**Evolución de los lenguajes de programación**

**(ensayo descriptivo)**

Nombre: Oscar Eduardo López Vargas

Materia: Técnicas de comunicación escrita

Fecha: 28/06/25

Docente: Juan Eduardo Sejas Castillo

Los lenguajes de programación han sufrido una gran evolución en las últimas décadas, evolucionando a un ritmo que pocos podrían haber imaginado. Cada vez que abrimos una aplicación o navegamos por internet, estamos presenciando el resultado de millones de instrucciones diseñadas para tender puentes entre el pensamiento humano y la comprensión de las máquinas. Es por esta razón los lenguajes de programación se volvieron la estructura de la tecnología.

Este ensayo buscamos conocer toda la historia de la evolución, explorando cómo hemos pasado de comunicarnos con máquinas mediante códigos casi incomprensibles hasta herramientas tan intuitivas que prácticamente nos permiten mantener conversaciones naturales con la tecnología. Recorreremos juntos las etapas más importantes de esta evolución, descubriendo cómo cada avance ha acercado más la tecnología a la forma natural en que los humanos pensamos y nos comunicamos.

Los lenguajes de programación tuvieron una gran evolución, gracias a que esta evolución fue progresiva y constante.

En las primeras etapas se programaba en un lenguaje denominado lenguaje máquina, en este se usaba el código binario es decir se utilizaba solo unos y ceros y la maquina podía interpretar estos dos números para poder entender lo que queríamos que haga, esta fue una de las formas más difícil de programar por lo cual se creó el lenguaje ensamblador, este remplazaba los numero binarios por símbolos, haciendo que todo el proceso de programar mucho más sencillo.

En el año 1950, aparecieron nuevos lenguajes que eran de alto nivel, estos permitían escribir instrucciones muy parecidos al nuestro lenguaje. El primero fue Fortran se creo en el año 1957 y estaba dirigido a las matemáticas y la ciencia, luego apareció Cobol que se utilizaba en aplicaciones administrativas y comerciales. Estos dos lenguajes marcaron un antes y un después en el mundo de los lenguajes de la programación, ya que introdujeron compiladores.

Durante los años 60 y 70, se consolidó el paradigma de la programación estructurada, este paradigma impuso orden y claridad en el diseño del software, muchos lenguajes como c y pascal dejaban que organicemos los códigos en bloques lógicos, creando así estructuras como condicionales, bucles y funciones, estas estructuras fueron muy funcionales y gracias a esto los códigos se entendían mejor

Lenguajes como Pascal y C permitieron organizar el código en bloques lógicos, empleando estructuras como condicionales, bucles y funciones, lo que mejoró significativamente la legibilidad y el mantenimiento del código.

En los años 80 y 90, la programación tuvo un giro de 180 grados y se empezó a orientar hacia objetos esto transformó la forma en la que se diseñaba el software, es considerada por muchos como un paradigma muy poderoso y eficiente gracias a la capacidad que tiene para mostrar sistemas complejos de manera más natural y organizada a través de conceptos nuevos como objetos, clases, herencia, encapsulamiento y polimorfismo, esta nueva forma de programación permitió una mejor reutilización del código, mayor mantenibilidad y una representación más intuitiva de los problemas del mundo real en el código lo que facilitó el trabajo colaborativo y la construcción de aplicaciones complejas. Entre los lenguajes más utilizados están Smalltalk, C++ y Java.

La llegada de Internet cambió todo el panorama de la programación de una manera que nadie había anticipado. HTML surgió como el organizador perfecto, estructurando toda la información para que cualquier navegador pudiera entenderla, mientras que JavaScript dotó a las páginas de vida e interactividad real. Lenguajes como PHP, Ruby y Python se convirtieron en los trabajadores incansables detrás de escena, siendo Python el que logró algo extraordinario: ser poderoso sin intimidar. Su sintaxis clara y versatilidad lo convirtieron en el favorito no solo para desarrollo web, sino también para aventurarse en territorios completamente nuevos como la inteligencia artificial y el análisis de datos.

La explosión de los dispositivos móviles y la sofisticación creciente de la web despertaron una necesidad urgente: herramientas más especializadas que pudieran enfrentar estos nuevos desafíos. Así nació una generación de lenguajes diseñados para resolver problemas específicos del mundo digital moderno.

Microsoft respondió en 2001 con C#, creando un puente perfecto entre la potencia de la programación tradicional y la simplicidad que demandaban los desarrolladores de Windows. Era exactamente lo que necesitaba el ecosistema .NET para florecer. Google, siempre pensando en grande, lanzó Go en 2009 como la respuesta ideal para la era de la computación en la nube: ligero, concurrente y capaz de manejar miles de conexiones sin inmutarse.

Rust llegó en 2010 con una propuesta que sonaba casi imposible: ofrecer la velocidad de los lenguajes más rápidos sin los temidos errores de memoria que atormentaban a los programadores desde hacía décadas. Era como prometer un auto de carreras completamente seguro. Apple, mientras tanto, trabajaba en su propia revolución con Swift, un lenguaje que pretendía hacer olvidar las complejidades de Objective-C y convertir el desarrollo de aplicaciones iOS y macOS en una experiencia más placentera y accesible.

Kotlin apareció en 2011 como la solución elegante que muchos desarrolladores Java habían estado esperando sin saberlo. Este es uno de los lenguajes de programación de moda, no porque sea el más sencillo, sino que Google lo ha señalado como el lenguaje de programación óptimo para Android., fue como recibir el sello de aprobación definitivo. Su combinación de concisión, seguridad y perfecta compatibilidad con Java lo convirtió en el favorito indiscutible para el desarrollo móvil.

Además, se han desarrollado tecnologías nuevas de programación visual y entornos no-code, que permiten crear aplicaciones sin necesidad de escribir código, democratizando así el acceso al desarrollo de software.

Por otro lado, el avance de la inteligencia artificial ha cambiado completamente la forma en la que sabíamos programar ya que ahora es mucho más sencillo y solo tenemos que dar órdenes a herramientas conocidas como copilotos de código, entre ellas GitHub Copilot o asistentes como ChatGPT gracias a todo esto los desarrolladores pueden obtener asesoramiento gratuito e instantáneo, código automático de relleno, corregir errores y crear secciones de código completas a partir de comandos básicos descritos en el lenguaje cotidiano. Este tipo de asistencia inteligente ha simplificado notablemente el proceso de programación, reduciendo las barreras técnicas para los principiantes y optimizando el flujo de trabajo de los profesionales. Ya no es necesario memorizar una gran cantidad de sintaxis o consultar manuales constantemente ahora solo se necesita expresar una intención clara, y la inteligencia artificial puede traducirla en instrucciones computacionales funcionales.

Gracias a todo lo investigado podemos aclarar que los lenguajes de programación han dejado de ser experiencia exclusiva y se han convertido en herramientas fáciles de usar, disponibles para personas de diversos campos como arte, enseñanza, atención médica o comercio La programación ya no es solo un trabajo técnico, sino que se ve como un concurso amplio del siglo XXI, esencial para la innovación, la resolución de problemas y contribuye activamente a la creación del mundo digital

Al contemplar esta evolución de los lenguajes de programación, resulta inevitable sentir cierta nostalgia por esos primeros programadores que pasaban horas enteras escribiendo líneas interminables de unos y ceros. Es fácil imaginar su frustración cuando un simple error los obligaba a revisar miles de líneas de código binario, pero también su inmensa satisfacción cuando finalmente lograban que la máquina hiciera exactamente lo que buscaban.

Es realmente fascinante pensar que hemos transitado de esa época donde programar era casi como descifrar jeroglíficos antiguos, hasta nuestros días donde cualquier persona puede conversar con una inteligencia artificial diciendo "necesito una aplicación que me ayude a organizar mis gastos" y recibir ayuda real para crearla. Una abuela que apenas maneja WhatsApp ahora podría, teóricamente, desarrollar su propia aplicación para catalogar sus recetas familiares.

Lo más emocionante de todo este recorrido es cómo la programación se ha vuelto genuinamente humana, accesible y cercana. Ya no se requiere ser un genio matemático ni memorizar cientos de comandos extraños. Ahora es posible sentarse frente a una computadora y mantener una conversación casi natural para crear algo extraordinario. Es como si las máquinas hubieran aprendido nuestro idioma en lugar de forzarnos a dominar el suyo.

Estamos claramente en el umbral de algo verdaderamente extraordinario. Muy pronto, crear software será tan natural como escribir una carta a un amigo o esbozar un dibujo en una servilleta. Esto significa que todas esas ideas brillantes que las personas guardan en su mente podrán finalmente materializarse en forma digital, sin importar si han estudiado ingeniería o arte, medicina o literatura. Esa transformación representa la verdadera revolución de nuestro tiempo.

**Bibliografía**

Epitech (2022), Evolución de los lenguajes de programación: Inicio y actualidad

<https://www.epitech-it.es/evolucion-lenguajes-de-programacion/>

UNAM (2004), Lenguajes de Programación

<https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2655/mod_resource/content/1/UAPA-Lenguajes-Programacion/index.html>

Anthony Urquiza, Evolución de la programación

<https://www.scribd.com/document/650460949/EVOLUCION-DE-LA-PROGRAMACION>

Vicente Trigo Aranda, Historia y evolución de los lenguajes de programación

<https://www.acta.es/medios/articulos/informatica_y_computacion/034083.pdf>