

## Fundamentos de programación en Python

<b>Descripción</b>	Al finalizar el módulo habrán desarrollado una visión general del proceso de desarrollo de código, además, serán capaces de leer e implementar algoritmos a partir de un diagrama de flujos con el lenguaje Python, así como desarrollar algoritmos comunes de control de flujo y utilizar estructuras de datos como listas, arreglos y conjuntos para resolver problemas de procesamiento de datos y finalmente desplegar datos de Python en la consola.
<b>Competencia general</b>	Codificar piezas de software de baja complejidad utilizando lenguaje Python para resolver problemáticas comunes de acuerdo a las necesidades de la industria.

Unidades de aprendizaje				
Unidad	Competencia	Resultados de aprendizaje	Contenidos	Duración
Introducción a Python	Reconocer los conceptos fundamentales del lenguaje Python para la construcción de programas	<ul style="list-style-type: none"> <li>Reconoce conceptos básicos del lenguaje Python, sintaxis, indentación y estructura.</li> <li>Distingue características propias del lenguaje Python versus otros lenguajes.</li> <li>Reconoce el entorno de ejecución y las herramientas complementarias de Python para el desarrollo.</li> </ul>	<b>Sincrónico</b> Conociendo Python <ul style="list-style-type: none"> <li>Qué es Python</li> <li>¿Qué tan relevante es Python en la industria?</li> <li>Áreas donde se utiliza Python</li> <li>Interpretado</li> <li>Dinámico o dinámicamente tipado</li> <li>Multiplataforma</li> <li>Definiciones importantes</li> </ul>	2 horas
			<b>Sincrónico</b> Entorno de ejecución <ul style="list-style-type: none"> <li>Descargando e instalando Anaconda</li> <li>Mi primer código en Python</li> <li>Salida de datos: print()</li> <li>Comentarios</li> <li>Comentarios en múltiples líneas</li> <li>Python como una calculadora</li> <li>Limitantes</li> </ul>	2 horas

	Distinguir los tipos de datos y sentencias básicas del lenguaje para la construcción de programas.	<ul style="list-style-type: none"> <li>Reconoce los tipos de datos usados en el entorno Python y su uso para la construcción de un programa.</li> <li>Reconoce los operadores matemáticos, lógicos y de comparación para la construcción de expresiones.</li> <li>Reconoce las sentencias básicas del lenguaje como condicionales y bucles para la construcción de programas.</li> </ul>	<b><u>Sincrónico</u></b> Tipos de datos y valores <ul style="list-style-type: none"> <li>Tipos de datos               <ul style="list-style-type: none"> <li>Valores Numéricos</li> <li>Strings</li> <li>Concatenación</li> <li>Duplicación</li> </ul> </li> <li>Métodos: Count (), Upper(), Lower(), Title(), Length(), Join()</li> <li>El salto de línea</li> <li>Valores Booleanos</li> <li>Variables               <ul style="list-style-type: none"> <li>Asignando un valor a una variable</li> <li>Manipulando variables</li> </ul> </li> <li>Transformando los datos</li> <li>Interpolación</li> <li>Precisión de Decimales</li> <li>Ingresando datos de manera interactiva</li> <li>Ingresando datos con input</li> </ul>	2 horas
			<b><u>Sincrónico</u></b> Operadores y librerías <ul style="list-style-type: none"> <li>Otros operadores Matemáticos</li> <li>Precedencia de operadores</li> <li>Orden de las operaciones</li> <li>Operaciones y paréntesis</li> <li>Librerías / Módulos</li> <li>Importar una librería</li> <li>Instalar librerías</li> <li>Instalar librerías</li> <li>pip</li> </ul>	2 horas
			<b><u>Asincrónico</u></b> Desafío evaluado	2 horas
			<b><u>Sincrónico</u></b> Revisión desafío	2 horas

Sentencias condicionales e iterativas	Utilizar sentencias condicionales para el control del flujo de un algoritmo que resuelve un problema simple acorde al lenguaje Python.	<ul style="list-style-type: none"> <li>• Codifica una rutina simple en Python a partir de un diagrama de flujo para dar solución a un problema.</li> <li>• Utiliza estilos y convenciones de programación para la elaboración de un código acorde a las buenas prácticas del lenguaje Python.</li> <li>• Utiliza instrucciones condicionales en un algoritmo para dar solución a un problema acorde al lenguaje Python.</li> </ul>	<b>Sincrónico</b> Algoritmos <ul style="list-style-type: none"> <li>• Algoritmo</li> <li>• Diagrama de flujo</li> <li>• Pseudocódigo</li> </ul>	2 horas
			<b>Sincrónico</b> Identificar los componentes de los diagramas de flujo <ul style="list-style-type: none"> <li>• Diagramas de flujo</li> <li>• Pseudocódigo</li> </ul>	2 horas
			<b>Asincrónico</b> Representar algoritmos de baja complejidad	2 horas
			<b>Sincrónico</b> Pruebas Lógicas <ul style="list-style-type: none"> <li>• Operaciones o pruebas lógicas</li> <li>• Los operadores más conocidos</li> <li>• Diferencia entre = y ==</li> </ul>	2 horas
			<b>Sincrónico</b> Control de Flujo <ul style="list-style-type: none"> <li>• La instrucción IF</li> <li>• La instrucción ELSE</li> <li>• La instrucción ELIF</li> </ul>	2 horas
			<b>Asincrónico</b> Desafío evaluado	2 horas
			<b>Sincrónico</b> Revisión desafío	2 horas

	Utilizar sentencias iterativas para la elaboración de un algoritmo que resuelve un problema acorde al lenguaje Python.	<ul style="list-style-type: none"> <li>• Utiliza sentencias FOR y WHILE para la elaboración de un algoritmo iterativo que resuelve un problema acorde al lenguaje Python.</li> <li>• Utiliza ciclos anidados y condiciones de salida para resolver un problema acorde al lenguaje Python.</li> <li>• Utiliza ciclos de instrucciones iterativas combinadas con sentencias if/else para resolver un problema acorde al lenguaje Python.</li> </ul>	<b>Sincrónico</b> El Ciclo While <ul style="list-style-type: none"> <li>• Introducción a Ciclos</li> <li>• Ciclo While</li> <li>• Iterar</li> <li>• Contando con while</li> <li>• Operadores de asignación</li> <li>• Contadores y Acumuladores</li> </ul>	2 horas
			<b>Sincrónico</b> El Ciclo For <ul style="list-style-type: none"> <li>• Ciclo For</li> <li>• Iterables</li> <li>• Utilizando estructuras de datos en un ciclo For</li> <li>• Otras funciones útiles al momento de iterar</li> <li>• Ciclos Anidados</li> </ul>	2 horas
			<b>Sincrónico</b> Python Comprehensions <ul style="list-style-type: none"> <li>• Transformando un ciclo for en un Comprehension</li> <li>• Condicionales con List Comprehensions</li> <li>• Operaciones de Filtrado</li> <li>• Dictionary Comprehensions</li> </ul>	2 horas
			<b>Asincrónico</b> Desafío evaluado	2 horas
			<b>Sincrónico</b> Revisión desafío	2 horas

Estructuras de datos y funciones	Utilizar estructuras de datos apropiadas para la elaboración de un algoritmo que resuelve un problema acorde al lenguaje Python.	<ul style="list-style-type: none"> <li>• Identifica las características de las distintas estructuras de datos para la resolución de problemas.</li> <li>• Utiliza operaciones de creación y acceso a los elementos de una estructura de datos acorde al lenguaje Python para resolver un problema.</li> <li>• Utiliza operaciones para la agregación, modificación y eliminación de elementos de una estructura de datos acorde al lenguaje Python para resolver un problema.</li> </ul>	<b>Sincrónico</b> Introducción a Estructuras de Datos <ul style="list-style-type: none"> <li>• Introducción a Listas</li> <li>• Índices</li> <li>• ARGV</li> <li>• Leyendo la documentación de listas</li> <li>• Métodos aplicables a listas: <code>append(x)</code> , <code>insert(i, x)</code> , <code>pop()</code> , <code>remove(x)</code> , <code>reverse()</code> , <code>sort()</code> , <code>index()</code></li> <li>• Operaciones: Concatenación y repetición de listas</li> </ul>	2 horas
			<b>Sincrónico</b> Introducción a los diccionarios (Parte I) <ul style="list-style-type: none"> <li>• Diccionarios</li> <li>• Lista versus Diccionario</li> <li>• Crear un diccionario</li> <li>• Acceder a un elemento dentro de un diccionario</li> <li>• Agregando un elemento a un diccionario</li> <li>• Cambiando un elemento dentro de un diccionario</li> <li>• Eliminar elementos de un diccionario</li> <li>• Unir diccionarios</li> <li>• Cuidado con las colisiones</li> </ul>	2 horas
			<b>Sincrónico</b> Introducción a los diccionarios (Parte II) <ul style="list-style-type: none"> <li>• Otros Métodos para diccionarios: <code>keys()</code> , <code>values()</code> , <code>items()</code> , <code>get()</code></li> <li>• Otras Estructuras de Datos: Tuplas y Sets</li> <li>• Convertir estructuras</li> <li>• Convertir un diccionario en una lista</li> <li>• Convertir una lista en un diccionario</li> <li>• La función <code>dir()</code></li> <li>• Otras funciones en Python: <code>sum()</code> , <code>max()</code> , <code>min()</code></li> </ul>	2 horas

Codificar un programa utilizando funciones para la reutilización de código acorde al lenguaje Python.	<ul style="list-style-type: none"> <li>Define funciones que utilizan parámetros de entrada y que producen un retorno para resolver un problema.</li> <li>Explica el alcance de una variable dentro y fuera de una función distinguiendo el concepto de variable local y global.</li> <li>Explica el sentido de utilizar funciones dentro de un programa distinguiendo su definición versus su invocación.</li> <li>Utiliza funciones preconstruidas y personalizadas por el usuario con paso de parámetros y que obtienen un retorno.</li> </ul>	<b>Asincrónico</b> Guía de ejercicios	2 horas
		<b>Asincrónico</b> Desafío evaluado	2 horas
		<b>Sincrónico</b> Revisión desafío	2 horas
		<b>Sincrónico</b> Funciones y variables <ul style="list-style-type: none"> <li>Necesidad de Funciones</li> <li>Principio DRY</li> <li>Parámetros y Argumentos</li> <li>Múltiples Retornos</li> <li>Tipos de Argumentos</li> <li>Funciones como argumentos</li> <li>Parámetros obligatorios y opcionales</li> <li>*Args y **Kwargs</li> <li>Variables locales y globales</li> </ul>	2 horas
		<b>Asincrónico</b> Desafío evaluado	2 horas
		<b>Sincrónico</b> Revisión desafío	2 horas
		<b>Sincrónico</b> Organización de un Proyecto en Python y modularización <ul style="list-style-type: none"> <li>Docstrings: Google, Sphinx, Docblockr, Numpy</li> <li>Refactorización</li> <li>Modularización</li> <li>Experiencia de Usuario: Pausas, Limpiar la Pantalla, Terminar el Programa</li> </ul>	2 horas

	Consumir APIs, con el fin de extraer datos externos para su utilización dentro de una aplicación.	<ul style="list-style-type: none"> <li>• Conecta a APIs mediante Python para extraer información relevante para su aplicación.</li> <li>• Aplica los distintos verbos REST para interactuar con la API de manera apropiada.</li> <li>• Aplica los conceptos básicos del lenguaje Python a través de la utilización de la información extraída mediante API para agregar versatilidad y robustez a sus aplicaciones.</li> </ul>	<b>Asincrónico</b> Desafío evaluado	2 horas
			<b>Sincrónico</b> Revisión desafío	2 horas
			<b>Sincrónico</b> Introducción a APIs <ul style="list-style-type: none"> <li>• Tipos de APIs existentes</li> <li>• Cómo se usa una API</li> <li>• Probando una API con Postman</li> <li>• Consumiendo una API desde Python</li> <li>• El código de la respuesta</li> <li>• El contenido de la respuesta</li> <li>• Transformar el request en una función</li> </ul>	2 horas
			<b>Sincrónico</b> Seguridad en las API <ul style="list-style-type: none"> <li>• Recursos</li> <li>• Las Rutas</li> <li>• Creando un recurso</li> <li>• Actualizando un recurso</li> <li>• Seguridad en las API <ul style="list-style-type: none"> <li>○ Encriptación por SSL</li> <li>○ Otros tipos de autenticación</li> <li>○ Precauciones sobre los token</li> </ul> </li> </ul>	2 horas
			<b>Asincrónico</b> Desafío guiado	2 horas
			<b>Asincrónico</b> Prueba	4 horas
			<b>Sincrónico</b> Revisión Prueba	2 horas

Implementación		
Recursos de Información	Características notebook*	Plataformas y Software**
	<ul style="list-style-type: none"> <li>● Sistema Operativo: Windows, Linux o Mac</li> <li>● Procesador: Intel Core I3</li> <li>● RAM: 8 GB, 128 SSD</li> </ul>	<ul style="list-style-type: none"> <li>● LMS</li> <li>● Navegador Google Chrome y Mozilla Firefox</li> <li>● Visual Studio Code</li> <li>● PSEINT</li> <li>● Terminal</li> <li>● POSTMAN</li> <li>● Python 3.7 o superior</li> </ul>

\* El notebook es por cuenta de todos los participantes: docente, ayudante y estudiantes.

\*\* Programas open source, por lo que el estudiante no necesita incurrir en gastos de licencias.