

»»» Actividad de Aprendizaje No. 4 – Resolución de problemas | Programación con Python (Guía de Ejercicios Prácticos)

<i>Tiempo de inversión</i>	: 4 horas (casa)
<i>Periodo de ejecución</i>	: 31 de marzo al 4 de abril
<i>Modalidad</i>	: trabajo en grupo
<i>Puntaje</i>	: actividad sumativa

Descripción de la actividad:

Los estudiantes resuelven una lista de ejercicios (**Guía de Ejercicios Prácticos**) propuesta por el docente donde podrán poner en práctica los conceptos y herramientas del lenguaje de programación Python.

Recursos de trabajo:

- Plataforma virtual
- Visual Studio Code – Python
- Git – GitHub

Guía de Ejercicios Prácticos

Funciones Internas, Módulos y Paquetes

Ejercicio #1: Análisis de datos de ventas

Desarrollar un programa que procese un conjunto de registros de ventas (por ejemplo, listas de diccionarios) para extraer información relevante. Los estudiantes deberán aplicar funciones internas como *map*, *filter* y *reduce* para transformar y filtrar los datos, calculando totales y promedios. Este ejercicio busca que los estudiantes identifiquen y aprovechen las funcionalidades nativas de Python para la manipulación eficiente de estructuras de datos.

Ejercicio #2: Módulo de ordenamiento

Se requiere que los estudiantes diseñen un módulo independiente que contenga implementaciones de algoritmos de ordenamiento simples (bubble sort). El objetivo es que, a partir de una función principal, se invoquen los métodos del módulo para ordenar una lista de números y demostrar la correcta separación de responsabilidades, fomentando la modularidad y la reutilización del código.

Ejercicio #3: Paquete de algoritmos de búsqueda

Desarrollar un programa para organizar diferentes algoritmos de búsqueda en un paquete estructurado. Los estudiantes deberán crear al menos dos módulos que contengan implementaciones de búsqueda lineal y búsqueda binaria, configurando adecuadamente el archivo **init.py**. Posteriormente, desde un

script principal, se utilizarán estas funciones para localizar elementos específicos en una colección de datos, resaltando la importancia de la organización en proyectos de mayor envergadura.

Clases y Objetos

Ejercicio #4: Gestión de productos e inventario

Diseñar una clase *Producto* que incluya atributos como código, nombre, precio y cantidad en stock. Además, los estudiantes deberán implementar una clase *Inventario* que administre una colección de objetos *Producto*, incorporando métodos para agregar, buscar, actualizar y eliminar productos. Este ejercicio permite modelar situaciones reales de gestión de ventas y refuerza el concepto de encapsulación y manejo de colecciones en programación orientada a objetos.

Ejercicio #5: Procesamiento de pedidos y clientes

Crear una clase *Cliente* con atributos básicos (por ejemplo, ID, nombre y contacto) y una clase *Pedido* que contenga información sobre el cliente, la lista de productos solicitados y el total de la venta. Se podrá incluir el uso de herencia para diferenciar entre tipos de clientes (por ejemplo, cliente regular y cliente VIP) y aplicar descuentos especiales, demostrando el uso de la herencia y el polimorfismo para adaptar el comportamiento de los objetos según el tipo de cliente.

Ejercicio #6 Facturación y reportes de ventas

Crear una clase *Factura* que simule el proceso de facturación de una venta. Los estudiantes deberán encapsular los datos internos de la factura (como los detalles de los productos, cantidades, precios y descuentos) y proveer métodos para calcular el total de la venta, generar reportes simples y validar la integridad de la información. Este ejercicio enfatiza la importancia de ocultar la implementación interna y de diseñar interfaces claras y seguras para la gestión de transacciones comerciales.

Lista de Cotejo		
Funciones Internas, Módulos y Paquetes		
Criterio	Descripción	Puntaje
Aplicación correcta de funciones internas (map, filter, reduce)	Se evaluará la capacidad para transformar y filtrar datos de forma eficiente en el contexto de ventas.	3
Diseño y uso adecuado de módulos	El código debe estar dividido en archivos lógicos, integrados correctamente en el programa principal para el procesamiento de datos de ventas.	3

Organización y estructuración en paquetes	El paquete debe estar correctamente configurado (incluyendo init.py) y demostrar una organización coherente y escalable en la implementación de algoritmos de búsqueda y ordenamiento en ventas.	4
Clases y objetos		
Definición y manipulación de clases y objetos en el contexto de ventas	Se evaluará el diseño correcto de clases (Producto, Inventario, Cliente, Pedido y Factura) y la implementación de métodos que permitan gestionar la información de ventas.	3
Implementación de herencia y polimorfismo (en el caso de clientes y pedidos)	El ejercicio debe mostrar una jerarquía clara y el uso adecuado de la herencia para diferenciar entre tipos de clientes o pedidos, adaptando el comportamiento de los objetos según su categoría.	3
Encapsulación y uso de métodos específicos en la facturación	Se verificará que los atributos internos estén debidamente protegidos y que los métodos permitan la interacción segura y efectiva con la clase Factura, garantizando la integridad de los datos y la generación correcta de reportes.	4
Total:		20