Logo

Description automatically generated

Tendencias en Desarrollo de Aplicaciones – INS340-01

Prof. Lorenzo Solano Martínez

**LENGUAJES DE PROGRAMACIÓN - PYTHON**

Presentado por:

Víctor Peña Tavárez - ID 1095406

7 de noviembre de 2022

**Lenguajes de Programación - Python**

**I) Completar ficha del lenguaje Python**

* **Propósito**

Propósito general

**• Paradigma**

Programación orientada a objetos: Los objetos se pueden definir simplemente como la instancia de una clase que contiene tanto los miembros de datos como las funciones del método.

En los paradigmas de programación orientada a procedimientos: en este paradigma, series de pasos computacionales se dividen en módulos, lo que significa que el código se agrupa en funciones y el código se ejecuta en serie paso a paso, por lo que básicamente combina el código en serie para instruir a una computadora con cada paso para realizar una determinada tarea.

Paradigmas de programación funcional: Es un paradigma en el que todo se vincula en un estilo de funciones matemáticas puras. Utiliza la función matemática y trata cada declaración como una expresión funcional, ya que una expresión se ejecuta para producir un valor.

**• Fortalezas y debilidades**

**Fortalezas**

* Es fácil de aprender: el tiempo necesario para aprender Python es más corto que para muchos otros idiomas;
* Es fácil de usar para escribir software nuevo: a menudo es posible escribir código más rápido cuando se usa Python;
* Es fácil de obtener, instalar y desplegar: Python es gratuito, abierto y multiplataforma;
* Fuerte comunidad de programadores.
* Biblioteca extensa

**Debilidades**

* Consumo de memoria alto debido a la flexibilidad de los tipos de datos;
* Lentitud debido a la naturaleza dinámica.

**• Implementación**

Un programa en Python primero se compila y luego se interpreta. La parte de compilación está oculta para el programador, por lo que muchos programadores creen que es un lenguaje interpretado. La parte de compilación se realiza primero cuando ejecutamos nuestro código y esto generará un código de bytes e internamente este código de bytes se convierte mediante la máquina virtual de python (p.v.m) de acuerdo con la plataforma subyacente (máquina + sistema operativo).

**• Runtime / Máquina Virtual**

El entorno de **runtime** es python.exe o /usr/bin/python. Es el ejecutable de Python el que interpretará su código de Python transformándolo en un código de bytes legible por CPU. Cuando realiza múltiples subprocesos, solo tiene un python en ejecución. Cuando multiprocesas, tienes varias pitones ejecutándose.

Por otra parte, la función de Python Virtual Machine (PVM) es convertir las instrucciones de código de bytes en código de máquina para que la computadora pueda ejecutar esas instrucciones de código de máquina y mostrar el resultado final. Para llevar a cabo esta conversión, PVM está equipado con un intérprete.

**• Plataformas soportadas**

Plataformas bien soportadas en Python 3.7 y 2.7:

* Linux
* Windows 8 y más reciente para Python 3.9
* FreeBSD 10 y más reciente
* macOS Snow Leopard (macOS 10.6, 2008) y más reciente

Plataforma compatible con best effort support:

* API de Android 24
* OpenBSD
* NetBSD
* AIX 6.1 y más reciente

**• Usuarios notables: productos, empresas, etc.**

**Empresas**

Python es un lenguaje de programación utilizado empresas y gigantes tecnológicos como Google, Facebook, Cisco, Netflix y más.

**Productos**

* Desarrollo web e Internet (por ejemplo, frameworks Django y Pyramid, microframeworks Flask y Bottle);
* Desarrollo de software (control de compilación, administración y pruebas: Scons, Buildbot, Apache Gump, Roundup, Trac);
* Aplicaciones empresariales (ERP y sistemas de comercio electrónico, Odoo, Tryton);
* Juegos (por ejemplo, la serie Battlefield);
* Sitios web y servicios (por ejemplo, Dropbox, UBER, Pinterest, BuzzFeed).

**• Historia: Creadores, primer y último release**

Python fue creado a principios de la década de 1990 por Guido van Rossum en Stichting Mathematisch Centrum en los Países Bajos como sucesor de un lenguaje llamado ABC. Guido sigue siendo el autor principal de Python, aunque incluye muchas contribuciones de otros.

**First release**: Python 0.9.0, 20 de febrero 1991

**Last release**: Python 3.11.0, 24 de octubre de 2022

* **Estado: Activo, Muerto, etc.**

Activo

**Referencia**

GeeksforGeeks. (2020, 21 abril). *Programming Paradigms in Python*. https://www.geeksforgeeks.org/programming-paradigms-in-python/

*History and License — Python 3.11.0 documentation*. (s. f.). Recuperado 5 de noviembre de 2022, de https://docs.python.org/3/license.html

*Supported platforms and architectures — Unofficial Python Development (Victor’s notes) documentation*. (s. f.). Recuperado 5 de noviembre de 2022, de https://pythondev.readthedocs.io/platforms.html

*About Python*. (s. f.). Recuperado 5 de noviembre de 2022, de https://pythoninstitute.org/about-python

*Python Documentation by Version*. (s. f.-b). Python.org. Recuperado 5 de noviembre de 2022, de https://www.python.org/doc/versions/

*What is runtime in context of Python? What does it consist of?* (2020, 18 febrero). Stack Overflow. https://stackoverflow.com/questions/60273813/what-is-runtime-in-context-of-python-what-does-it-consist-of

KeepCoding, R. (2022, 12 septiembre). *Ventajas y Desventajas de Python*. KeepCoding Tech School. https://keepcoding.io/blog/ventajas-y-desventajas-de-python/