

M9 Desarrollo de aplicaciones orientadas a objetos FIN A

Actividad 2

|  |  |
| --- | --- |
| **Tutor:** | **Karla Ramos García** |
| **Estudiante:** | **José Ramón Ibáñez Posadas** |
| **Matricula:** | **BNL098377** |

|  |  |
| --- | --- |
| Monterrey, Nuevo León | viernes, 19 de septiembre de 2025 |

Introducción

La programación orientada a objetos (POO) es un paradigma de programación que ha transformado la manera en que se crean las aplicaciones. Desde sus orígenes, ha evolucionado para convertirse en una base fundamental para el desarrollo de software moderno.

Este trabajo explorará la evolución histórica de la POO a través de una línea de tiempo, analizando cómo sus principios han sido adoptados y perfeccionados en diversos lenguajes de programación como Java, C#, Eiffel, Smalltalk y los lenguajes de scripting.

En las siguientes secciones, se examinarán las características distintivas de estos lenguajes, así como sus aplicaciones prácticas, lo que permitirá comprender mejor el impacto de la POO en el desarrollo de software y cómo sigue siendo relevante en la actualidad.

Desarrollo

**Línea de tiempo**

**A line of colorful dots

AI-generated content may be incorrect.**

<https://lucid.app/lucidspark/08a04681-d576-4b29-9886-4dd57b05c7e7/edit?viewport_loc=2326%2C1891%2C2622%2C1543%2C0_0&invitationId=inv_3d67c95a-5ff1-4f4e-b596-b5a951e06e95>

**Cuadro sinóptico**

|  |  |  |  |
| --- | --- | --- | --- |
| Lenguaje | Definición | Características | Ejemplo de Aplicación |
| Java | Un lenguaje de programación basado en clases, creado por James Gosling en 1995 como un componente de la plataforma Java de Sun Microsystems. | - Se puede ejecutar en cualquier máquina que tenga una Máquina Virtual de Java (JVM), por lo que la arquitectura no es importante . | - Es un lenguaje basado en clases, simple y de propósito general . |
| C# | Un lenguaje orientado a objetos desarrollado y estandarizado por Microsoft en el año 2001. | - Su sintaxis es expresiva, sencilla y fácil de aprender, especialmente si ya estás familiarizado con C y C++ . | - Permite la herencia, el polimorfismo y el encapsulamiento . |
| Eiffel | Un lenguaje de programación orientado a objetos | puro basado en el estándar ISO y diseñado por Bertrand Meyer. | - Enfocado en el diseño por contrato, lo que facilita la programación orientada a objetos . |
| Smalltalk | Un lenguaje de programación orientado a objetos | puro, creado por Alan Kay para brindar un ambiente de experimentación y creatividad a sus usuarios. | - Se distingue por ser de tipado dinámico y reflexivo, además de contar con un recolector de basura . |
| Lenguajes de Scripting | Un tipo de lenguaje de programación que es interpretado, lo que significa que el código fuente se interpreta en el momento de la ejecución, comando por comando. | - Son más fáciles de escribir y se implementan a través de intérpretes . | - El código es más corto que en los lenguajes compilados . |

Conclusión

El estudio de la Programación Orientada a Objetos me ha permitido comprender que este paradigma no es simplemente una forma de escribir código, sino una manera de modelar el mundo real en el software. La evolución de la POO refleja la búsqueda constante de los desarrolladores por crear sistemas más organizados, modulares y fáciles de mantener. Lenguajes como C++ sentaron las bases al ser híbridos, mientras que otros como Smalltalk y Eiffel demostraron el potencial de un enfoque puramente orientado a objetos.

Personalmente, me parece fascinante cómo la programación ha avanzado de la vinculación estática, donde la función a ejecutar se conocía de antemano, a la vinculación dinámica, que permite que la decisión se tome en tiempo de ejecución, haciendo los programas más flexibles y adaptables. Esto es un pilar del polimorfismo y se puede ver en ejemplos tan sencillos como el cálculo del sueldo de un empleado por comisión vs. un empleado de base. Me parece increíble cómo este concepto, que es fundamental en el software, tiene paralelos en la vida real, como en el ejemplo de una puerta, donde el método "Dibujar" cambia dependiendo del tipo de puerta.

Además, la diferencia entre lenguajes compilados y lenguajes de scripting es una distinción crucial que he aprendido. A pesar de que los lenguajes de scripting como JavaScript y PHP se ejecutan de manera diferente y a menudo son más sencillos de escribir, su impacto en la creación de páginas web dinámicas es innegable. La flexibilidad de estos lenguajes para interactuar con los usuarios y los servidores ha sido clave en la transformación del internet, algo que utilizo a diario sin darme cuenta de la complejidad subyacente. En resumen, la POO es una herramienta poderosa que no solo se refleja en el código, sino que también nos ayuda a entender cómo la lógica y la estructura se aplican en la tecnología que nos rodea.

**Bibliografía**

**A screenshot of a computer

AI-generated content may be incorrect.**