



Asignatura: Algoritmos y Estructuras de Datos

Guía Didáctica No. 3

Semana del 31 de marzo al 4 de abril de 2025

Colectivo de Asignatura

Managua, 31 de marzo 2025

I. CONTENIDOS

Unidad I: Introducción al Lenguaje de Programación Python

- Funciones Internas, Módulos y Paquetes
- Clases y Objetos

II. LOGROS DE APRENDIZAJES

Al finalizar la semana, los estudiantes serán capaces de:

1. Utilizar funciones internas de Python, así como importar y gestionar módulos y paquetes, aplicándolos en la estructuración y reutilización eficiente del código para la resolución de problemas computacionales.
2. Diseñar e implementar clases y objetos en Python, aplicando los principios de la programación orientada a objetos para modelar y resolver problemas del mundo real de manera estructurada y reutilizable.

III. MATERIAL DE ESTUDIO Y RECURSOS

- Material de lectura
 - <https://docs.python.org/es/3.13/tutorial/index.html>
 - Cuevas, A. (2016). Capítulo 4: Programación Funcional. En Python 3: Curso práctico (1^{ra}. Ed., pp. 145 – 179). RA-MA Editorial
 - Cuevas, A. (2016). Capítulo 5: Programación Orientada a Objetos. En Python 3: Curso práctico (1^{ra}. Ed., pp. 183 – 240). RA-MA Editorial
- Recursos
 - Visual Studio Code - Python
 - Git - GitHub
 - UAM Virtual

ORIENTACIONES METODOLÓGICAS

Para alcanzar los logros de aprendizajes planteados, trabajaremos con las siguientes estrategias y recursos:

»»» **Actividad de Aprendizaje No. 1** - Presentación electrónica en PowerPoint: Funciones y Módulos | Clases y Objetos

<i>Tiempo de inversión</i>	: 1 hora
<i>Periodo de ejecución</i>	: 31 de marzo al 4 de abril
<i>Modalidad</i>	: grupal
<i>Puntaje</i>	: actividad formativa

Descripción de la actividad:

El docente presenta los conceptos esenciales de la programación modular. Los estudiantes aplicarán funciones internas, módulos y paquetes para desarrollar programas de forma modular y escalable, promoviendo una codificación organizada y reutilizable. Además, se introducirá la programación orientada a objetos mediante la creación y manipulación de clases y objetos, lo que les permitirá modelar problemas de manera intuitiva y eficiente.

»»» **Actividad de Aprendizaje No. 2** - Práctica de laboratorio | Funciones y Módulos | Clases y Objetos

<i>Tiempo de inversión</i>	: 3 horas
<i>Periodo de ejecución</i>	: 31 de marzo al 4 de abril
<i>Modalidad</i>	: parejas
<i>Puntaje</i>	: actividad formativa

Descripción de la actividad:

Los estudiantes mediante el método hands-on learning o aprendizaje práctico, son guiados por el docente para replicar en la computadora ejercicios previamente seleccionados. Comenzaremos explorando el uso de funciones internas, módulos y paquetes para desarrollar soluciones de manera organizada y modular, facilitando la reutilización del código. Posteriormente, se introducirá la programación orientada a objetos a través de la creación y manipulación de clases y objetos, permitiendo modelar problemas de la vida real de forma intuitiva.

Recursos de trabajo:

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

»»» Actividad de Aprendizaje No. 3 – Lectura obligatoria

<i>Tiempo de inversión</i>	: 2 horas (casa)
<i>Periodo de ejecución</i>	: 31 de marzo al 4 de abril
<i>Modalidad</i>	: individual
<i>Puntaje</i>	: actividad formativa

Descripción de la actividad:

Los estudiantes realizan una lectura del material proporcionado por el docente, replicando los ejercicios resueltos, en el caso que la lectura los presente.

- Cuevas, A. (2016). Capítulo 4: Programación Funcional. En Python 3: Curso práctico (1^{ra}. Ed., pp. 145 – 179). RA-MA Editorial
- Cuevas, A. (2016). Capítulo 5: Programación Orientada a Objetos. En Python 3: Curso práctico (1^{ra}. Ed., pp. 183 – 240). RA-MA Editorial

»»» 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

»»» Actividad de Aprendizaje No. 5 – Evaluación práctica | Instalación y configuración del entorno de ejecución y desarrollo de Python

<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 formativa

Descripción de la actividad:

Los estudiantes realizan la instalación y configuración del entorno de desarrollo de Python en sus computadoras personales.

- ✓ Todos los estudiantes participan, entregando un documento con la explicación del proceso realizado.

Recursos de trabajo:

- Plataforma virtual
- Visual Studio Code – Python

Guía de Ejercicios Prácticos

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		
Criterio	Descripción	Puntaje
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.	
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.	
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.	