**Contexto**

Seis estudiantes quedan encerrados en el aula de informática de la planta 4 por un espíritu vengativo: Dr. Byte, un profesor de programación que murió tras un error en un código crítico. Su alma quedó atrapada en los sistemas del instituto, y ahora reta a los estudiantes a resolver acertijos de Python para escapar.

**Planta 4: El Aula de Informática**

Objetivo: Salir del aula resolviendo acertijos de Python.

Acertijo 1: Los Cuadrados Rotos

* Dinámica: Encuentran 4 trozos de papel en la papelera con fragmentos de una cuadrícula numérica. Al unirlos, aparece una secuencia: 3, 1, 7, 9.
* Prueba Python:

python

*# Ordenar los números para obtener la contraseña*

numeros = [3, 1, 7, 9]

contraseña = ''.join(map(str, sorted(numeros))) *# Resultado: "1379"*

* Solución: Introducen 1379 en un ordenador bloqueado, que muestra un mensaje: "La puerta se abre con suma(5, 3)". Deben definir la función:

python

**def** suma(a, b):

**return** a + b

Al ejecutar suma(5, 3), obtienen 8 y abren la puerta.

**Planta 3: El Pasillo de las Variables Perdidas**

Ambiente: Pizarras con código incompleto y un diccionario vacío en una mesa.

Acertijo 2: El Diccionario Fantasma

* Dinámica: Encuentran un diccionario {"puerta": "PLANTA\_", "número": ?} y un 2 tachado en la pared.
* Prueba Python:

python

*# Concatenar para formar la contraseña*

**print**("PLANTA\_" + str(2)) *# Salida: "PLANTA\_2"*

* Solución: Usan PLANTA\_2 para activar el ascensor.

**Planta 2: El Laboratorio de Bucles**

Ambiente: Luces estroboscópicas y un ordenador con un bucle infinito.

Acertijo 3: El Bucle Sin Fin

* Código defectuoso:

python

**for** i **in** range(10):

**if** i % 2 == 0:

**print**(i)

*# ¡Falta algo!*

* Prueba Python: Agregar break tras print(i) para evitar repeticiones.
* Solución: Corregido, se imprime 0 y se abre una caja con un USB que activa el ascensor.

**Planta 1: El Cuarto de las Excepciones**

Ambiente: Libros de Python antiguos y un espejo que refleja código al revés.

Acertijo 4: La Función Recursiva

* Código con error:

python

**def** factorial(n):

**return** n \* factorial(n-1) *# Error: falta caso base*

* Prueba Python: Añadir if n == 0: return 1.
* Solución: Al calcular factorial(3), obtienen 6. El espejo revela 0420, la clave para la planta 0.

**Planta 0: El Santuario de Dr. Byte**

Ambiente: Un servidor central con un código encriptado y una puerta blindada.

Acertijo Final: El Enigma del Módulo

* Prueba Python:

python

**def** enigma(a, b):

**return** (a \*\* b) % 10 *# Ejemplo: enigma(3, 4) → 81 % 10 = 1*

* Solución: Resuelven enigma(3, 4) y obtienen 1. Al ingresarlo, liberan a Dr. Byte y escapan.

**Elementos de Misterio**

* Fantasmas: Sombras proyectan código falso en las paredes para confundir.
* Easter Eggs: Mensajes ocultos en comentarios de código (ej: # Busca en el armario).
* Tiempo: Un reloj cuenta regresivamente, con pausas al resolver acertijos correctamente.

Nota: Cada prueba refuerza conceptos de Python: funciones, bucles, diccionarios y operadores.