

DESARROLLO DE SOFTWARE CON PYTHON FULL STACK



PRESENTACIÓN DE PARTICIPANTES

1. Nombre y Como le gusta que lo llamen
2. De que empresa viene
3. Cargo
4. Porque decidió estudiar Python

EVALUACIÓN DE CONOCIMIENTOS PREVIOS DE LOS PARTICIPANTES

<https://forms.gle/H2a4MTA5Cq99fsb98>

1. Nociones de programación fundamentales como programación estructurada y orientada a objetos
2. Conocimientos de desarrollo web en lenguajes del lado del cliente como HTML, CSS y Javascript
3. Conocimientos de desarrollo web en lenguajes del lado del servidor como PHP, ASP .NET
4. Conocimientos sobre servicios web REST API
5. Manejo de bases de datos relaciones
6. Manejo de servidores web
7. Uso de Git

HERRAMIENTA PARA DOCUMENTAR EL CURSO

1. CherryTree (Windows)
2. <https://www.giuspen.com/cherrytree/>
3. Obsidian (MAC OSX)
4. <https://obsidian.md/>

MODULO 1. INTRODUCCIÓN AL DESARROLLO DE SOFTWARE CON PYTHON

1. PROGRAMACIÓN ESTRUCTURADA CON PYTHON

CONTENIDOS A DESARROLLAR

1. Fundamentos de los lenguajes de programación
2. Conceptos fundamentales
3. Entorno de desarrollo
4. Estructuras de control

FUNDAMENTOS DE LOS LENGUAJES DE PROGRAMACIÓN

1. Un lenguaje de programación es un lenguaje que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático
- Wikipedia

LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL

1. Alfabeto
2. Léxico
3. Sintaxis
4. Semántica

Un programa escrito en un lenguaje de programación de alto nivel se llama **código fuente**

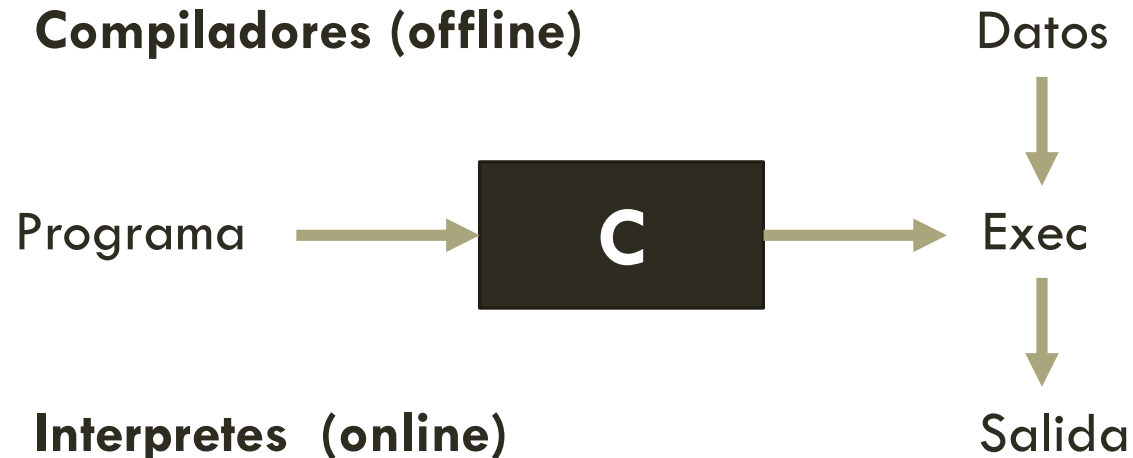
Programa fuente  Traductor  Programa objetivo

COMPILADOS VS INTERPRETADOS

?

COMPILACIÓN VS INTERPRETACIÓN

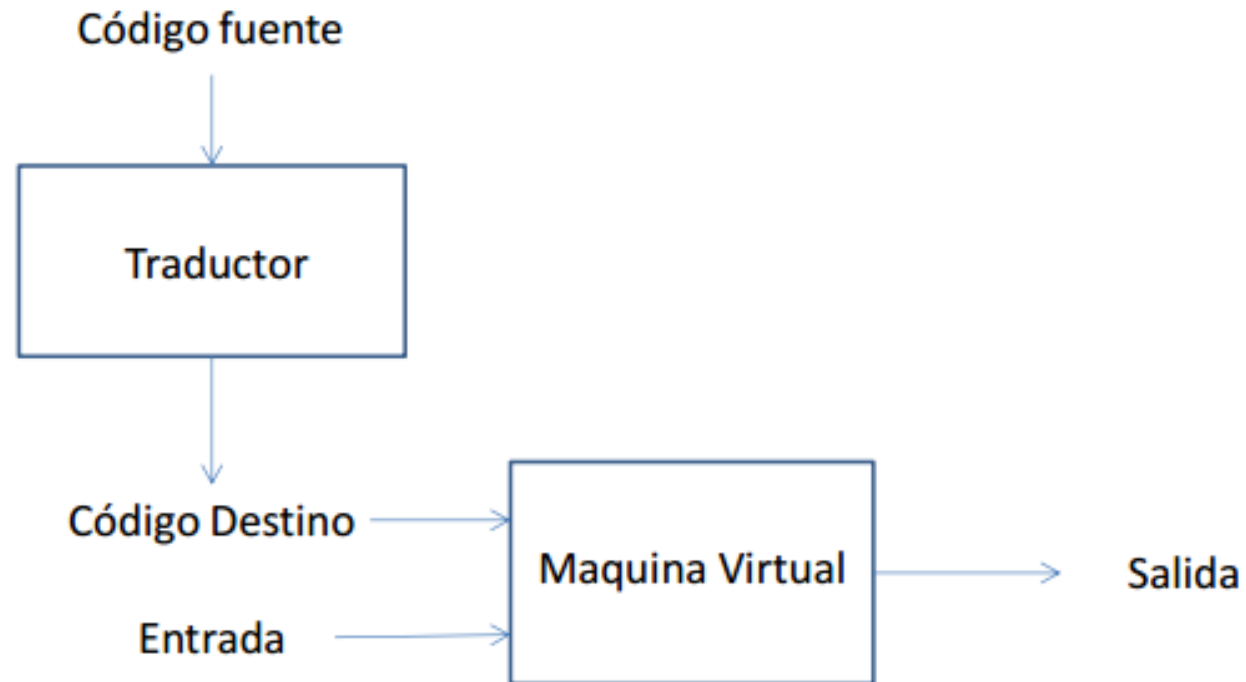
Compiladores (offline)



Interpretes (online)



HÍBRIDOS



Y PYTHON?

- Python es interpretado
- Es gratis
- Python es un lenguaje de programación de alto nivel, interpretado, orientado a objetos y de uso generalizado con semántica dinámica, que se utiliza para la programación de propósito general.

CONCEPTOS FUNDAMENTALES

- Que es Python?
- Python es un lenguaje de programación de alto nivel, multiplataforma y ampliamente documentado, con una curva de aprendizaje muy baja y desarrollos eficiente de proyectos

- Usos de Python

Desarrollo Web

Ciencia y Educación

Desarrollo de Interfaces Gráficos

Desarrollo Software.

Machine Learning

Visualización de Datos

Finanzas y Trading

Ethical Hacking

QUE HACE ESPECIAL A PYTHON

Hay muchas razones. Ya hemos enumerado algunas de ellas, pero vamos a enumerarlas de una manera más práctica:

Es **fácil de aprender** - El tiempo necesario para aprender Python es más corto que en muchos otros lenguajes; esto significa que es posible comenzar la programación real más rápido.

Es **fácil de enseñar** - La carga de trabajo de enseñanza es menor que la que necesitan otros lenguajes; esto significa que el profesor puede poner más énfasis en las técnicas de programación generales (independientes del lenguaje).

Es **fácil de utilizar** - Para escribir software nuevo; a menudo es posible escribir código más rápido cuando se usa Python.

Es **fácil de entender** - A menudo, también es más fácil entender el código de otra persona más rápido si está escrito en Python.

Es **fácil de obtener, instalar y desplegar** - Python es gratuito, abierto y multiplataforma.

PYTHON 2 VS PYTHON 3

- Estas dos versiones de Python no son compatibles entre sí.
- Las secuencias de comandos de Python 2 no se ejecutarán en un entorno de Python 3 y viceversa
- Por lo que si deseas que un intérprete de Python 3 ejecute el código Python 2 anterior, la única solución posible es volver a escribirlo, no desde cero, por supuesto

TIPADO DINÁMICO VS TIPADO ESTÁTICO

- *Tipado*. Es la definición del tipo que tendrá una variable dada.
- El *tipado estático* nos obliga a definir desde el principio el tipo de una variable, ejemplos de lenguajes con tipado estático son C, C++, Pascal, Java, Objective-C, C#, entre otros.
- El *tipado dinámico* nos da la facilidad de no definir los tipos al declarar una variable, algunos ejemplos son PHP, JavaScript, Groovy, Python, Perl, entre otros.

INFERENCIA DE TIPOS

- Un **Sistema de Tipos** de un lenguaje especifica cuales operaciones son válidas para los tipos.
- La inferencia de tipos asigna automáticamente un tipo de datos a una expresión sin necesidad de que el programador lo escriba.
- $(e_1: \text{int} \wedge e_2: \text{int}) \Rightarrow e_1 + e_2: \text{Int}$

TIPADO FUERTE VS TIPADO DÉBIL (NO TIPADOS)

- Un lenguaje de programación es **fuertemente tipado** si no se permiten violaciones de los tipos de datos, es decir, dado el valor de una variable de un tipo concreto, no se puede usar como si fuera de otro tipo distinto a menos que se haga una conversión.
- Python y Java
- Los lenguajes de programación **no tipados** o **débilmente tipados** no controlan los tipos de las variables que declaran, de este modo, es posible usar variables de cualquier tipo en un mismo escenario. Por ejemplo, una función puede recibir como parámetro un valor entero, cadena de caracteres, flotante, etc.
- JavaScript y PHP

PYINSTALLER

- Como generamos un ejecutable?

PYINSTALLER

- PyInstaller lee un script de Python escrito. Analiza el código para descubrir todos los demás módulos y bibliotecas que el script necesita para ejecutarse. Luego recoge copias de todos esos archivos - ¡incluyendo el intérprete de Python activo - y los coloca con el script en una sola carpeta, u opcionalmente en un solo archivo ejecutable.
- PyInstaller está probado en Windows, Mac OS X y GNU/Linux. Sin embargo, no es un compilador cruzado: para hacer una aplicación para Windows se ejecuta PyInstaller en Windows; para hacer una aplicación para GNU/Linux se ejecuta en GNU/Linux, etc.

ENTORNO DE DESARROLLO - PYTHON

- Python
- Instalar Python en Windows

Descargarlo desde <https://www.python.org/downloads/>

- users/appdata es la carpeta por si no aparece en el path

ENTORNO DE DESARROLLO — SUBLIME TEXT

- Descargar de: <https://www.sublimetext.com/>
- Poner sublime en modo python3
- ctrl + shift + p
- install package -> package control: install package -> python 3
- tools->build system->python3
- ctrl + shift + p
- install package -> package control: install package -> all autocomplete
- View->sidebar->show open files
- Crear carpeta en documentos y crear archivo dentro con touch python.py
- open folder->abrir carpeta
- ctrl + shift + p
- install package -> package control: install package -> File Icons

ENTORNO DE DESARROLLO - PYCHARM

- Python + PyCharm
- Instalar Python en Windows

Descargarlo desde <https://www.python.org/downloads/>

- Instalar Entorno de desarrollo Pycharm
- Existen muchos, como [Atom](#), [Sublime Text](#) o [Visual Studio Code](#), pero nosotros usaremos [PyCharm](#).
- <https://www.jetbrains.com/es-es/pycharm/download/>

ENTORNO DE DESARROLLO — IDLE Y JUPYTER

Además de sus muchos componentes útiles, la instalación estándar de Python 3 contiene una aplicación muy simple pero extremadamente útil llamada IDLE.

IDLE es un acrónimo de: Integrated Development and Learning Environment (Desarrollo Integrado y Entorno de Aprendizaje).

Sin instalación

- Accede a <https://jupyter.org/try> y busca “**Try JupyterLab**”.

PRIMER PROGRAMA EN PYTHON

- Seguir indicaciones del instructor
- Definición de variables

ESTRUCTURAS DE CONTROL

- Las estructuras de control, permiten **cambiar el flujo de ejecución de un programa**, haciendo que ciertos bloques de código se ejecuten si y solo si se dan unas condiciones particulares o se repitan una cantidad determinada o indeterminada de veces.

- If
- Else
- For
- while

y también algo más avanzados, como lo son los

- iteradores
- clases iterables y
- uso del break/continue/try.

DESARROLLO DE EJEMPLOS

Seguir indicaciones del instructor

PREGUNTAS?

Gracias por su atención!!!!