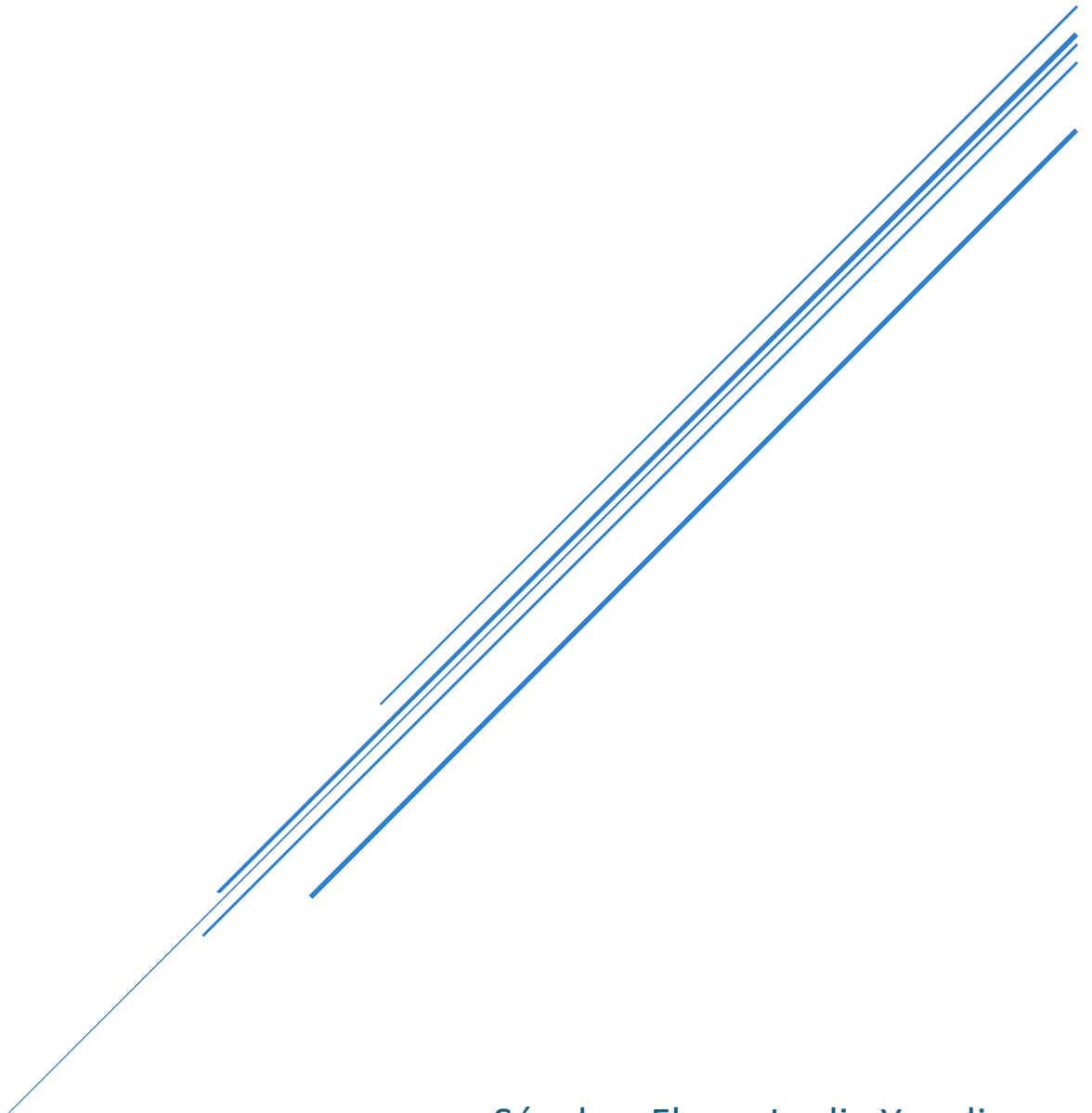


PYTHON

Introducción a Ingeniería en Sistemas Computacionales



Sánchez Flores Leslie Yoselin
201831

¿Qué es Python?

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Python es conocido por su sintaxis clara y legible, lo que facilita el aprendizaje y la escritura de código. Algunas características clave de Python incluyen:

Sintaxis sencilla: Python utiliza una sintaxis que es fácil de leer y escribir, lo que lo hace ideal para principiantes.

Multiparadigma: Python admite programación orientada a objetos, programación funcional y programación procedimental.

Bibliotecas extensas: Python cuenta con una gran cantidad de bibliotecas y módulos que permiten realizar una amplia variedad de tareas, desde desarrollo web hasta análisis de datos y aprendizaje automático.

Portabilidad: Python es compatible con múltiples plataformas, incluyendo Windows, macOS y Linux.

Comunidad activa: Python tiene una comunidad de desarrolladores muy activa que contribuye al desarrollo de nuevas herramientas y recursos.

Lenguajes interpretados

Un lenguaje interpretado es un tipo de lenguaje de programación en el que el código fuente se ejecuta directamente por un intérprete, en lugar de ser compilado a código máquina antes de su ejecución. Esto significa que el código se traduce y ejecuta línea por línea, lo que puede ofrecer ciertas ventajas y desventajas:

Ventajas de los lenguajes interpretados:

Portabilidad: Los lenguajes interpretados suelen ser más portables, ya que el intérprete puede ejecutarse en diferentes plataformas sin necesidad de recompilar el código.

Desarrollo rápido: La ejecución directa del código fuente permite una rápida iteración y prueba de código, lo que puede acelerar el desarrollo.

Flexibilidad: Los lenguajes interpretados suelen ser más flexibles y dinámicos, permitiendo cambios en el código durante la ejecución.

Desventajas de los lenguajes interpretados:

Rendimiento: La ejecución línea por línea puede ser más lenta en comparación con los lenguajes compilados, que se traducen a código máquina antes de la ejecución.

Errores en tiempo de ejecución: Los errores en el código pueden no ser detectados hasta que se ejecuta la línea problemática, lo que puede dificultar la depuración.

Ejemplos de lenguajes interpretados

Además de Python, otros ejemplos de lenguajes interpretados incluyen:

JavaScript: Utilizado principalmente para desarrollo web.

Ruby: Conocido por su simplicidad y elegancia.

PHP: Utilizado para desarrollo web del lado del servidor.

Perl: Utilizado para procesamiento de texto y administración de sistemas.

Referencias:

Diferencia entre lenguajes de programación compilados e interpretados | Aulab Hackademy.
(s. f.). aulab.es. <https://aulab.es/noticia/18/diferencia-entre-lenguajes-de-programacion-compilados-e-interpretados>

Erick. (2022, 23 septiembre). Lenguajes de programación interpretados vs compilados: ¿Cuál es la diferencia? freeCodeCamp.org.
<https://www.freecodecamp.org/espanol/news/lenguajes-compilados-vs-interpretados/>