



# Programación I

---

## Introducción a la programación

# Programa

---

Resultados de aprendizaje: Soluciona problemas básicos de programación elaborando e implementando algoritmos de manera autónoma, utilizando lenguajes de programación para el procesamiento básico de datos

Evaluaciones:

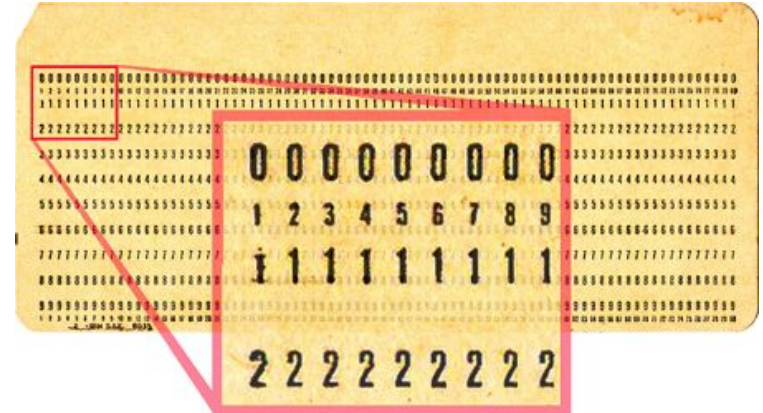
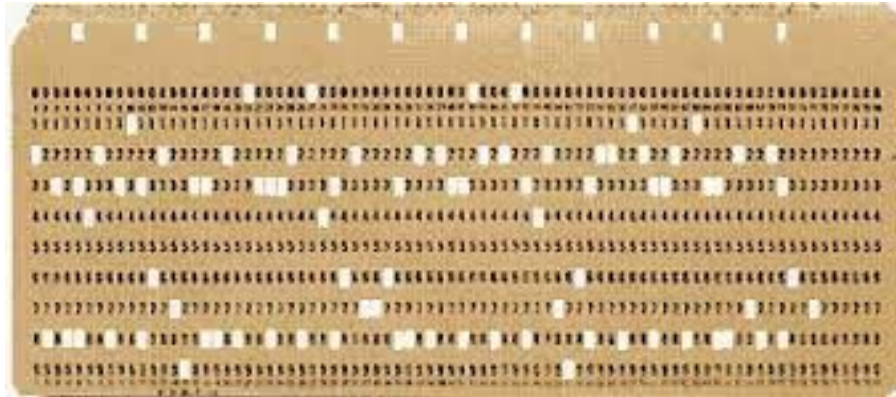
Tipo	Porcentaje	Fecha(aproximada)
Evaluación 1	20% Nota Final	11-15/04
Evaluación 2	30% Nota Final	16-20/05
Evaluación 3	30% Nota Final	20-24/06
Evaluación Continua (Tareas, Investigaciones y Talleres)	20% Nota Final	

# ¿Qué es la programación?

Programación es un **proceso** donde se realiza una **planeación** y **creación** de algún programa, llevando a cabo una **resolución de problemas**, a través de técnicas de **análisis**, **definición** de los requerimientos del problema y **diseño**. Si esto es llevado a la práctica es la realización de un algoritmo para resolver un problema.

# Tarjeta perforadas

La tarjeta perforada es una lámina hecha de cartulina que contiene información en forma de perforaciones según un código binario. Fue uno de los primeros medios que se utilizaron para ingresar información e instrucciones a una computadora en los



# Código del apolo 11

Margaret Hamilton, la científica computacional, matemática e ingeniería estadounidense que hizo posible el alunizaje en 1969 por haber diseñado el programa informático que controló el funcionamiento de todos los sistemas durante la misión espacial.

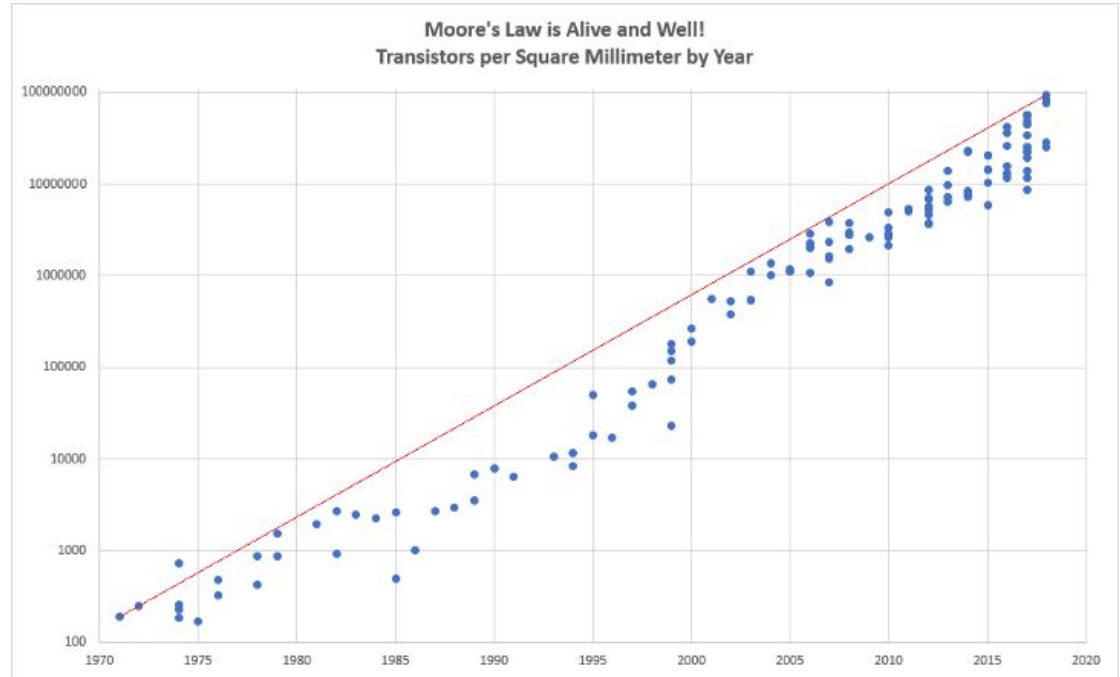


# Ley de Moore

-La ley de Moore expresa que aproximadamente cada 2 años se duplica el número de transistores en un microprocesador.

-A medida que los componentes y los ingredientes de las plataformas con base de silicio crecen en desempeño se vuelven exponencialmente más económicos de producir, y por lo tanto más abundantes, poderosos y transparentemente integrados en nuestras vidas diarias.

-Imagen: Transistores por milímetro cuadrado por año



# ¿Qué es un Algoritmo?

La palabra algoritmos se deriva del nombre **Muhammad ibn Mūsā al-Khwārizmī**, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

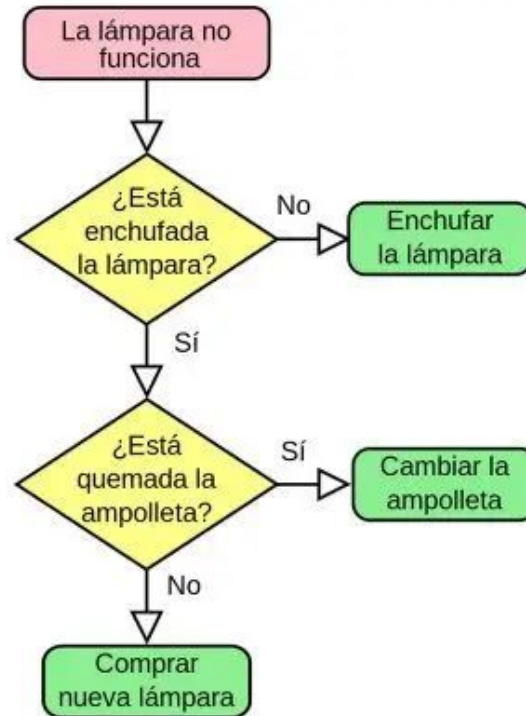
Un algoritmo se puede definir como una serie de pasos ordenados, para resolver un problema el cual posee las siguientes características:

- Debe ser **finito**, posee un número determinado de pasos.
- Debe ser **preciso**, no puede contener ambigüedades.
- Debe ser **definido**, si se sigue dos veces, se debe llegar al mismo resultado.
- Se estructura en tres partes, **Entrada**, **Proceso** y **Salida**.

Algunos ejemplos de algoritmos son instrucciones para hacer una receta de cocina, obtener el mínimo común múltiplo, etcétera.








La forma de representar un lenguaje de algoritmo es mediante un lenguaje algorítmico el cual está formado por una serie de símbolos y reglas que utilizan para describir de manera explícita un proceso.

# Gráficos (diagrama de flujo)





# Gráficos (diagrama de flujo)

	Inicio y fin del diagrama de flujo
	Entrada y salida
	Decisión, indica si realiza una comparación de valores
	Proceso
	Conector dentro de una página
	Líneas de flujo o dirección. Indica la secuencia en que se realizan las operaciones
	Subprograma

# No gráficos (pseudocódigo)

```
Inicio
    aprobados ← 0
    reprobados ← 0
    resultado ← 0
    estudiantes ← 1
Mientras estudiantes <= 10
    Leer resultado
    Si resultado == 1 entonces
        aprobados ← aprobados + 1
    SiNo
        reprobados ← reprobados + 1
    FinSi
    estudiantes ← estudiantes + 1
FinMientras
Mostrar aprobados
Mostrar reprobados
Si aprobados > 8 entonces
    Mostrar "Aumentar la colegiatura"
FinSi
Fin
```

# Fases en la resolución de problemas

Si bien el diseñar programa es un proceso creativo se pueden considerar los siguientes pasos comunes, que se deben seguir

- Análisis del problema
- Diseño del algoritmo
- Codificación
- Compilación y ejecución
- Verificación
- Depuración
- Documentación
- Mantenimiento

# Fases en la resolución de problemas

Los puntos más importantes de fases son:

**Análisis:** Se analiza el problema tomando en cuenta todas las especificaciones de los requisitos dado por el cliente o por la persona que encarga el programa.

**Diseño:** se diseña la solución, el cual dará como resultado un algoritmo que resuelve el problema.

**Codificación:** La solución se escribe detalladamente las instrucciones generando el código fuente, este es escrito en un lenguaje de programación de alto nivel.

**Ejecución, verificación y depuración:** Se realiza la ejecución del programa, se comprueba rigurosamente y se eliminan todos los errores que puedan aparecer.

**Documentación:** Se realiza la escritura del ciclo de vida del software, donde se detalla el análisis, diseño y codificación, junto con manuales y referencias.

**Mantenimiento:** Se realizan actualizaciones al programa, cada vez que sea necesario.

# Ejemplo 1

Problema: Dada una base y altura de un rectángulo calcular el área.

**Paso 1 Análisis:** Identificamos las variables de entradas que son base y altura y las variables de salida que son el área.

**Entradas:** base, altura  $\in \mathbb{R}$

**Salidas:** area  $\in \mathbb{R}$

**Proceso:** area = base \* altura

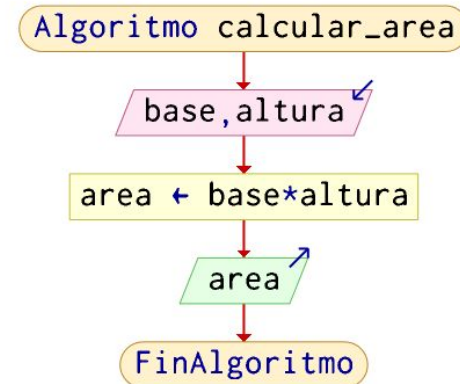
# Ejemplo 1

### Paso 2 Diseño:

Pseudocódigo:

```
1  Algoritmo calcular_area
2      Leer base, altura
3      area ← base*altura
4      Escribir area
5  FinAlgoritmo
```

Diagrama de flujo:



---

# Ejemplo (Problema)

Problema: Una tienda ofrece un descuento del 10% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.

Análisis:

Diseño:

---

# Ejemplo (Problema)

Problema: Una tienda ofrece un descuento del 10% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.

Análisis:

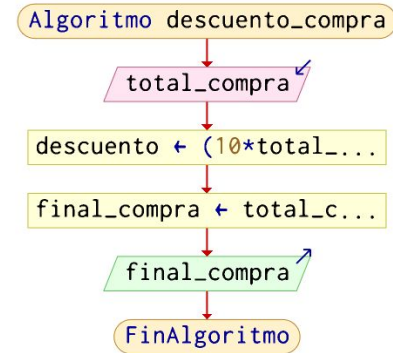
- Datos de entrada: total\_compra
- Datos de salida: final\_compra
- Proceso:
  - $\text{descuento} = (10 * \text{total\_compra}) / 100$
  - $\text{final\_compra} = \text{total\_compra} - \text{descuento}$



# Ejemplo (Problema)

Diseño:

```
Algoritmo descuentocompra
  Leer total_compra
  descuento <- (10 * total_compra) / 100
  final_compra = total_compra - descuento
  Escribir final_compra
FinAlgoritmo
```



# ¿Que es un lenguaje de programación?

Un lenguaje de programación es una serie de instrucciones, permitiendo que un programador pueda comunicarse con la computadora.

Cada lenguaje de programación tiene un conjunto de instrucciones, que la computadora entenderá directamente en su código de máquina o bien se traducirá a código máquina.

Las instrucciones básicas y comunes en casi todos los lenguajes de programación son:

- Instrucciones de entrada/salida, instrucciones de transferencias de información entre dispositivos periféricos y la memoria central, “como leer” de o “escribir en”
- Instrucciones de cálculo, permitiendo realizar operaciones aritméticas
- Instrucciones de control, estas permiten modificar la secuencia de la ejecución del programa

Los tipos de lenguaje de programación son:

- Lenguaje Máquina (Sus instrucciones son secuencias de 0 y 1 como 11010000,00110011,...)
- Lenguaje de bajo nivel (ensambladores)
- Lenguaje de alto nivel (c++, c, c#, Java, BASIC, PHP, JavaScript, Python)

# Traducción del lenguaje

Existe un proceso el cual es la traducción de un lenguaje de alto nivel a un lenguaje de máquina comprensible por la computadora. Esto se realiza mediante un programa llamado traductor, los cuales se pueden dividir en compiladores e intérpretes.

Lenguajes compilado:

- Un compilador es un programa que traduce los programas fuentes escrito en lenguaje de alto nivel a lenguaje máquina. La traducción se realiza en una sola operación denominada compilación del programa.

Lenguaje intérpretes:

- Un intérprete es un traductor que toma un programa fuente, lo traduce, luego lo ejecuta. El sistema de traducción traduce la primera sentencia del programa a lenguaje máquina, se detiene la traducción, ejecuta la sentencia, a continuación traduce la siguiente sentencia y así repite el proceso hasta terminar el programa.

# ¿Qué es Python?

- Lenguaje de programación interpretado y multiplataforma.
- Lenguaje de alto nivel.
- enfocado principalmente a la legibilidad y facilidad de aprendizaje y uso.
- Potenciado altamente para el IOT, inteligencia artificial, Backend y Data science.



# Ventajas y desventajas de Python

Ventajas	Desventajas
<ul style="list-style-type: none"><li>● Simplifica la programación.</li><li>● Flexibilidad con los tipos de datos</li><li>● Funciona en múltiples plataformas</li><li>● Mantiene una comunidad activa</li></ul>	<ul style="list-style-type: none"><li>● Al ser un lenguaje interpretado su ejecución es más lenta que los lenguajes no interpretados.</li></ul>

# Herramientas

- Una IDE es un entorno de desarrollo integrado o interactivo, que proporciona servicios que nos facilitan la programación.

Algunos IDE son:

- PyCharm
- Spyder
- Visual studio code
- Sublime Text 3
- Atom

---

# Conclusiones

¿Qué es un algoritmo?

¿Cuales son los puntos más importantes de la fases de resolución de problemas?