

# Taller Inicial: Introducción a la Lógica y Algoritmos

Johan Sebastián Aldana Ducuara

18 de septiembre de 2025

## 1. Objetivo

Desarrollar habilidades básicas en lógica y algoritmos a través del análisis de argumentos, proposiciones lógicas y la generación de pseudocódigo.

## 2. Parte 1: Fundamentos de Lógica

### 2.1. Estructura del argumento y lógica formal e informal

#### 2.1.1. Actividad 1: Identificación de argumentos

Lee los siguientes enunciados y determina cuál es la premisa y cuál es la conclusión. ¿Son argumentos válidos? ¿Por qué?

1. **Enunciado:** Si llueve, entonces la calle está mojada. Ha llovido, por lo tanto, la calle está mojada.

**Premisa 1:** Si llueve, entonces la calle está mojada.

**Premisa 2:** Ha llovido.

**Conclusión:** La calle está mojada.

**¿Es válido?** Sí, porque la conclusión sigue lógicamente de las premisas.

2. **Enunciado:** Todos los perros ladran. Rex es un perro. Por lo tanto, Rex ladra.

**Premisa 1:** Todos los perros ladran.

**Premisa 2:** Rex es un perro.

**Conclusión:** Rex ladra.

**¿Es válido?** Sí, porque la conclusión es una consecuencia lógica de las premisas.

3. **Enunciado:** María estudia mucho, por lo que debe ser una persona inteligente.

**Premisa:** María estudia mucho.

**Conclusión:** María debe ser una persona inteligente.

**¿Es válido?** No, porque la premisa no garantiza la conclusión (podría haber otras razones por las que María estudia mucho).

### 2.1.2. Actividad 2: Evaluación de argumentos

Identifica premisas verdaderas o falsas en los siguientes casos:

1. **Enunciado:** Todos los gatos son mamíferos. Todos los mamíferos vuelan. Por lo tanto, todos los gatos vuelan.

**Premisa 1:** Todos los gatos son mamíferos. (Verdadera).

**Premisa 2:** Todos los mamíferos vuelan. (Falsa).

**Conclusión:** Todos los gatos vuelan. (Falsa).

**¿Es válido?** No, porque una premisa es falsa.

2. **Enunciado:** Si Juan estudia, aprobará el examen. Juan estudió. Por lo tanto, aprobó el examen.

**Premisa 1:** Si Juan estudia, aprobará el examen.

**Premisa 2:** Juan estudió.

**Conclusión:** Juan aprobó el examen.

**¿Es válido?** Sí, porque la conclusión sigue lógicamente de las premisas.

## 2.2. Proposiciones lógicas y operadores lógicos

### 2.2.1. Actividad 3: Tablas de verdad

**¿Qué es una tabla de verdad?** Una tabla de verdad es una herramienta que muestra todos los posibles valores de verdad de una expresión lógica, dependiendo de los valores de verdad de sus variables.

**Tablas de verdad para las siguientes expresiones:** Considera  $P$ : "Hoy es lunes" y  $Q$ : ".Está soleado".

1.  $P \wedge Q$  (P AND Q).
2.  $P \vee Q$  (P OR Q).
3.  $P \rightarrow Q$  (Si P, entonces Q).
4.  $P \leftrightarrow Q$  (P si y solo si Q).

$P$	$Q$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

### 3. Parte 2: Fundamentos de Algoritmos

#### 3.1. Conceptos básicos y pseudocódigo

##### 3.1.1. Actividad 4: ¿Qué es un algoritmo?

**Definición:** Un algoritmo es un conjunto de pasos ordenados y finitos que se siguen para resolver un problema o realizar una tarea.

**Ejemplos de algoritmos en la vida cotidiana:**

- Receta de cocina.
- Instrucciones para llegar a un lugar.
- Método para resolver una ecuación matemática.

**Importancia de la programación estructurada:** Permite escribir código más claro, mantenable y fácil de depurar.

##### 3.1.2. Actividad 5: Etapas del desarrollo de un programa

Relaciona cada etapa con su definición:

1. **Análisis del problema:** Entender el problema y definir los requisitos.
2. **Diseño del algoritmo:** Crear un plan para resolver el problema.
3. **Codificación:** Escribir el código en un lenguaje de programación.
4. **Compilación y ejecución:** Convertir el código a lenguaje máquina y ejecutarlo.
5. **Verificación y depuración:** Corregir errores y asegurar que el programa funcione correctamente.
6. **Documentación:** Explicar cómo funciona el programa para futuras referencias.

### 3.2. Representación de algoritmos y estructuras de control

#### 3.2.1. Actividad 6: Pseudocódigo y diagramas de flujo

Pseudocódigo para comparar dos números:

```
Inicio
    Leer número1
    Leer número2
    Si número1 > número2 entonces
        Imprimir "El número mayor es: ", número1
    Sino
        Imprimir "El número mayor es: ", número2
    Fin Si
Fin
```

**Diagrama de flujo:** (Descripción: empezar con un óvalo “Inicio”, luego cajas para leer los números, un rombo para la comparación y flechas para las decisiones).

#### 3.2.2. Actividad 7: Uso de estructuras de control

Pseudocódigo para verificar si un número es par o impar:

```
Inicio
    Leer número
    Si número % 2 == 0 entonces
        Imprimir "El número es par"
    Sino
        Imprimir "El número es impar"
    Fin Si
Fin
```

#### 3.2.3. Actividad 8: Estructuras de repetición

Pseudocódigo para sumar números del 1 al 20:

```
Inicio
    suma = 0
    Para i desde 1 hasta 20 hacer
        suma = suma + i
    Fin Para
    Imprimir "La suma total es: ", suma
Fin
```

**Pseudocódigo para sumar solo números pares:**

```
Inicio
    suma = 0
    Para i desde 1 hasta 20 hacer
        Si i % 2 == 0 entonces
            suma = suma + i
        Fin Si
    Fin Para
    Imprimir "La suma de los números pares es: ", suma
Fin
```

## 4. Cierre y Reflexión

- **Lo más fácil:** Identificar argumentos y construir tablas de verdad.
- **Lo más difícil:** Diseñar algoritmos complejos o depurar errores.
- **Relación entre lógica y programación:** La lógica es la base de la programación, ya que permite estructurar el pensamiento para resolver problemas.
- **Aplicaciones prácticas:** Los algoritmos se usan en recetas, rutas de transporte, procesos industriales, etc.