



**Ministerio de Educación**

**Universidad tecnológica de Panamá**

**Estudios Generales**

**Salón**

**2EG7116**

**Materia**

**Desarrollo lógico y algorítmico**

**Profesor**

**Napoleón Ibarra**

**Estudiante**

**Olmedo Alvarado**

**Tarea #1**

**Historia de la programación**

**Fecha de entrega:**

**25/8/2025**

## Historia de la programación

Programar es darle instrucciones a una computadora para que ejecute una orden. Hoy tenemos herramientas como ChatGPT que pueden generar códigos, pero llegar hasta ese punto tomó siglos de ideas, experimentos y aportes de muchas personas que fueron moldeando lo que hoy llamamos “programación”.

Empiezo con una pregunta clásica:

### **¿Qué vino antes, el huevo o la gallina?**

Mi respuesta es la gallina. Nació de un huevo, pero no de una gallina como tal, sino de otro animal que con el tiempo y la evolución fue mutando hasta lo que conocemos hoy.

Alvarado, ¿Y qué tiene que ver esto con la programación?

Pasa algo parecido. Antes de tener computadoras y lenguajes de programación como los conocemos, hubo muchos pasos intermedios que nos llevaron hasta aquí.

Ya en la antigüedad se usaban secuencias de pasos para resolver problemas. El algoritmo de Euclides (siglo III a.C.) servía para calcular el máximo común divisor, y en Babilonia (aprox. 1800 a.C.) se aplicaban procedimientos repetitivos para aproximar raíces, una idea muy parecida a lo que hoy llamamos iteraciones. Más adelante, en 1614, John Napier publicó sus tablas de logaritmos que facilitaban los cálculos, y en 1620 apareció la regla de cálculo, que siguió usándose hasta el siglo XX. No eran “programas”, pero sí mostraban cómo resolver problemas siguiendo pasos definidos.

En 1679, Leibniz presentó el sistema binario (0 y 1) y en 1694 construyó una calculadora que hacía operaciones básicas automáticamente. Estos dos aportes resultaron claves: el lenguaje numérico perfecto para las máquinas y la idea de automatizar cálculos.

Después, en 1801, Joseph Marie Jacquard inventó un telar que funcionaba con tarjetas perforadas que dictaban el patrón del tejido. Esa idea inspiró a Charles Babbage, que entre 1833 y 1842 diseñó la Máquina Analítica, considerada el primer modelo de computadora programable. Ada Lovelace fue la primera en

escribir algoritmos pensados para esa máquina y en darse cuenta de que podía servir no solo para cálculos, sino para manipular símbolos o incluso crear música.

En 1854, George Boole publicó el álgebra booleana, que expresaba el razonamiento lógico en términos de verdadero o falso. Años después, en 1937, Claude Shannon probó que esa misma lógica podía aplicarse en circuitos eléctricos, lo que conectó las matemáticas con la ingeniería y sentó las bases del hardware moderno.

En 1936, Alan Turing describió la Máquina de Turing, un modelo que podía representar cualquier cálculo posible. Junto con los trabajos de Alonzo Church, surgió la tesis de Church–Turing, que hasta hoy define los límites de lo que una máquina puede calcular.

La Segunda Guerra Mundial aceleró las cosas. En 1941 Konrad Zuse construyó la Z3, en Inglaterra se creó Colossus (1943) y en EE.UU. apareció el ENIAC (1946). Al principio se programaban con cables y paneles, pero en 1949 llegó la idea del programa almacenado en memoria (Manchester Baby, EDSAC), lo que permitió cargar y modificar código como datos.

En los años cincuenta nacieron los primeros lenguajes. Del lenguaje máquina se pasó al ensamblador, y luego llegaron FORTRAN (1957), COBOL (1959), LISP (1958) y ALGOL, que introdujo conceptos como bloques y recursión.

En los sesenta y setenta, la programación se diversificó: Pascal (1970) ayudó a estructurar mejor el código, C (1972) fue clave junto con UNIX, y aparecieron otros paradigmas con lenguajes como Prolog, Smalltalk y Scheme.

Los ochenta y noventa consolidaron la orientación a objetos con C++ y Java, y lenguajes como Python, MATLAB, SQL, Perl, PHP y JavaScript marcaron la ciencia, las bases de datos y el desarrollo web.

Ya en los 2000 y 2010 surgieron C#, Scala, Go, Rust, Swift, Kotlin y TypeScript. Python y R se volvieron protagonistas en ciencia de datos e inteligencia artificial. Al mismo tiempo, la nube, las GPUs y los microservicios cambiaron por completo la forma de construir software.

Hoy la programación está en todo: desde un microcontrolador en un electrodoméstico hasta servicios globales en la nube. Lenguajes como Rust, Go, Python y TypeScript conviven con herramientas como Git y con la inteligencia artificial, que incluso puede escribir código. Sin embargo, los fundamentos no han cambiado: seguimos trabajando con algoritmos, binario y lógica booleana.

¿Cuál es su concepto y características?

Programar es diseñar instrucciones claras y ordenadas para que una computadora realice tareas o resuelva problemas. Estas instrucciones pueden repetirse, tomar decisiones según condiciones y combinarse en estructuras más grandes.

¿Importancia o relevancia?

La programación permite automatizar tareas, resolver problemas complejos y es la base de toda la tecnología digital actual. Desde un celular hasta un sistema de inteligencia artificial, todo funciona gracias a programas escritos por alguien.

II. Parte

[https://www.canva.com/design/DAGxF4oSc7o/70CUhFbkpvfAXpEOelBZzg/edit?utm\\_content=DAGxF4oSc7o&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGxF4oSc7o/70CUhFbkpvfAXpEOelBZzg/edit?utm_content=DAGxF4oSc7o&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)