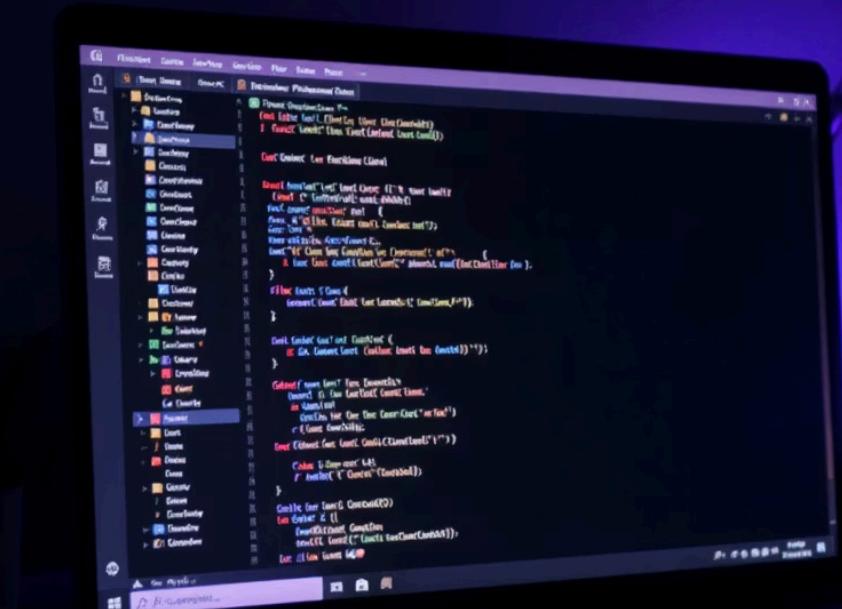


Cuaderno de Evidencias Interactivo: Desarrollando Habilidades en Python

Por Delucio Fuentes Ricardo Ismael

Esta presentación explora el Cuaderno de Evidencias, una plataforma educativa. Desarrollada en Python con Tkinter, ayuda a rastrear el progreso de programación. Consolidará tus conocimientos en Python 3.9+.



```
class Notebook:
    def __init__(self):
        self.pages = []

    def add_page(self, page):
        self.pages.append(page)

    def remove_page(self, page):
        if page in self.pages:
            self.pages.remove(page)
        else:
            print("Page not found")

    def get_page(self, index):
        if index < len(self.pages):
            return self.pages[index]
        else:
            print("Index out of range")

    def __str__(self):
        return f"Cuaderno de Evidencias con {len(self.pages)} páginas"

class Page:
    def __init__(self, content):
        self.content = content

    def __str__(self):
        return self.content

class NoteBook:
    def __init__(self):
        self.notebook = Notebook()

    def add_note(self, note):
        self.notebook.add_page(note)

    def remove_note(self, note):
        self.notebook.remove_page(note)

    def get_note(self, index):
        return self.notebook.get_page(index)

    def __str__(self):
        return str(self.notebook)
```

¿Qué es el Cuaderno de Evidencias?

Repositorio Digital

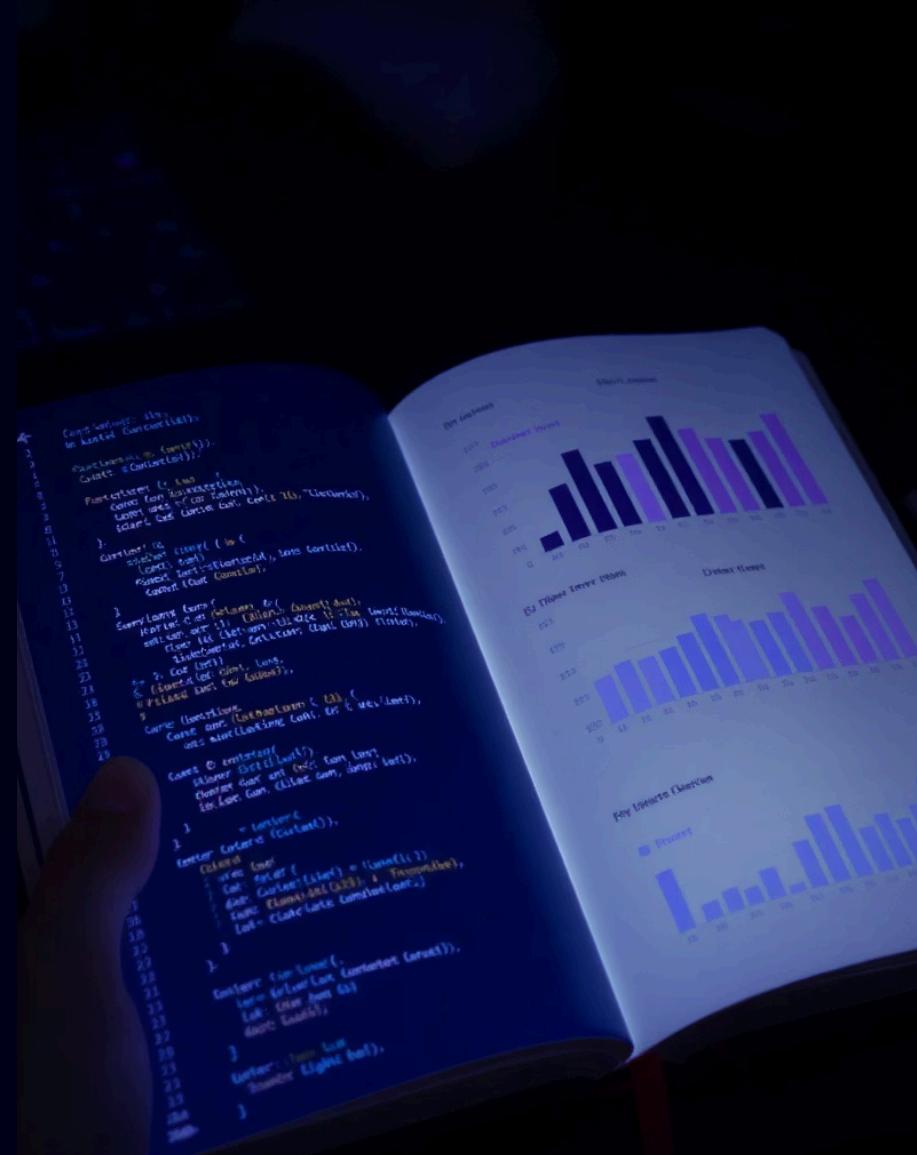
Almacena actividades y proyectos de programación.

Aprendizaje Activo

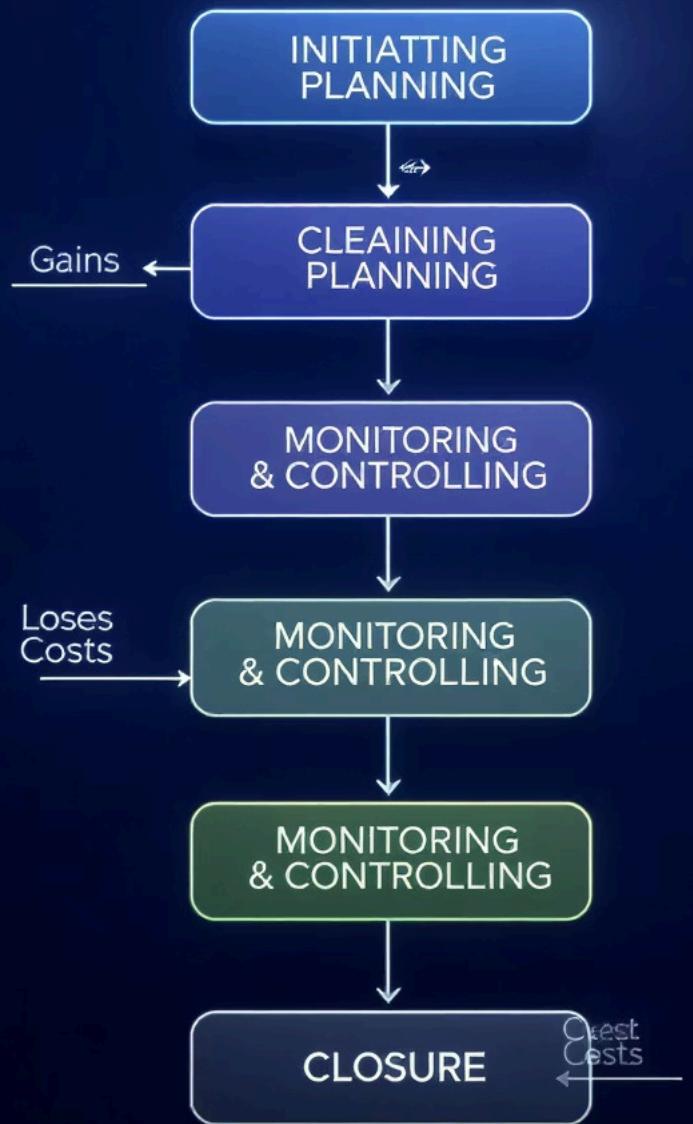
Fomenta la exploración y el estudio autodirigido.

IDE Integrado

Ofrece un entorno de ejecución para tus códigos.



PROJECT CHART



Estructura Cascada del Cuaderno

Fase 1: Requisitos

Comprender el problema y definir el alcance.

Fase 2: Diseño

Planificar la solución y su arquitectura.

Fase 3: Implementación

Codificar la solución en lenguaje Python.

Fase 4: Pruebas

Verificar la funcionalidad y corregir errores.

Los módulos se organizan en fases de desarrollo. La progresión aumenta en complejidad. Cada fase se valida antes de pasar a la siguiente.

Actividad Destacada: Simulación de un Cine



Módulo Avanzado

Parte del Cuaderno de Evidencias.



Alta Complejidad

Una de las 5 actividades más complejas.



Concurrencia

Enfocado en hilos y gestión de recursos.



Hilos

Entender interacciones de múltiples hilos.



El Desafío: Gestionando las Ventanillas de un Cine

Se simula la compra y reserva de boletos. El reto es evitar la sobreventa. Esto se logra gestionando accesos simultáneos a las butacas.



Simulación 3 Hilos

Prueba inicial de concurrencia.



Simulación 5 Hilos

Escenario más complejo con mayor interacción.



Simulación Secuencial

Línea base para comparar rendimiento.

Concurrencia en Acción: Detalles Técnicos

```
def llegada_clientes(self, total, tl):
    for i in range(1, total +1):
        time.sleep(tl)
        cliente = f"Cliente-{i}"
        self.cola_clientes.put(cliente)
        with self.lock:
            self.ventana.after(0, self.actualizar_info,f"Llego {cliente} (En cola: {self.cola_clientes.qsize()})")

def caja_con(self, numero):
    while True:
        try:
            cliente =self.cola_clientes.get(timeout=2)
        except queue.Empty:
            break

        if not self.sem.acquire(timeout=1):
            with self.lock:
                self.ventana.after(0, self.actualizar_info,f"[Caja-{numero}] No pudo atender a {cliente} (timeout)")
            continue

        try:
            with self.lock:
                self.ventana.after(0, self.actualizar_info,f"[Caja-{numero}] Atendiendo a {cliente}")
                time.sleep(random.uniform(0.5,1.0))

            with self.lock:
                self.atendidos+=1
                self.ventana.after(0, self.actualizar_info,f"[Caja-{numero}] Termino con {cliente} | Total atendidos: {self.atendidos}")
        finally:
            self.sem.release()
            self.cola_clientes.task_done()
```

</>

Threading Python

Simulación de usuarios con hilos.



Mutex (Lock)

Protege recursos compartidos, como butacas.



Tkinter Visual

Muestra el estado de cada butaca.



Manejo de Colas

Procesamiento ordenado de solicitudes.

Aprendizajes Clave de la Simulación

1 Race Conditions

Entender y prevenir estas condiciones.

2 Sincronización

Uso práctico de Lock y Semaphore.

3 Sistemas Robustos

Diseño para entornos multiusuario.

4 Rendimiento

Optimización en operaciones concurrentes.

5 Depuración

Identificación de errores en aplicaciones con hilos.

```
5 print  
6 -----  
7 print({ "Hello, World""
```

Conclusión

El Cuaderno de Evidencias es una herramienta didáctica integral. La simulación de cine demuestra la concurrencia. Es aplicable a sistemas distribuidos y web. Proporciona una base sólida para futuros desarrollos en Python.