



## PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

### GUÍA DE APRENDIZAJE (Programación Estructurada Algoritmos (Python))

#### 1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Tecnología en Análisis y Desarrollo de Software
- **Código del Programa de Formación:** 228118
- **Nombre del Proyecto Formativo** (si aplica): Desarrollo de Software Empresarial Innovador Alineado con las Políticas y Estrategias Gubernamentales Locales y Nacionales.
- **Fase del Proyecto** (si aplica): Analizar
- **Actividad de Proyecto Formativo** (si aplica): Estructurar Procesos Logicos para la Construcción de Algoritmos y Bases de Datos
- **Competencia:** Evaluar requisitos de la solución de software de acuerdo con metodologías de análisis y estándares
- **Resultados de Aprendizaje:** Desarrollar procesos lógicos a través de la implementación de algoritmos.
- **Duración de la Guía de Aprendizaje** (horas): 120 horas

#### 2. PRESENTACIÓN

¡Bienvenidos a esta emocionante experiencia de aprendizaje con Programación Estructurada y Algoritmos en Python! A través de esta guía de aprendizaje, se explorarán los principios fundamentales de la programación estructurada, permitiendo a los aprendices desarrollar habilidades esenciales para la resolución de problemas mediante el diseño y la implementación de algoritmos eficientes en Python.

El estudio de la programación estructurada proporciona una base sólida para la comprensión de la lógica computacional, promoviendo el desarrollo del pensamiento algorítmico y la capacidad de analizar, diseñar y optimizar soluciones informáticas. A lo largo de esta guía, se fortalecerá la capacidad de abstracción, organización y estructuración del código, competencias clave en el desarrollo de software.

La metodología de aprendizaje fomenta tanto el trabajo autónomo como el colaborativo, incentivando a los aprendices a gestionar sus tiempos y recursos de manera eficaz. A través de actividades prácticas y desafiantes, se incentivará la aplicación de conocimientos previos para



consolidar el aprendizaje, estableciendo conexiones con nuevos conceptos y promoviendo un enfoque de mejora continua.

Además, la programación estructurada no solo permite adquirir destrezas técnicas, sino que también fortalece el pensamiento crítico y la capacidad de toma de decisiones en la solución de problemas computacionales. La colaboración en equipo será un pilar fundamental en este proceso, propiciando el intercambio de ideas y la construcción colectiva del conocimiento.

Esta guía de aprendizaje está diseñada para guiar paso a paso en la exploración de estructuras de control, tipos de datos, manipulación de archivos y el desarrollo de algoritmos que permitirán resolver problemas de manera eficiente. Se recomienda una participación activa en las actividades propuestas para maximizar la comprensión y aplicabilidad de los conceptos adquiridos.

¡Es momento de comenzar esta aventura con el desarrollo de las actividades!

### 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es):** La guía de aprendizaje sobre Programación Estructurada - Algoritmos con Python, consta de cuatro actividades clave. Primero, se realizará una reflexión inicial sobre los conocimientos previos de los aprendices. Luego, se contextualizarán los conceptos necesarios para el aprendizaje. A través de diversas actividades prácticas, los aprendices se apropiarán del conocimiento y se familiarizarán con la programación estructurada - algoritmos con Python, aplicarán lo aprendido en un proyecto final, integrando conceptos fundamentales para el desarrollo de software y de esta manera consolidar su aprendizaje.

#### 3.1 Actividades de reflexión inicial: "La Aventura de los Algoritmos"

**Descripción de la actividad:** En esta actividad, los aprendices explorarán la importancia del pensamiento algorítmico en la programación estructurada. A través de CodeCombat, se enfrentarán a desafíos que requieren el uso de instrucciones secuenciales, condicionales y bucles para avanzar en el juego. La experiencia será guiada por una situación problemática que los motivará a analizar la importancia de los algoritmos en la resolución de problemas reales.

##### Desarrollo de la Actividad

##### 1. Situación Problemática y Preguntas de Reflexión Inicial

*Se presentará la siguiente situación:*



Imagina que formas parte de un equipo de desarrollo de software y necesitas crear un programa que ayude a un robot a navegar en un laberinto lleno de obstáculos. Sin un conjunto de instrucciones claras, el robot no podrá avanzar. ¿Cómo podrías organizar estas instrucciones de manera eficiente?

*Preguntas para discusión en grupos:*

- ¿Por qué es importante organizar las instrucciones en un orden lógico?
- ¿En qué situaciones cotidianas utilizamos algoritmos sin darnos cuenta?
- ¿Cómo un algoritmo bien diseñado puede optimizar procesos en la vida real?

## **2. Exploración en CodeCombat**

Los aprendices ingresarán a CodeCombat y jugarán el primer nivel de la campaña "Kithgard Dungeon", donde deberán escribir código en Python para mover a su héroe utilizando estructuras de programación básicas.

Durante la actividad, se motivará a los aprendices a identificar cómo las instrucciones en el juego se asemejan a la creación de algoritmos en la vida real.

## **3. Discusión y Reflexión Final**

Se analizará la experiencia en CodeCombat, identificando los retos enfrentados y las estrategias utilizadas.

*Preguntas de cierre:*

- ¿Cómo influyó el orden de las instrucciones en el éxito del desafío?
- ¿Qué similitudes encuentras entre resolver problemas en el juego y la programación en Python?
- ¿Cómo podrías aplicar lo aprendido en CodeCombat para desarrollar un programa en la vida real?

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje Basado en Problemas (ABP): Se utilizará una situación problemática que motive la reflexión y el análisis crítico.
- Discusión en pequeños grupos: Los aprendices trabajarán en equipos para intercambiar ideas y experiencias previas.

**Materiales de formación:** Papel periódico, marcadores, colores, lápices y/o computador con acceso a internet, pantalla y/o televisor para proyectar, LMS (zajuna).



**Material de apoyo:**

- CodeCombat - Kithgard Dungeon: <https://codecombat.com>
- Python.org - Fundamentos de Programación en Python: <https://www.python.org/doc/>

**Duración de la actividad:** 02 horas.

### **3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje: Cuestionario de conocimientos previos sobre Fundamentos de Programación.**

**Descripción de la actividad:** Cada aprendiz completarán el cuestionario de selección múltiple con única respuesta para evaluar sus conocimientos previos sobre Fundamentos de Programación. Las preguntas están diseñadas para identificar su nivel de comprensión en cuanto a los conceptos esenciales de Programación Estructurada. El cuestionario servirá como herramienta de diagnóstico para que los aprendices reflexionen sobre sus saberes actuales y reconozcan las áreas en las que necesitan profundizar. Al final, se revisarán las respuestas correctas y se brindará una retroalimentación colectiva.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (zajuna).

**Estrategias o técnicas didácticas activas:**

- Cuestionario de selección múltiple con retroalimentación inmediata
- Evaluación diagnóstica individual

**Materiales de formación:** computador con acceso a internet, pantalla y/o televisor para proyectar, LMS (zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

**Duración de la actividad:** 02 horas.



### 3.3 Actividades de apropiación: Fundamentos de Programación Estructurada

**Descripción de la actividad No.01 – Estructuras Secuenciales y Condicionales:** En esta actividad, cada aprendiz desarrollará una solución para abordar la problemática planteada en la situación problemática, aplicando los conocimientos esenciales de programación estructurada. Se espera que los aprendices analicen los requerimientos de la situación, identifiquen las estructuras lógicas y algoritmos necesarios para resolver el problema de manera eficiente.

Una vez completada la solución, los aprendices la compartirán en grupo para analizar diferentes enfoques, identificar oportunidades de mejora y discutir las mejores prácticas de codificación y optimización de algoritmos. Este ejercicio fomentará el desarrollo del pensamiento lógico y la aplicación efectiva de los conceptos fundamentales de programación.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (Zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje basado en proyectos
- Evaluación entre pares

**Materiales de formación:** Equipo de cómputo, un navegador web, acceso a internet, pantalla y/o televisor para proyectar, LMS (Zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

**Evidencias de aprendizaje:**

- Taller de Estructuras Secuenciales y Condicionales en Python.

**Instrumentos de evaluación:**

- Rúbrica de evaluación de Estructuras Secuenciales y Condicionales en Python.

**Duración de la actividad:** 25 horas.



**Descripción de la actividad No.02 – Estructuras Cíclicas y Manejo de Excepciones:** En esta actividad, cada aprendiz desarrollará una solución para abordar la problemática planteada en la situación problémica, aplicando los conocimientos esenciales de programación estructurada. Se espera que los aprendices analicen los requerimientos de la situación, identifiquen las estructuras lógicas y algoritmos necesarios para resolver el problema de manera eficiente.

Una vez completada la solución, los aprendices la compartirán en grupo para analizar diferentes enfoques, identificar oportunidades de mejora y discutir las mejores prácticas de codificación y optimización de algoritmos. Este ejercicio fomentará el desarrollo del pensamiento lógico y la aplicación efectiva de los conceptos fundamentales de programación.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (Zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje basado en proyectos
- Evaluación entre pares

**Materiales de formación:** Equipo de cómputo, un navegador web, acceso a internet, pantalla y/o televisor para proyectar, LMS (Zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

**Evidencias de aprendizaje:**

- Taller de Estructuras Cíclicas y Manejo de Excepciones en Python.

**Instrumentos de evaluación:**

- Rúbrica de evaluación de Estructuras Cíclicas y Manejo de Excepciones en Python.

**Duración de la actividad:** 25 horas.



**Descripción de la actividad No.03 – Estructuras Almacenamiento:** En esta actividad, cada aprendiz desarrollará una solución para abordar la problemática planteada en la situación problémica, aplicando los conocimientos esenciales de programación estructurada. Se espera que los aprendices analicen los requerimientos de la situación, identifiquen las estructuras lógicas y algoritmos necesarios para resolver el problema de manera eficiente.

Una vez completada la solución, los aprendices la compartirán en grupo para analizar diferentes enfoques, identificar oportunidades de mejora y discutir las mejores prácticas de codificación y optimización de algoritmos. Este ejercicio fomentará el desarrollo del pensamiento lógico y la aplicación efectiva de los conceptos fundamentales de programación.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje basado en proyectos
- Evaluación entre pares

**Materiales de formación:** Equipo de cómputo, un navegador web, acceso a internet, pantalla y/o televisor para proyectar, LMS (zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

**Evidencias de aprendizaje:**

- Taller de Estructuras Almacenamiento en Python.

**Instrumentos de evaluación:**

- Rúbrica de evaluación de Estructuras Almacenamiento en Python.

**Duración de la actividad:** 25 horas.



**Descripción de la actividad No.04 – Funciones, Procedimientos y Archivos:** En esta actividad, cada aprendiz desarrollará una solución para abordar la problemática planteada en la situación problémica, aplicando los conocimientos esenciales de programación estructurada. Se espera que los aprendices analicen los requerimientos de la situación, identifiquen las estructuras lógicas y algoritmos necesarios para resolver el problema de manera eficiente.

Una vez completada la solución, los aprendices la compartirán en grupo para analizar diferentes enfoques, identificar oportunidades de mejora y discutir las mejores prácticas de codificación y optimización de algoritmos. Este ejercicio fomentará el desarrollo del pensamiento lógico y la aplicación efectiva de los conceptos fundamentales de programación.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje basado en proyectos
- Evaluación entre pares

**Materiales de formación:** Equipo de cómputo, un navegador web, acceso a internet, pantalla y/o televisor para proyectar, LMS (zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

**Evidencias de aprendizaje:**

- Taller de Funciones, Procedimientos y Archivos en Python.

**Instrumentos de evaluación:**

- Rúbrica de evaluación de Funciones, Procedimientos y Archivos en Python.

**Duración de la actividad:** 25 horas.





### **3.4 Actividades de Transferencia el Conocimiento: Desarrollo de un Proyecto Integrado con Programación Estructurada en Python**

**Descripción de la actividad:** En esta actividad, los aprendices presentarán y socializarán la solución propuesta para el desarrollo de un proyecto en Python, aplicando los principios de programación estructurada. Durante esta etapa, cada equipo explicará la arquitectura de su programa, detallando la organización del código, el uso de estructuras secuenciales, condicionales y repetitivas, así como la modularización de funciones para mejorar la eficiencia y reutilización del código.

Cada equipo deberá justificar su enfoque de solución, describiendo los requerimientos del problema planteado y demostrando cómo su implementación cumple con las especificaciones del caso de estudio. Se abordarán aspectos clave como la gestión de entrada y salida de datos, validaciones, optimización del flujo del programa y manejo de errores.

Además, los aprendices evidenciarán su proceso de desarrollo a través de diagramas de flujo, pseudocódigo y ejemplos de ejecución del programa, destacando cómo su solución mejora la legibilidad, mantenimiento y escalabilidad del software. Se fomentará el análisis crítico y la retroalimentación colaborativa, permitiendo que los participantes comparen enfoques, identifiquen oportunidades de mejora y validen la aplicabilidad de sus soluciones en escenarios reales.

**Ambiente requerido:** Ambiente de pluritecnológico con ventilación, mobiliario ergonómico, iluminación, acceso a internet, computadores de escritorio y/o portátil, pantalla y/o televisor para proyectar, LMS (zajuna).

**Estrategias o técnicas didácticas activas:**

- Aprendizaje basado en proyectos
- Presentaciones grupales
- Discusiones interactivas

**Materiales de formación:** computador con acceso a internet, pantalla y/o televisor para proyectar, LMS (zajuna).

**Material de apoyo:**

- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>



**Evidencias de aprendizaje:**

- Proyecto de Programacion Estructurada en Python.

**Instrumentos de evaluación:**

- Rúbrica de evaluación del Proyecto de Programacion Estructurada en Python.

**Duración de la actividad:** 16 horas.

**4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.**

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Analizar	Estructurar Procesos Logicos para la Construcción de Algoritmos y Bases de Datos	Actividad No.01 – Estructuras Secuenciales y Condicionales	Taller de Estructuras Secuenciales y Condicionales en Python	<ul style="list-style-type: none"> <li>• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.</li> <li>• Crea soluciones a problemas mediante algoritmos que incluyen estructuras secuenciales, condicionales y cíclicas</li> </ul>	ABP - Rúbrica
		Actividad No.02 – Estructuras Cíclicas y Manejo de Excepciones	Taller de Estructuras Cíclicas y Manejo de Excepciones en Python	<ul style="list-style-type: none"> <li>• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.</li> <li>• Crea soluciones a problemas mediante algoritmos que incluyen estructuras secuenciales, condicionales y cíclicas</li> </ul>	ABP - Rúbrica



		Actividad No.03 – Estructuras Almacenamiento	Taller de Estructuras Almacenamiento en Python	<ul style="list-style-type: none"> <li>• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.</li> <li>• Manipula arreglos en diferentes dimensiones para dar solución a problemas reales.</li> </ul>	ABP - Rúbrica
		Actividad No.04 – Funciones, Procedimientos y Archivos	Taller de Funciones, Procedimientos y Archivos en Python	<ul style="list-style-type: none"> <li>• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.</li> <li>• Crea funciones y procedimientos en la solución de algoritmos para ordenar y simplificar los códigos.</li> </ul>	ABP - Rúbrica
		Actividades de Transferencia	Informe técnico del Proyecto de programación estructurada en Python	<ul style="list-style-type: none"> <li>• Soluciona problemas de lógica proposicional incorporando habilidades propias en el como programador.</li> <li>• Administra la información de los usuarios por medio de archivos, permitiendo el ingreso, modificación y eliminación de los datos</li> </ul>	ABP - Rúbrica

## 5. GLOSARIO DE TÉRMINOS

- **Algoritmo:** Conjunto ordenado de pasos finitos que permiten resolver un problema o realizar una tarea específica.
- **Asignación:** Operación que asocia un valor a una variable mediante el operador = en Python.
- **Bloque de Código:** Conjunto de instrucciones agrupadas mediante indentación, que pertenecen a una misma estructura de control o función.



- **Comentario:** Texto dentro del código fuente que no se ejecuta y se utiliza para documentar el programa.
- **Condición:** Expresión booleana que determina si un bloque de código debe ejecutarse o no dentro de una estructura condicional.
- **Estructura de Control:** Conjunto de instrucciones que modifican el flujo de ejecución de un programa, como decisiones (if), bucles (for, while) y manejo de excepciones (try-except).
- **Estructura Secuencial:** Tipo de estructura en la que las instrucciones se ejecutan en el orden en que aparecen, sin bifurcaciones ni repeticiones.
- **Estructura Condicional:** Estructura de control que permite ejecutar diferentes bloques de código dependiendo del cumplimiento de una condición (if, elif, else).
- **Estructura Repetitiva (Bucles):** Mecanismo que permite ejecutar un bloque de código múltiples veces mientras se cumpla una condición (for, while).
- **Función:** Bloque de código reutilizable que realiza una tarea específica y puede recibir parámetros y devolver un valor.
- **Indentación:** Espaciado obligatorio en Python que indica el inicio y fin de bloques de código dentro de estructuras de control y funciones.
- **Input/Output (Entrada/Salida):** Métodos para recibir información del usuario (input()) y mostrar resultados en pantalla (print()).
- **Iteración:** Proceso de repetir un bloque de código dentro de un bucle hasta que se cumpla una condición de salida.
- **Lista:** Tipo de dato en Python que almacena múltiples valores en una secuencia ordenada y mutable.
- **Loop (Bucle):** Mecanismo que permite repetir la ejecución de un conjunto de instrucciones varias veces.
- **Módulo:** Archivo que contiene código reutilizable, como funciones y clases, que puede ser importado en otros programas.
- **Parámetro:** Variable que se define en una función para recibir valores externos cuando la función es llamada.
- **Return:** Palabra clave utilizada dentro de una función para devolver un valor al código que la llamó.
- **Tipo de Dato:** Clasificación de los valores en Python, como int (entero), float (decimal), str (cadena de texto) y bool (booleano).
- **Variable:** Nombre que se usa para almacenar un valor en memoria y poder manipularlo durante la ejecución del programa.
- **While:** Bucle que ejecuta un bloque de código mientras una condición sea verdadera.
- **For:** Bucle que recorre una secuencia de elementos, como listas o rangos numéricos, ejecutando un bloque de código en cada iteración.
- **Excepción:** Evento inesperado que interrumpe la ejecución de un programa, como errores en la entrada de datos o intentos de dividir por cero.
- **Break:** Palabra clave que interrumpe la ejecución de un bucle antes de que finalice.
- **Continue:** Palabra clave que salta la iteración actual de un bucle y continúa con la siguiente.



## 6. REFERENTES BIBLIOGRÁFICOS

- Arias, Ángel. Aprende a programar a python. Columbia, 2019.
- Buttu, Marco. El gran libro de Python. España : Marcombo, 2016.
- Salazar Perdomo, Patricia. Empezando a programar en Python / Bogotá : Editorial Escuela Colombiana de Ingeniería, ©2019.
- Shaw, Zed A. Aprenda a programar con python / Madrid : Anaya-Multimedia, 2014.
- Python Software Foundation. (s.f.). Python documentation. Python.org. Recuperado el 16 de febrero de 2025, de <https://www.python.org/doc/>
- El Libro de Python. (s.f.). Guías y recursos de Python. Recuperado el 16 de febrero de 2025, de <https://ellibrodepython.com/>
- Programarya. (s.f.). Curso de Python. Recuperado el 16 de febrero de 2025, de <https://www.programarya.com/Cursos/Python>
- Codecademy. (s.f.). Learn Python 3. Recuperado el 16 de febrero de 2025, de <https://www.codecademy.com/learn/learn-python-3>

## 7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
<b>Autor (es)</b>	Alvaro Pérez Niño	Instructor	Centro de Servicios y Gestión Empresarial	Febrero 06 de 2025

## 8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
<b>Autor (es)</b>					