Historia de la Programación: De la Arcilla a los Bucles
Autor: Kimi (Kimi-ai)
Publicación: 18 de septiembre de 2025

Introducción

La programación no nació con los microprocesadores: es el arte de organizar pasos para obtener un resultado deseado. Este ensayo recorre seis hitos que muestran cómo el ser humano ha *codificado* instrucciones desde el barro hasta los bucles, pasando por engranes, telares y ábacos.

---

a) El genoma humano: un programa que se ejecuta en carbono

El ADN contiene ≈3.2×10^9 pares de bases que codifican ≈20 000 "sub-rutinas" (genes). Su estructura es *secuencial, condicional y repetitiva*:

- **Secuencial**: promotor → gen → terminador.
- **Condicional**: operadores lógicos (AND/OR) implementados por proteínas reguladoras.
- **Repetitiva**: bucles *for* (repeticiones en tándem) y *while* (alargamiento de telómeros).

Como cualquier software, el genoma acumula *patches* (mutaciones) y *forks* (recombinación). La diferencia: la *CPU* es la química celular y la *memoria* es carbono, no silicio.

---

b) Tablillas de arcilla: trigonometría en base 60 y memoria humana

Hacia 1800 a.C. los babilonios grababan tablillas con *sexagesimal* (base 60). La famosa **Plimpton 322** contiene 15 filas de números que hoy reconocemos como **triplas pitagóricas**:

```
a² + b² = c² (con ángulos 30°, 45°, 60°)
```

Cada tablilla era un *array* de 3 columnas; el *sistema operativo* era el escriba, que **memorizaba algoritmos de multiplicación y división** en base 60 sin papel auxiliar.
La **memoria humana** actuaba como *RAM*: cargaba valores, operaba y volvía a grabar en arcilla. La trigonometría nació como *código legible* para agrimensores y astrónomos.

---

c) El ábaco: memoria caché en carne y hueso

El ábaco griego (300 a.C.) convierte números en *patrones físicos*. El usuario **almacena dígitos en los dedos** (caché L1) y **resultados parciales en la mente** (caché L2).
Algoritmo típico:

1. Cargar operando en hilos.
2. Sumar/Restar con *carry* visual.
3. Escribir resultado o pasar a siguiente operación.

Es **programación sin electricidad**: el *bus* son los movimientos de cuentas; el *clock*, el ritmo del usuario.

---

d) La Anticitera: engranes como *if-else* mecánicos

Hacia 100 a.C. los griegos construyen el **mecanismo de Anticitera**: 37 engranes dentados que calculan:

- Fases lunares
- Eclipses
- Posición de planetas

Cada engrane es un *case* de un *switch*: si el eje gira X grados, el engrane Y desplaza el puntero Z.
La **disposición de dientes** codifica *constantes* (relaciones de transmisión) y *condicionales* (acoplamientos que solo activan en ciertas posiciones).
Es el **programa más antiguo que se ejecuta sin parar** mientras el usuario gira la manivela: un *loop infinito* mecánico.

---

e) Jacquard (1801): el telar que lee *punch-cards*

Joseph Marie Jacquard convierte las **tarjetas perforadas** en *instrucciones* para telares. Cada fila de orificios es un *opcode*:

- 1 = levantar hilo
- 0 = dejar bajo

El telar **interpreta** la tarjeta secuencialmente → patrón complejo en la tela.
Impacto: **primer lenguaje de alto nivel** (el diseñador piensa en patrones, no en levas) y **primer intérprete** mecánico.

---

f) Z3 (1941): bucles almacenados

Konrad Zuse completa la **Z3**: relés que ejecutan **8 instrucciones** y **1 bucle**.
El programa reside en **cinta perforada** (stored-program) y puede **saltar hacia atrás** → *loop* real.
Ejemplo: multiplicación por sumas repetidas:
复制

```
LOAD  R1, multiplicando
LOAD  R2, multiplicador
LOOP: ADD R3, R1
      DEC R2
      JNZ LOOP   ← salta si R2 ≠ 0
```

Es el **punto de inflexión**: por primera vez el *hardware* no cambia; solo cambia la **cinta**, es decir, el *software*.

Conclusión

Desde el genoma hasta los relés, la programación ha sido **codificar pasos en un medio que los ejecute**. Arcilla, cuentas, engranes, telares o relés: todos obedecen la misma lógica que hoy corremos en Python: **secuencia, selección y repetición**.
La única diferencia es la velocidad; la esencia, la misma.

Referencias

(1) DevCamp, *Historia de la programación*, 2024.
(2) Lahistoria.info, *Orígenes y evolución*, 2024.
(3) Luis Llamas, *Breve historia*, 2023.
(4) CirculosCreativos, *Cronología*, 2024.
(5) MPRU, *Introducción a la programación*, 2024.