de programación.
- Escalable: Puede extenderse con más características.

Invitación al diálogo:

¿Preguntas sobre cómo funciona alguna parte?

¿Ideas para mejorarlo? ¡Estoy abierto(a) a discutirlo!



GRACIAS