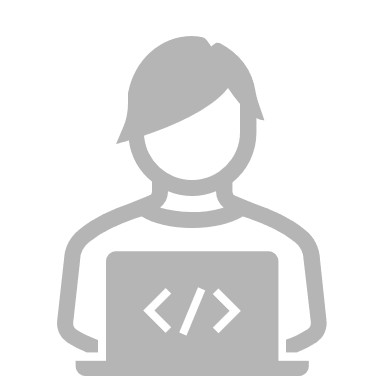
**Trabajo practico Nº1**

## **Curso K-2055**

Integrantes:

Tobias Viale

Natalia Lembo

Juan Cruz Neira

Román de Vita

Javier Capristo

### A lo largo de este trabajo practico se buscará desarrollar una pequeña investigación sobre los lenguajes de programación Python y Java. Se comentará un poco los orígenes de ambos lenguajes, junto con sus principales características, y se procederá a realizar una comparación de rendimiento entre ellos. Adicionalmente, se buscaron las BNF las cuales se encontrarán anexadas al final de este trabajo.

## **Un poco de la historia de Java**

Java nace como un lenguaje de programación fácil de utilizar: algunos de los programadores de Sun Microsystems, cansados de “pelear” con C++, obtienen la autorización de sus jefes para desarrollar un lenguaje de programación que sea muy sencillo de utilizar, y que pueda ser ejecutado sobre dispositivos pequeños: televisores, electrodomésticos, o cualquier implemento eléctrico instalado en casa. Este proyecto empieza a demandar tiempo y dinero, y se obtiene un primer producto, OAK, con el cual no se logra la aceptación esperada, y el proyecto queda descartado. Un tiempo después, con la aparición de WWW, se piensa que el proyecto debe ser revaluado, y se le da un nuevo impulso, apareciendo Java.Pero, ¿qué es lo que se presenta como Java? Lo que Sun presenta al mundo como su nuevo lenguaje de programación es algo más: es un lenguaje de programación orientado a objetos, que además incluye una máquina virtual, y una serie de desarrollos básicos que pueden ser empleados por los programadores para simplificar sus nuevos desarrollos; además permite la inclusión de porciones de código ejecutables en las páginas que se publican en internet, a través de WWW. Sun permite también que otros fabricantes de software tomen a Java como el centro de sus nuevas herramientas de software, es decir, desarrollen entornos de programación considerando a Java como su corazón; de manera que en un tiempo relativamente corto IBM, Borland, Oracle y muchos otros (incluyendo a Microsoft) están desarrollando herramientas que les permitan ofrecer productos portables entre diferentes máquinas y sistemas operativos, pues todos compilan para la misma máquina: la máquina virtual de Java.

Entre sus principales características, algunas de las que ya se mencionaron, se encuentran:

* Independiente de la arquitectura: Gracias a la máquina virtual de Java (Java Virtual Machine JVM) se puede ejecutar el mismo programa en cualquier sistema operativo (Linux, Windows, MacOS, etc.) y en cualquier hardware sin necesidad de hacer modificaciones sobre el mismo. Esto convierte a Java en un lenguaje multiplataforma y 100% portable.
* Lenguaje orientado a objetos: permite la definición de clases y la instancia de objetos de estas. Además, tiene todas las características comunes de la programación orientada a objetos: polimorfismo, herencia y enlazado dinámico.
* Interpretado y compilado: Java es un lenguaje particular porque es compilado, pero es compilado a un lenguaje intermedio llamado bytecode, que después es interpretado. Los creadores de Java querían crear un lenguaje compilado, pero que se pudiera ejecutar en cualquier sistema operativo y procesador sin necesidad de crear varios ejecutables.
* Multihilo: Java es capaz de ejecutar múltiples tareas de forma simultánea, lo que reduce el tiempo de ejecución y mejora el rendimiento del programa.

Existen muchas más características, pero consideramos que estas son las mas relevantes y las que hicieron a Java lo que hoy es.

## **Un poco de la historia de Python**

Python es un lenguaje de programación de alto nivel creado por Guido van Rossum, programador de computadoras holandés, conocido principalmente por ser el creador del lenguaje de programación Python. Nació el 31 de enero de 1956 en La Haya, Países Bajos. Van Rossum estudió matemáticas y ciencias de la computación en la Universidad de Ámsterdam.

Su desarrollo comenzó a finales de los años 80 y la primera versión pública, Python 0.9.0, se lanzó en febrero de 1991. El nombre del lenguaje fue inspirado por la afición de van Rossum por el grupo de comedia británico "Monty Python's Flying Circus".

En sus primeros años, Python se ganó una reputación por su sintaxis clara y legible, que enfatiza la legibilidad del código mediante el uso de sangrías y un enfoque en el uso de lenguaje natural. Aunque no era el lenguaje más rápido en términos de ejecución, su enfoque en la productividad y la facilidad de uso lo hicieron atractivo para desarrolladores de diferentes niveles de experiencia.

A medida que pasó el tiempo, Python fue ganando popularidad en la comunidad de desarrollo debido a su versatilidad y la amplia gama de bibliotecas y frameworks disponibles. Uno de los hitos importantes en la historia de Python fue el lanzamiento de Python 2.0 en 2000, que introdujo características como comprensiones de listas y Unicode. Sin embargo, con el tiempo, la comunidad comenzó a reconocer limitaciones y errores en el diseño de Python 2, lo que llevó a un esfuerzo para crear una versión más moderna y mejorada del lenguaje.

Este esfuerzo culminó en el lanzamiento de Python 3.0 en diciembre de 2008. Python 3 introdujo cambios significativos en la sintaxis y en las bibliotecas estándar, con un enfoque en la limpieza de elementos obsoletos y en la mejora de la consistencia del lenguaje. Sin embargo, la transición de Python 2 a Python 3 llevó tiempo debido a la incompatibilidad entre las versiones, y muchos desarrolladores continuaron utilizando Python 2 durante algún tiempo.

Con el tiempo, Python se convirtió en uno de los lenguajes de programación más populares y ampliamente utilizados en una variedad de campos, incluyendo desarrollo web, análisis de datos, inteligencia artificial, automatización, ciencia, educación y más. Su comunidad activa y su ecosistema de bibliotecas ricas contribuyeron significativamente a su adopción generalizada.

Python continúa creciendo en popularidad y se mantenía como uno de los lenguajes más influyentes y utilizados en la industria del desarrollo de software.

Cuando Python fue lanzado por primera vez en 1991, el contexto informático estaba marcado por varios elementos clave:

***Lenguajes de programación existentes:*** En ese momento, ya existían varios lenguajes de programación populares como C, C++, Java, Perl y otros. C y C++ eran ampliamente utilizados para programación de sistemas y aplicaciones de alto rendimiento, mientras que Perl era conocido por su capacidad para manipulación de texto y scripting.

***Énfasis en la legibilidad del código:*** Guido van Rossum, el creador de Python, tenía como objetivo principal crear un lenguaje que fuera fácil de leer y escribir. Esto se reflejó en la sintaxis de Python, que se caracteriza por el uso de sangrías en lugar de llaves para delimitar bloques de código, lo que fomenta un código más limpio y estructurado.

***Programación orientada a objetos:*** Python fue diseñado desde el principio para ser un lenguaje de programación orientado a objetos. Esto era coherente con una tendencia creciente en la industria hacia la programación orientada a objetos como paradigma de desarrollo.

***Enfoque en la productividad:*** Aunque Python no era el lenguaje más rápido en términos de ejecución, su enfoque en la productividad del desarrollador y en la facilidad de uso lo hacían atractivo para la creación rápida de prototipos y el desarrollo de aplicaciones.

***Internet emergente:*** A medida que Python se desarrollaba, internet también estaba en crecimiento. La posibilidad de escribir scripts para automatizar tareas en línea y desarrollar aplicaciones web contribuyó a la creciente popularidad de Python

***Bibliotecas y frameworks:*** A lo largo de los años, Python desarrolló una amplia gama de bibliotecas y frameworks para diversas aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje automático. Esto hizo que Python fuera aún más atractivo para una variedad de casos de uso.

Resumiendo, Python fue lanzado en un momento en el que había una variedad de lenguajes de programación disponibles, pero se destacó por su enfoque en la legibilidad, la programación orientada a objetos y la productividad del desarrollador. Estas características, junto con el crecimiento de Internet y la creación de bibliotecas y frameworks útiles, contribuyeron a la rápida adopción y evolución de Python en la comunidad de desarrollo.

## **Comparación de rendimiento (Benchmark).**

El análisis comparativo del rendimiento entre los distintos lenguajes de programación es esencial para entender cómo las diferentes tecnologías abordan los mismos problemas algorítmicos. La velocidad a la que se ejecuta un determinado programa puede verse muy influenciada por el número de núcleos presentes en el procesador en el cual se hace el benchmark, los algoritmos se prueban en diferentes tipos de hardware, desde máquinas con un solo core hasta máquinas octa-core. Los valores que se miden en cada prueba son: tiempo de ejecución, cantidad de memoria utilizada, y número de línea del código fuente del programa entre otros.

En este caso, realizamos una prueba de rendimiento entre Python y Java utilizando el algoritmo de ordenamiento por inserción, con una lista de diez mil valores, entre 0 y 10000 generados aleatoriamente.

El algoritmo de ordenamiento por inserción construye gradualmente una lista ordenada insertando elementos uno a uno en su posición adecuada en la parte ya ordenada. Aunque es un algoritmo simple, su rendimiento es inferior en comparación con otros algoritmos ya que su complejidad en la notación Big O es n^2, lo que lo hace menos eficiente para listas largas en comparación con otros algoritmos de ordenamiento.

Sobre Python: es un lenguaje interpretado que prioriza la sintaxis clara y la facilidad de desarrollo. Si bien es conocido por su simplicidad y flexibilidad, también puede tener una ejecución más lenta en comparación con lenguajes compilados.

Mientras que Java: es un lenguaje de programación compilado que enfatiza el rendimiento y la portabilidad. Aunque Java puede requerir más código para ciertas tareas en comparación con Python, su rendimiento suele ser muy superior.

Patrón de fondo

Descripción generada automáticamenteJava:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Python:

Patrón de fondo

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

**Resultados y Consideraciones:**

En esta prueba de rendimiento, observamos que Java mostró un rendimiento superior en términos de velocidad en comparación con Python. La eficiencia de Java se debe en gran medida a su compilador Just-In-Time. La Java virtual machine llama directamente al código compilado, como el código no se interpreta, la compilación no requiere tiempo de procesador ni uso de memoria.

En conclusión, el rendimiento de un algoritmo puede variar según el lenguaje de programación utilizado. Si bien Java demostró un rendimiento más rápido en esta prueba, Python tiene ventajas en términos de legibilidad y facilidad de desarrollo.

Anexo1: BNF Java

[Link a la BNF](https://cs.au.dk/~amoeller/RegAut/JavaBNF.html)

Anexo2: BNF Python

[Link a la BNF](https://docs.python.org/3/reference/grammar.html)