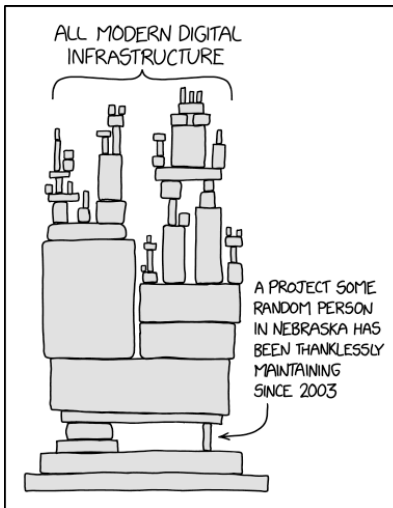


CP06-A mi si me función-a

Curso 2025-2026



1. Refactorización

Cambia cada ejercicio de las clases prácticas anteriores y modela la solución para solucionarlo a través de una función que recibe una entrada y retorna un valor en lugar de una variable que almacena el valor o entrada y salida de la consola.

2. Cuestión de diseño

Se tienen como datos en forma de lista que guarda los resultados académicos de distintos estudiantes. En cada posición se guarda una tupla de una cadena de texto y una lista de enteros, la cadena corresponde al nombre del estudiante y la lista corresponde a las notas en cada una de sus evaluaciones parciales.

De estos datos se desea obtener:

- El promedio de asistencia a las evaluaciones (porcentaje de evaluaciones asistidas) sabiendo que se hicieron n . En caso de no tener este n se asume que fueron 12 evaluaciones.
- Sabiendo que la inasistencia a una evaluación se considera como suspenso (nota 2), una lista con los nombres de los estudiantes aprobados al considerar el promedio de notas en las evaluaciones.
- El nombre del estudiante con mejor nota, para poder entregarle un gladiolo.

Para obtener esta información debe implementar una función de nombre `process_data` que reciba como parámetro la lista con todas las entradas y retorne una tupla de tamaño 3 con todo lo que se pide.

En este ejercicio se desea que enfrenten el problema de forma modular, intentando separar fragmentos de la solución en funciones más pequeñas. Para esto puede ser buena idea seguir estas dos metodologías:

Descomposición

Tu proceso de pensamiento debe ser:

1. **Empezar por el objetivo final:** ¿Cuál es el resultado completo que necesito?
2. **Dividir en sub-problemas:** ¿Cuáles son los 3 o 4 grandes pasos lógicos y específicos para llegar a ese resultado?
3. **Crear funciones especialistas:** Escribe una función para cada uno de esos pasos. Cada función resolverá una parte concreta y a medida del problema principal.

Composición

Tu proceso de pensamiento debe ser:

1. **Empezar por las operaciones básicas:** Ignora el objetivo final por un momento. ¿Cuáles son las acciones más pequeñas, fundamentales y repetibles que necesitaré (como sumar una lista, filtrar datos, aplicar una fórmula)?
2. **Crear herramientas genéricas:** Escribe una función para cada una de estas acciones básicas. Estas funciones deben ser universales y no saber nada del problema específico.
3. **Construir la solución:** Utiliza tus herramientas, combinándolas como si fueran bloques, para construir la solución al problema completo.

Nota: Intenta hacer distintas soluciones para el mismo problema.

3. Ordenando con criterio I

Define una función `sort_with_key` que reciba una lista de tuplas de la forma (clave, valor) y devuelva una nueva lista con las claves ordenadas de forma ascendente según su valor correspondiente.

Ejemplo: Si la entrada es [(1, 30), (2, 20), (3, 40), (4, 10), (5, 50)], la función debe devolver [4, 2, 1, 3, 5].

4. Mantenimiento

Reduce este código manteniendo relativamente fácil añadirle funcionalidad adicional (como producto y clientes)

```
# --- Sección 1: Registro del Producto Alfa ---
producto_1_nombre = "Manzana Cósmica"
producto_1_precio = 5.50
producto_1_cantidad = 100
producto_1_en_stock = True

print("--- Ficha de Producto ---")
print("Nombre:", producto_1_nombre)
print("Precio Unitario:", producto_1_precio)
print("Cantidad en Inventario:", producto_1_cantidad)
print("¿Está en stock?", producto_1_en_stock)
valor_total_producto_1 = producto_1_precio * producto_1_cantidad
print("Valor total del inventario de este producto:", valor_total_producto_1)
print("-----")
```

```

print("\n") # Un poco de espacio para que no se amontone

# --- Sección 2: Registro del Producto Beta ---
producto_2_nombre = "Plátano Galáctico"
producto_2_precio = 3.25
producto_2_cantidad = 250
producto_2_en_stock = True

print("---- Ficha de Producto ----")
print("Nombre:", producto_2_nombre)
print("Precio Unitario:", producto_2_precio)
print("Cantidad en Inventario:", producto_2_cantidad)
print("¿Está en stock?", producto_2_en_stock)
valor_total_producto_2 = producto_2_precio * producto_2_cantidad
print("Valor total del inventario de este producto:", valor_total_producto_2)
print("-----")
print("\n") # Un poco de espacio para que no se amontone

# --- Sección 3: Registro del Producto Gamma ---
producto_3_nombre = "Naranja Nebulosa"
producto_3_precio = 7.80
producto_3_cantidad = 0
producto_3_en_stock = False

print("---- Ficha de Producto ----")
print("Nombre:", producto_3_nombre)
print("Precio Unitario:", producto_3_precio)
print("Cantidad en Inventario:", producto_3_cantidad)
print("¿Está en stock?", producto_3_en_stock)
valor_total_producto_3 = producto_3_precio * producto_3_cantidad
print("Valor total del inventario de este producto:", valor_total_producto_3)
print("-----")
print("\n") # Un poco de espacio para que no se amontone

# --- Sección 4: Procesamiento del Cliente 1 ---
cliente_1_nombre = "Alex"
cliente_1_apellido = "Reyes"
cliente_1_ciudad = "Villa Código"

print("---- Perfil de Cliente ----")
print("Nombre Completo:", cliente_1_nombre, cliente_1_apellido)
print("Edad: 28 años")
print("Ciudad de residencia:", cliente_1_ciudad)
print("Estatus: Cliente VIP. ¡Ofrecer descuentos!")
# Mensaje de bienvenida
print("Bienvenido,", cliente_1_nombre, ". Gracias por registrarte.")
print("-----")
print("\n")

```

```

# --- Sección 5: Procesamiento del Cliente 2 ---
cliente_2_nombre = "Beatriz"
cliente_2_apellido = "Sosa"
cliente_2_ciudad = "Pueblo Python"

print("--- Perfil de Cliente ---")
print("Nombre Completo:", cliente_2_nombre, cliente_2_apellido)
print("Edad: 20 años")
print("Ciudad de residencia:", cliente_2_ciudad)
print("Estatus: Cliente regular.")
# Mensaje de bienvenida
print("Bienvenido,", cliente_2_nombre, ". Gracias por registrarte.")
print("-----")
print("\n")

# --- Sección 6: Procesamiento del Cliente 3 ---
cliente_3_nombre = "Carla"
cliente_3_apellido = "Gómez"
cliente_3_ciudad = "Villa Código"

print("--- Perfil de Cliente ---")
print("Nombre Completo:", cliente_3_nombre, cliente_3_apellido)
print("Edad: 35 años")
print("Ciudad de residencia:", cliente_3_ciudad)
print("Estatus: Cliente VIP. ¡Ofrecer descuentos!")
# Mensaje de bienvenida
print("Bienvenido,", cliente_3_nombre, ". Gracias por registrarte.")
print("-----")
print("\n")

# --- Sección 7: Un final redundante ---
print("Procesamiento de datos inicial completado.")
print("El sistema está listo para más entradas manuales.")
print("Por favor, copie y pegue el siguiente bloque para un nuevo producto.")
print("O copie y pegue el bloque anterior para un nuevo cliente.")
print("La eficiencia es nuestra pasión (nótese el sarcasmo).")
print("Procesamiento de datos inicial completado.")
print("El sistema está listo para más entradas manuales.")
print("Por favor, copie y pegue el siguiente bloque para un nuevo producto.")
print("O copie y pegue el bloque anterior para un nuevo cliente.")
print("La eficiencia es nuestra pasión (nótese el sarcasmo).")
print("Procesamiento de datos inicial completado.")
print("El sistema está listo para más entradas manuales.")

```

5. Rueda matricial

Implemente una función que dada una matriz y un entero k rote todos los elementos de la matriz k veces hacia la derecha y devolviendo una nueva matriz con el resultado.

Ejemplo: $k=1$. Matriz original:

```
[  
  [1, 2, 3, 4],  
  [5, 6, 7, 8],  
  [9, 10, 11, 12],  
  [13, 14, 15, 16]  
]
```

Matriz rotada:

```
[  
  [13, 9, 5, 1],  
  [14, 10, 6, 2],  
  [15, 11, 7, 3],  
  [16, 12, 8, 4]  
]
```

6. Multiplicación de matrices

Defina una función que dadas dos listas del mismo tamaño calcule el producto escalar de estas, es decir, entre $[a_0, a_1, \dots, a_{n-1}, a_n]$ y $[b_0, b_1, \dots, b_{n-1}, b_n]$, calcular el valor $a_0 \cdot b_0 + a_1 \cdot b_1 + \dots + a_n \cdot b_n$

Defina otra función que calcule la multiplicación de dos matrices en donde utilice la función anterior. ¿Qué otras funciones auxiliares pueden ayudar a su implementación?