



Ejercicio Final: El Desafío Jerárquico

Recursividad, OS y Persistencia Avanzada

Un proyecto que transformará tu forma de pensar sobre la organización de datos



Objetivo Central y Metodología

El Cambio de Paradigma

Evolucionamos del modelo simple de CSV único hacia la gestión jerárquica avanzada mediante carpetas. Los datos ahora se organizan por filtros y niveles, reflejando estructuras del mundo real.

Equipos: 2-3 personas trabajando en colaboración

Entregables Obligatorios

- Repositorio GitHub con código completo
- README detallado explicando diseño
- Video explicativo del funcionamiento
- Estructura de 3 niveles implementada

PILAR 1: Diseño de la Estructura



Investigación y Planificación

Tu equipo diseña la jerarquía de 3 niveles que se mapeará directamente al sistema de archivos



Diccionarios en Python

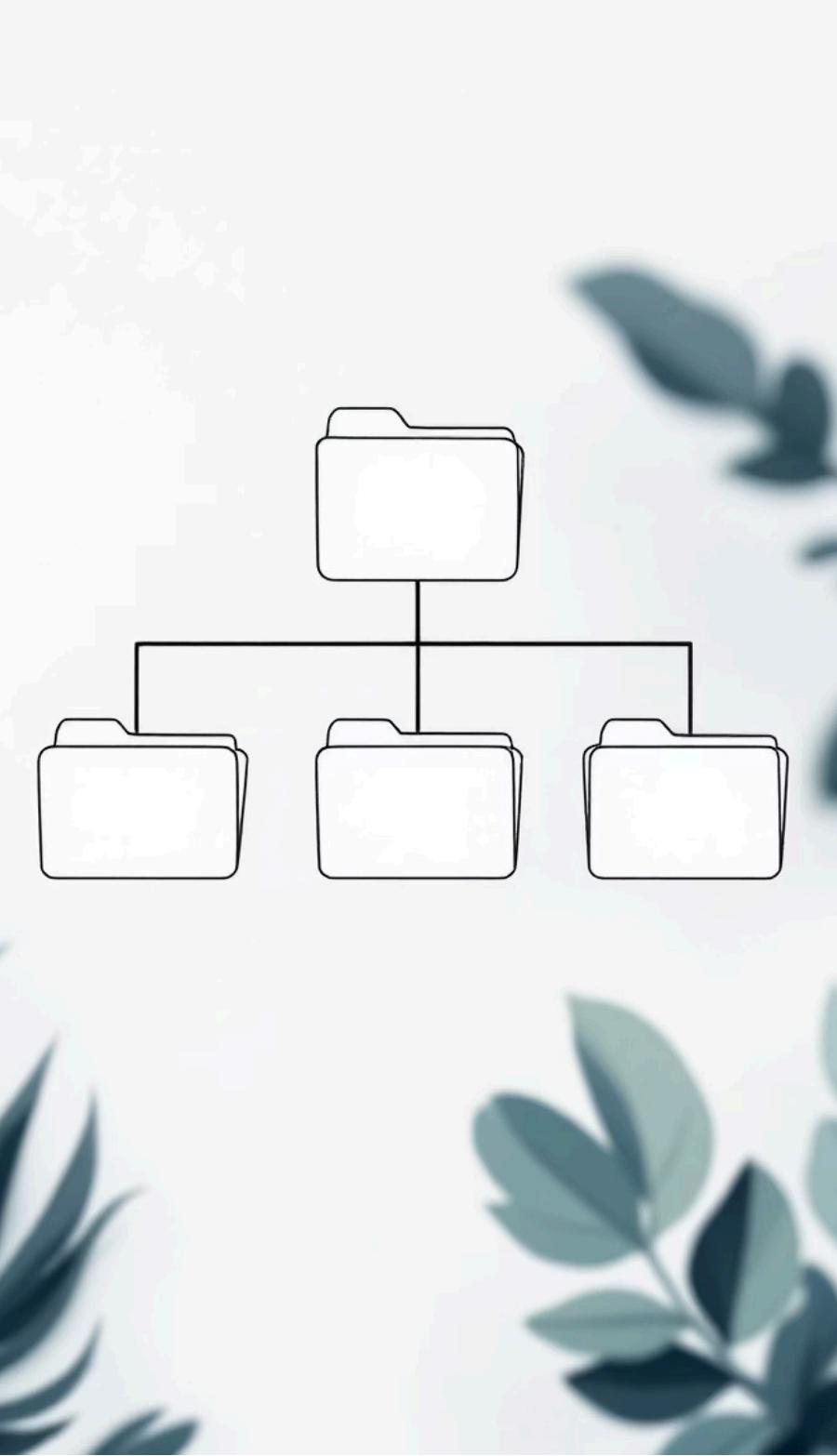
Cada ítem individual debe manejarse como un **diccionario** para máxima flexibilidad



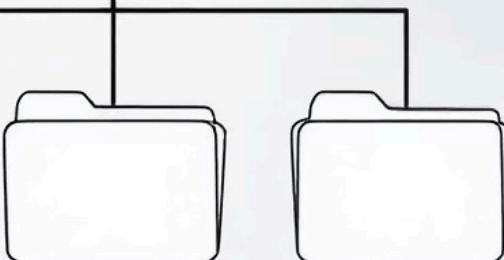
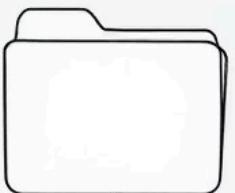
Documentación Clara

Explica tu diseño en el README y demuéstralos en el video con ejemplos concretos

- Ejemplo:** Continente → País → Ciudad/Datos.csv o Categoría → Subcategoría → Producto.csv



PILAR 2: Persistencia con Librería OS



Creación Dinámica de Carpetas

Usa la librería os para crear automáticamente la jerarquía de directorios que representan tus filtros

Manejo Seguro de Archivos

Estructura obligatoria: `with open(...)` as `archivo:` para garantizar el cierre automático y seguro

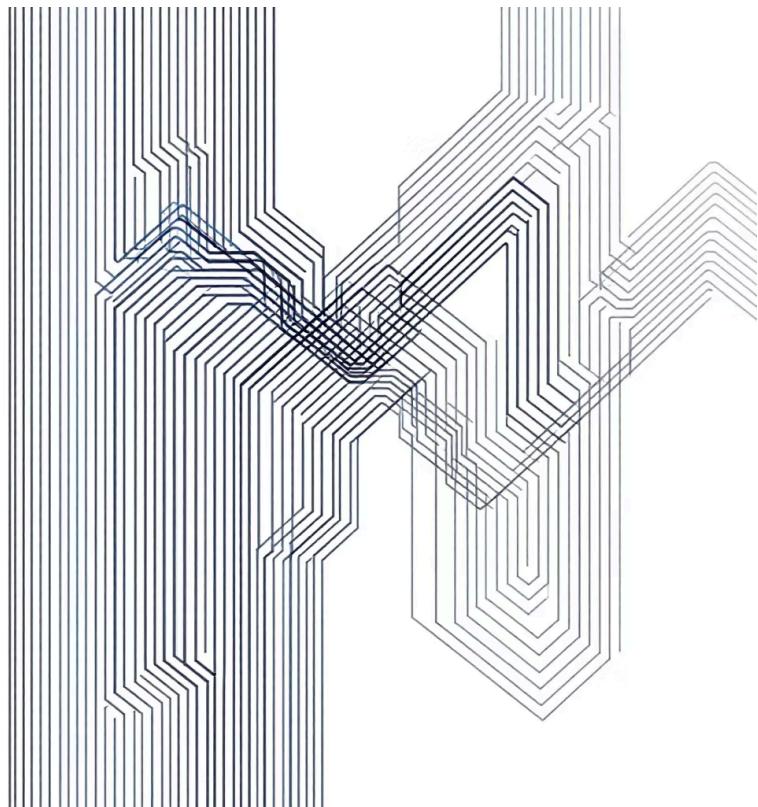
Modo de Escritura Append

Utiliza el modo '`a`' (append) al escribir para añadir contenido sin sobrescribir datos existentes

La función de agregar debe **verificar y crear** toda la jerarquía de directorios si no existe previamente

PILAR 3: Lectura Recursiva Global

Requisito Crítico Obligatorio



01

¿Qué es Recursividad?

Una función que se llama a sí misma para resolver subproblemas progresivamente más pequeños

02

Caso Base

Condición de corte que detiene la recursión (ej: encontrar el CSV o llegar al final del directorio)

03

Paso Recursivo

La función se llama a sí misma con un subproblema (ej: explorar el siguiente subdirectorio)

- ❑ **Propósito:** Consolidar todos los datos de todos los CSV dispersos en una única lista de diccionarios para procesamiento posterior

Fundamentos de Calidad

Prácticas Básicas Requeridas ✓

1 Modularización

Usa funciones dedicadas para cada operación: CRUD, estadísticas, validaciones

2 Estructuras de Datos

Combina listas y diccionarios para gestionar información de forma eficiente

3 Estilo PEP 8

Indentación consistente con **4 espacios** (nunca tabulaciones mezcladas)

4 Persistencia Correcta

Archivos CSV con lectura/escritura usando `with open()`

5 Funcionalidades Completas

Ordenamiento por 2+ criterios, estadísticas (promedio, suma, conteo por categoría)



ANTI-PATRONES

Errores NO Aceptados □

No Usar Recursividad

Usar bucles iterativos (for/while) para recorrer carpetas **no cumple el requisito central**. La recursión es obligatoria.

Manejo Inseguro de Archivos

Abrir archivos sin la cláusula `with` provoca fugas de recursos y errores potenciales

Excepciones Genéricas

Usar `except:` sin especificar tipo oculta errores críticos. Especifica: `except OSError:`

Recursividad Infinita

No definir o implementar mal el caso base resulta en `RecursionError` y bloqueo del programa

Inconsistencia de Estilo

Mezclar tabulaciones y espacios genera errores de indentación impredecibles

Diseño Débil

Elegir un dominio que no justifica 3 niveles de jerarquía demuestra falta de planificación

Resumen y Próximos Pasos

Los 3 Pilares

- Diseño jerárquico de 3 niveles
- Librería OS para carpetas dinámicas
- Recursividad obligatoria para lectura

Calidad del Código

- Modularización y estructuras correctas
- PEP 8 con 4 espacios
- Manejo seguro con `with`

Entregables

- GitHub: código + README
- Video explicativo completo
- Evita todos los anti-patrones

Este proyecto es tu oportunidad de demostrar dominio técnico real. La recursividad, el manejo de archivos y el diseño jerárquico son habilidades fundamentales que te distinguirán como programador.

¡Revisa la fecha límite y comienza a planificar con tu equipo hoy mismo!

