

---

# PROGRAMA DE ESTUDIOS: CURSO DE INTRODUCCIÓN A PYTHON

---

# Área: Prerrequisitos

## Introducción a Python

### Información básica

#### [Información del curso]

1	Título	<i>Introducción a Python</i>
2	Año propuesto	<i>2024</i>
3	Semestre	<i>Primero (Agosto a Octubre)</i>
4	Departamento	<i>Centro de Innovación TIC (FPUNA)</i>
5	Nivel	<i>Básico</i>
6	Formato de clase	<i>Online</i>

#### [Horario y Lugar]

1	Días	<i>Viernes</i>
2	Horario	<i>de 19:00 a 21:30</i>
3	Ubicación	<i>Plataforma EDUCA</i>

#### [Información del instructor]

1	Nombre	<i>Prof. Mg. Carlos Filippi</i>
2	Oficina	<i>N/A</i>
3	Contacto	<i>carlippi@pol.una.py</i>
4	Contacto	<i>+595 981 403225</i>

#### [Perfil del docente]



Ingeniero Electromecánico por la Universidad Nacional de Asunción (UNA), con estudios de Maestría en Educación por la Universidad Ca' Foscari de Venecia y la Universidad Católica Nuestra Señora de la Asunción.

Actualmente, se encuentra elaborando su tesis del Doctorado en Educación con énfasis en la Gestión de la Educación Superior de la Universidad Nacional de Asunción.

Profesor Titular del Departamento de Informática de la Facultad Politécnica - UNA, posee amplia experiencia como docente universitario en carreras de grado y

programas de postgrado ocupando, además, diversos cargos directivos y de gestión en universidades nacionales.

Es miembro del IEEE (Institute of Electrical and Electronics Engineers) y la IEEE Computer Society y Par Evaluador para carreras de informática de la Agencia Nacional de Evaluación y Acreditación de la Educación Superior (ANEAES), tanto en el modelo nacional de Acreditación como en el modelo Arcusur. En el año 2017 fue distinguido por el Centro de Estudiantes de la Facultad Politécnica de la Universidad Nacional de Asunción como “Docente del Año”.

## Prerrequisitos

Ninguno.

## Descripción del curso

Este curso ofrece una introducción a la programación de computadores y se orienta a desarrollar habilidades para la solución de problemas y construcción de algoritmos eficaces, aplicando como herramienta principal el lenguaje Python.

En un mundo cada vez más digitalizado, la programación se ha convertido en una habilidad fundamental, no solo para los desarrolladores de software, sino para cualquier persona que quiera resolver problemas de manera eficiente, automatizar tareas y analizar datos de forma efectiva.

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue diseñado con la filosofía de facilitar la lectura y la escritura de código, lo que lo hace ideal para aquellos que están comenzando en el mundo de la programación.

Algunas de las ventajas de aprender Python incluyen:

- **Simplicidad y Legibilidad:** La sintaxis de Python es clara y concisa, lo que facilita la comprensión del código.
- **Comunidad Activa y Recursos Abundantes:** Existe una gran cantidad de documentación, tutoriales y foros en línea, lo que permite aprender y resolver problemas rápidamente.
- **Versatilidad:** Python se utiliza en diversos campos como desarrollo web, análisis de datos, inteligencia artificial, ciencia de datos, automatización de sistemas, y más.
- **Bibliotecas y Frameworks:** Cuenta con una amplia gama de bibliotecas y frameworks que simplifican la realización de tareas complejas.

# Objetivos del curso

## 1. Objetivo general:

- Desarrollar competencias básicas para resolver problemas mediante programas de computadora escritos en el lenguaje de programación Python.

## 2. Objetivos específicos:

- Desarrollar el enfoque algorítmico para abordar y resolver problemas de manera estructurada y eficiente.
- Comprender los conceptos básicos de la programación: variables, tipos de datos, operadores, estructuras de control de flujo y funciones.
- Escribir programas en Python aplicando buenas prácticas de codificación.
- Utilizar bibliotecas estándar y externas para realizar tareas comunes.

# Método de evaluación

La evaluación final de curso se realizará mediante un cuestionario que abordará aspectos conceptuales presentados y habilidades prácticas desarrolladas durante el curso.

## Condiciones para la aprobación:

<b>Mínimo para obtener el certificado</b>	<b>&gt; 70%</b>	<i>En base a la calificación obtenida en la evaluación final del curso.</i>
---	-----------------	---

# Actividades del curso

Para el desarrollo del curso se emplea el enfoque de “aprendizaje activo”. Cada clase síncrona contará con exposiciones magistrales y actividades prácticas a cargo de los participantes con la asistencia del docente. El alumno tendrá a disposición:

- La utilización de la plataforma Educa, la cual le permite el aprendizaje activo y colaborativo, con acceso a diferentes actividades y recursos de aprendizaje, que se detallan a continuación:
- Enlace a la sala de Google Meet que se emplea para el desarrollo de las clases síncronas.
- Acceso a materiales de estudio.
- Mensajería Interna, Foros de Debate y Chat.
- Problemas propuestos.

# Cronograma del curso

Semana	Tema	Tipo de clases	Materiales
1	Conceptos de Computación, Programas y Algoritmos.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
2	Lenguaje Python. Ambiente de programación.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
3	Instrucciones, Variables, Constantes, Operadores y Expresiones.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
4	Tipos de datos en Python.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
5	Estructuras de datos en Python.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea acciones</li> </ul>
6	Estructuras de control de flujo: selectivas y repetitivas.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
7	Funciones y módulos.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>
8	Concepto y ejemplos de librerías.	3 horas de clases virtuales síncronas y 2 horas de prácticas independientes del alumno	<ul style="list-style-type: none"> <li>● Presentaciones</li> <li>● Aula virtual</li> <li>● Banco de ejercicios</li> <li>● Foros en línea</li> </ul>

\* Un curso está diseñado para un periodo de 8 semanas y un total de 40 horas requeridas (5 horas por semana).

# Contenidos del curso

1. Concepto de algoritmo/programa.
2. Mi primer programa.
3. Entrada/Salida de datos.
4. Palabras reservadas, variables y operadores.
5. Expresiones aritméticas.
6. Tipos de datos numéricos, lógicos y cadenas.
7. Información y fuentes de consulta sobre Python.
8. Herramienta para escribir programas Python.
9. Problemas de aplicación.
10. Estructuras de selección: Selección simple, Selección doble, Selección múltiple.
11. Estructuras de repetición: while, for.
12. Programación estructurada.
13. Problemas de aplicación.
14. Método para construcción de programas.
15. Cadenas de caracteres.
16. Operaciones.
17. Funciones con cadenas.
18. Problemas de aplicación.
19. Estructura de datos lista.
20. Propiedades de las listas.
21. Operaciones con listas.
22. Problemas de aplicación.
23. Creación de funciones y módulos.
24. Estructuras de datos propias de Python (Colecciones): Listas, Conjuntos, Tuplas, Diccionarios.
25. Librería estándar, Librería NumPy, pandas y Mathplotlib
26. Problemas de aplicación.