



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

UNIVERSIDAD ESTATAL AMAZÓNICA



TEMA: Viaje a través de la Evolución de los Lenguajes de Programación

NOMBRES Y APELLIDOS: Carlos Daniel Castillo Cabrera; Delgado Mala

Marcos Vinicio

ASIGNATURA: 2525 - PROGRAMACION ORIENTADA A OBJETOS

CODIGO: UEA-L-UFB-030-A

PARALELO: (A)

Lago Agrio, 08 de junio 2025

 www.uea.edu.ec

 Km. 2. 1/2 vía Puyo a Tena (Paso Lateral)

 032892-118 / 032892-188 032892-098 / 032896-188 032896-476

#UEAesExcelencia

Carlos Daniel Castillo Cabrera

Viaje a través de la Evolución de los Lenguajes de Programación

Informe de Investigación

1. Introducción

Los lenguajes de programación han evolucionado significativamente desde sus inicios, adaptándose a las necesidades tecnológicas y mejorando la eficiencia en el desarrollo de software. Este informe explora la evolución de tres lenguajes clave: **C++, Java y Python**, analizando su origen, hitos importantes y su impacto en la programación moderna, especialmente en la **Programación Orientada a Objetos (POO)**.

2. Lenguajes Seleccionados y su Evolución

C++

Origen (1979-1985)

- Creado por **Bjarne Stroustrup** como una extensión de **C** para añadir programación orientada a objetos.
- Nombre original: "**C with Classes**".
- Primer compilador en **1985**.



Hitos Clave

- **1998**: Estandarización ISO (C++98).
- **2011**: C++11 introdujo auto, lambdas y multihilo.
- **2020**: C++20 con módulos y corrutinas.

Ejemplo de Código

```
#include <iostream>
using namespace std;

class Animal {
public:
    void speak() {
        cout << "Animal sound!" << endl;
    }
};

int main() {
    Animal dog;
    dog.speak();
    return 0;
}
```

Impacto en POO: Popularizó clases, herencia y polimorfismo.

Java

Origen (1991-1995)

- Creado por **James Gosling** (Sun Microsystems) para dispositivos embebidos.
- Lema: "**Write Once, Run Anywhere**" (gracias a la JVM).



Hitos Clave

- **1996:** JDK 1.0 (primer lanzamiento).
- **2004:** Java 5 (genéricos, autoboxing).
- **2014:** Java 8 (lambdas, Stream API).

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, Java!");  
    }  
}
```

Impacto en POO: Enfoque en objetos puros, herencia simple e interfaces.

Python

Origen (1991)

- Creado por **Guido van Rossum** como un lenguaje fácil de leer.
- Filosofía: "**Legibilidad cuenta**" (Zen de Python).



Hitos Clave

- **2000:** Python 2.0 (list comprehensions).
- **2008:** Python 3.0 (ruptura de compatibilidad).
- **2020:** Python 3.9 (diccionarios merge).

Ejemplo de Código

```
class Dog:  
    def __init__(self, name):  
        self.name = name  
  
    def bark(self):  
        print(f"{self.name} says Woof!")  
  
my_dog = Dog("Rex")  
my_dog.bark()
```

Impacto en POO: Sintaxis clara, multiparadigma (POO + funcional).

3. Comparación y Contraste

Característica	C++	Java	Python
Paradigma	Multiparadigma	Orientado a Objetos	Multiparadigma
Tipado	Estático	Estático	Dinámico
Rendimiento	Alto	Medio-Alto	Medio-Bajo
Uso Principal	Sistemas, Juegos	Empresas, Android	IA, Web, Scripting

Influencia en POO:

- C++ introdujo POO en lenguajes de bajo nivel.
- Java estandarizó POO en entornos empresariales.
- Python hizo POO accesible para principiantes.

Reflexión Personal

Aprendizajes clave (Carlos Castillo):

1. La evolución de los lenguajes refleja las necesidades de la industria (velocidad, legibilidad, portabilidad).
2. La **POO** se ha adaptado a diferentes contextos (desde sistemas embebidos hasta inteligencia artificial).



Impacto en mi enfoque hacia POO:

- Ahora valoro más la legibilidad (Python) y la estructura (Java).
- Entiendo que C++ es poderoso pero complejo, ideal para optimización.
- La POO no es "un estándar único", sino un conjunto de principios flexibles.

Persona 2 (Marcos Delgado)

POO como herramienta flexible:

- No hay una "POO perfecta": Java es estricto, Python es flexible y C++ permite bajo nivel.
- La elección del lenguaje depende del proyecto (ej: Android → Java, IA → Python).

Legibilidad y comunidad:

- **Python** demuestra que un código fácil de mantener es tan importante como su funcionalidad.
- El soporte comunitario (librerías, frameworks) define el éxito de un lenguaje hoy.

Impacto en el aprendizaje:

- Entender la historia ayuda a evitar "reinventar la rueda" (ej: punteros en C++ ya tienen alternativas en Java/Python).
- La POO se aprende mejor empezando con lenguajes como Python antes de pasar a Java o C++.



5. Presentación Breve

Puntos clave para compartir:

- **C++** revolucionó la POO en sistemas de bajo nivel.
- **Java** democratizó la portabilidad con la JVM.
- **Python** hizo la programación accesible y escalable.

Conclusión

La evolución de los lenguajes de programación ha sido impulsada por la necesidad de mejorar la productividad, legibilidad y eficiencia. **C++**, **Java** y **Python** representan hitos clave en la adopción de la POO, cada uno con enfoques distintos pero complementarios.

Bibliografía

Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley Professional.

Bloch, J. (2017). *Effective Java* (3rd ed.). Addison-Wesley Professional.

Van Rossum, G., & Drake, F. L. (2023). *Python 3.11 documentation*. Python Software Foundation.

<https://docs.python.org/3/>

International Organization for Standardization. (2020). *ISO/IEC 14882:2020: Programming languages — C++*.

<https://www.iso.org/standard/79358.html>

Oracle. (2023). *The Java® language specification*.

<https://docs.oracle.com/javase/specs/>

Stroustrup, B. (2014, September). *The essence of C++* [Conference presentation]. CppCon 2014, Bellevue, WA, United States.

<https://youtu.be/86xWVb4XIyE>



UEA
UNIVERSIDAD
ESTATAL AMAZÓNICA

ACM SIGPLAN. (2020). *History of programming languages (HOPL-IV)*.
<https://hopl4.sigplan.org/>

✉ www.uea.edu.ec

📍 Km. 2. 1/2 vía Puyo a Tena (Paso Lateral)

📞 032892-118 / 032892-188 032892-098 / 032896-188 032896-476

#UEAesExcelencia
Carlos Daniel Castillo Cabrera