

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Escuela Profesional de Ingeniería de Estadística e Informática

Informe Técnico de Desarrollo sobre Tecnología: Lenguaje y Paradigma

Estudiante: Herson Romario Condori Mamani

Profesor: Fred Torres Cruz

13 de enero de 2025

Índice

1. Introducción	2
2. Lenguajes de Programación	2
2.1. Definición	2
2.2. Características Generales	2
2.3. Lenguajes Destacados	2
2.3.1. Java	2
2.3.2. Python	3
2.3.3. C++	3
3. Paradigmas de Programación	3
3.1. Definición	3
3.2. Principales Paradigmas	4
3.3. Paradigmas y Lenguajes	4
3.3.1. Orientado a Objetos	4
3.3.2. Multiparadigma	4
3.3.3. Funcional	4
4. Casos de Uso	4
4.1. Java	4
4.2. Python	4
4.3. C++	5
5. Conclusión	5
6. Código	5

1. Introducción

El desarrollo de software requiere la selección adecuada de tecnologías que se ajusten a las necesidades del proyecto. Este informe explora las características de tres lenguajes de programación (Java, Python y C++) y los paradigmas asociados a ellos, destacando sus ventajas, limitaciones y casos de uso más comunes.

2. Lenguajes de Programación

2.1. Definición

Un lenguaje de programación es un conjunto de reglas y símbolos que permiten a los desarrolladores comunicarse con las computadoras para crear aplicaciones y sistemas.

2.2. Características Generales

- **Sintaxis:** Conjunto de reglas que define la estructura del código.
- **Portabilidad:** Capacidad de ejecutar código en diferentes plataformas sin modificaciones significativas.
- **Soporte de librerías:** Disponibilidad de componentes reutilizables para acelerar el desarrollo.
- **Comunidad:** Cantidad de desarrolladores que contribuyen al ecosistema del lenguaje.

2.3. Lenguajes Destacados

2.3.1. Java

Java es un lenguaje de programación orientado a objetos creado por Sun Microsystems en 1995. Es ampliamente utilizado por su robustez, seguridad y portabilidad.

Ventajas de Java

- **Independencia de Plataforma:** La máquina virtual de Java (JVM) permite ejecutar el mismo código en diferentes sistemas operativos.
- **Seguridad:** Su modelo de ejecución restringe operaciones potencialmente peligrosas.
- **Comunidad Activa:** Amplia documentación y soporte.

Limitaciones de Java

- **Rendimiento:** Puede ser inferior al de lenguajes compilados directamente a máquina.
- **Curva de Aprendizaje:** La sintaxis y el entorno pueden ser complejos para principiantes.

2.3.2. Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, creado por Guido van Rossum en 1991. Es conocido por su simplicidad y legibilidad.

Ventajas de Python

- **Sintaxis Clara:** Facilita el aprendizaje y el desarrollo rápido.
- **Versatilidad:** Usado en desarrollo web, ciencia de datos, inteligencia artificial y más.
- **Gran Ecosistema:** Amplia colección de librerías y frameworks.

Limitaciones de Python

- **Rendimiento:** Es más lento que los lenguajes compilados.
- **Consumo de Memoria:** Puede ser mayor debido a su naturaleza interpretada.

2.3.3. C++

C++ es un lenguaje de programación de alto rendimiento creado por Bjarne Stroustrup en 1983. Combina características de programación de bajo nivel y alto nivel.

Ventajas de C++

- **Alto Rendimiento:** Adecuado para sistemas y aplicaciones críticos en tiempo real.
- **Control Fino:** Permite gestión manual de memoria y optimización avanzada.
- **Compatibilidad:** Puede integrarse con código en C y otras tecnologías.

Limitaciones de C++

- **Complejidad:** Su sintaxis y manejo de memoria pueden ser desafiantes.
- **Errores Difíciles de Depurar:** Problemas como fugas de memoria pueden ser comunes.

3. Paradigmas de Programación

3.1. Definición

Un paradigma de programación es un estilo o enfoque para resolver problemas computacionales. Define la forma en que los desarrolladores estructuran y organizan el código.

3.2. Principales Paradigmas

- **Imperativo:** Enfocado en cómo lograr los resultados mediante instrucciones secuenciales.
- **Declarativo:** Enfocado en qué se quiere lograr, dejando el cómo al lenguaje o framework.
- **Orientado a Objetos:** Basado en el concepto de "objetos" que representan entidades del mundo real.
- **Funcional:** Se centra en el uso de funciones puras y evita estados mutables.

3.3. Paradigmas y Lenguajes

3.3.1. Orientado a Objetos

- **Java:** Ejemplo clásico del paradigma orientado a objetos, con soporte robusto para herencia, encapsulación y polimorfismo.
- **C++:** Combina orientación a objetos y capacidades de bajo nivel, lo que lo hace ideal para sistemas complejos.

3.3.2. Multiparadigma

- **Python:** Soporta múltiples paradigmas, incluyendo imperativo, funcional y orientado a objetos, adaptándose a diversos estilos de programación.

3.3.3. Funcional

Aunque no es exclusivo, Python y C++ también permiten estilos funcionales, mientras que Java lo soporta parcialmente mediante expresiones lambda y streams.

4. Casos de Uso

4.1. Java

- **Aplicaciones Empresariales:** Gracias a su robustez y soporte para grandes sistemas.
- **Desarrollo Móvil:** Base del desarrollo de aplicaciones Android.
- **Aplicaciones Web:** A través de frameworks como Spring y Hibernate.

4.2. Python

- **Ciencia de Datos:** Usado ampliamente en análisis de datos y aprendizaje automático.
- **Automatización:** Ideal para tareas repetitivas y scripts.
- **Desarrollo Web:** Frameworks como Django y Flask.

4.3. C++

- **Sistemas Embebidos:** Por su rendimiento y control de hardware.
- **Videojuegos:** Motor Unreal Engine utiliza C++ como base.
- **Aplicaciones de Alto Rendimiento:** Sistemas financieros y simulaciones.

5. Conclusión

Los lenguajes Java, Python y C++ y sus paradigmas asociados ofrecen herramientas poderosas para el desarrollo de software. Cada uno tiene fortalezas únicas que lo hacen adecuado para diferentes tipos de proyectos. La selección del lenguaje y paradigma debe alinearse con los requisitos y restricciones del proyecto para maximizar su efectividad.

6. Código

<https://github.com/romariomamani/graficador-de-funciones>