

Manual de Laboratorio LÓGICA DE SISTEMAS



**FACULTAD DE INGENIERIA
PRIMER SEMESTRE 2026**

PROGRAMACIÓN DE ACTIVIDADES

Práctica	Día / Hora	Tema central	Objetivo	Contenidos
1	Lunes 8:00 a 12:00	Fundamentos de lógica y conectivos	Comprender los principios básicos de la lógica y representar razonamientos mediante conectivos	<ul style="list-style-type: none"> - Concepto de lógica - Premisa y conclusión - Tipos de razonamiento - Conectivos lógicos (AND, OR, NOT, \rightarrow, \leftrightarrow) - Traducción básica a lenguaje lógico
2	Martes 8:00 a 12:00	Tablas de verdad y sistema binario	Analizar expresiones lógicas mediante tablas de verdad y comprender su relación con el sistema binario	<ul style="list-style-type: none"> - Tablas de verdad (1 y 2 variables) - Evaluación de expresiones lógicas - Condicional y bicondicional - Introducción a números binarios (0 y 1) - Relación lógica \leftrightarrow computación
3	Miércoles 8:00 a 12:00	Lógica aplicada en programación (C++)	Aplicar la lógica en programación mediante estructuras condicionales y operadores lógicos	<ul style="list-style-type: none"> - Introducción a C++ - Variables y entrada de datos - Estructuras if y else - Operadores lógicos (&&, , !) - Uso de switch - Resolución de problemas simples
4	Jueves 8:00 a 12:00	Proyecto integrador: lógica en sistemas reales (Arduino)	Implementar un sistema real utilizando lógica y programación para el control de dispositivos	<ul style="list-style-type: none"> - Arduino y salidas digitales - Control de LEDs - Secuencia lógica (semáforo) - Uso de condiciones en sistemas físicos - Modificación de comportamiento (modo noche) - Integración lógica + programación

Materiales

Materiales de uso personal:

Equipo básico:

- Computadora (laptop o de escritorio)
- Cuaderno o libreta para anotaciones

Software (se dará mayor información en la 1ra clase):

- Compilador de C++ (Dev C++, Code::Blocks o similar)
- Arduino IDE

Materiales que deben de adquirir en grupos de máximo 6 personas:

Material para práctica con Arduino:

- 1 Arduino Nano (o genérico)
- Cargador de teléfono compatible para el Arduino
- 1 Protoboard
- 3 LEDs:

1 rojo
1 amarillo
1 verde

- 3 resistencias (220 Ω o 330 Ω)
- Cables de conexión (jumpers)
- Cable USB para Arduino

INSTRUCCIONES PARA REALIZAR LAS PRÁCTICAS

Para la realización adecuada de las prácticas deberán atenderse las siguientes indicaciones:

1. Presentarse puntualmente a la hora del inicio del laboratorio y permanecer durante la duración de este.
2. Realizar las actividades y hojas de trabajo planteadas durante la práctica.
3. Participación y cuidado de cada uno de los integrantes del grupo en todo momento de la práctica.
4. Conocer la teoría (leer el manual antes de presentarse a cada práctica).
5. **No se permite el uso de teléfono celular dentro del laboratorio**, Si tiene llamadas laborales deberá atender las mismas únicamente en el horario de receso.
6. Si sale del salón de clases sin la autorización del docente perderá el valor de la práctica.
7. No puede atender visitas durante la realización de la práctica.
8. El horario de receso es únicamente de 15 minutos.
9. **Respeto dentro del laboratorio hacia los catedráticos o compañeros (as).**
10. Cada grupo debe revisar cuidadosamente el equipo que le corresponde; al ingresar al laboratorio, deben de elegir un coordinador por grupo. Al terminar la práctica, deben permanecer dentro del laboratorio únicamente dichos coordinadores para que juntamente con el instructor revisen, mesa por mesa, que el equipo utilizado se encuentre en las mismas condiciones en las que fue entregado. En caso de cualquier faltante o rotura, el grupo completo debe encargarse de reponer el equipo.
11. Al finalizar la práctica deberá entregarse al instructor la hoja con los datos originales, que contiene en una forma breve y concisa todas las observaciones

NORMAS DE SEGURIDAD Y PREVENCIÓN DE ACCIDENTES EN EL LABORATORIO

1. Los ojos deben ser protegidos durante todo el periodo de laboratorio.
2. Cualquier accidente, aún la menor lesión debe informarse de inmediato al instructor del laboratorio.
3. No intente ningún experimento no autorizado, sólo deben realizarse las prácticas explicadas por el instructor y la guía de laboratorio.
4. Nunca debe dejar de prestar atención al experimento en curso.
5. Leer el manual de laboratorio cuidadosamente antes de ingresar al mismo, esto le ayudará en la toma de datos y al desarrollo de la práctica
6. Mantener siempre limpias las mesas y aparatos de laboratorio y colocar sobre estas solo aquellos utensilios que sean indispensables para la práctica.

La falta a cualquiera de los incisos anteriores será motivo de una inasistencia.

Considere que se prohíbe terminantemente comer, beber y fumar. Éstos también serán motivos para ser retirado de la práctica.

Recuerde que para tener derecho al puntaje y aprobar el curso deberá presentarse a las prácticas y realizar las evaluaciones en línea, las cuales estarán habilitadas del 25 de mayo 2026 a las 8:00 al 29 de mayo 2026 a las 18:00.

Introducción

La lógica es una disciplina fundamental en la formación científica, matemática e ingenieril, pues proporciona las herramientas conceptuales y formales necesarias para analizar, construir y evaluar razonamientos. Su estudio permite comprender la estructura de los argumentos, distinguir entre inferencias válidas e inválidas, formalizar enunciados del lenguaje natural y trabajar con sistemas simbólicos precisos y rigurosos.

El desarrollo de la lógica moderna ha dado lugar a lenguajes formales, sistemas axiomáticos, métodos de deducción y modelos semánticos que permiten estudiar el razonamiento desde una perspectiva matemática. Estos elementos son esenciales para áreas como la computación, la inteligencia artificial, la teoría de la programación, la verificación formal, la matemática y la filosofía analítica.

Este manual reúne una serie de prácticas diseñadas para guiar al estudiante desde los conceptos fundamentales de la lógica clásica hasta sistemas más avanzados como la lógica de predicados, la deducción natural y algunas extensiones contemporáneas. Cada laboratorio combina actividades teóricas y ejercicios prácticos con software, permitiendo al estudiante experimentar con traducciones, tablas de verdad, deducciones, modelos e interpretaciones.

Propósitos de aplicación

- Facilitar el aprendizaje significativo de los principios fundamentales de la lógica formal mediante actividades teóricas y experimentales.
- Desarrollar habilidades de análisis, formalización, deducción y evaluación de argumentos.
- Promover el uso de lenguajes formales y sistemas simbólicos para representar razonamientos.
- Integrar herramientas computacionales para la verificación de tablas de verdad, deducciones, modelos e interpretaciones.
- Fomentar el pensamiento crítico, la precisión conceptual y la capacidad de resolver problemas mediante métodos lógicos.

Objetivos

- Comprender los principios básicos de la lógica, identificando argumentos, premisas y conclusiones en situaciones cotidianas.
- Analizar razonamientos y estructuras lógicas, diferenciando entre argumentos correctos e incorrectos.
- Interpretar y utilizar conectivos lógicos (AND, OR, NOT) para representar ideas de manera estructurada.
- Traducir enunciados del lenguaje natural a expresiones lógicas, facilitando su análisis y comprensión.
- Construir e interpretar tablas de verdad, evaluando diferentes escenarios y resultados posibles.
- Aplicar la lógica en la resolución de problemas prácticos, utilizando herramientas de simulación.
- Relacionar la lógica con la programación, comprendiendo cómo se implementa en estructuras de decisión.
- Desarrollar programas básicos en C++, utilizando condiciones (if, else) y operadores lógicos para resolver problemas simples.

Campo de aplicación

El presente manual está dirigido a estudiantes que tengan asignado el curso de **Lógica de Sistemas FE727** de la carrera de Ingeniería en Sistemas

PRÁCTICA No. 01

Introducción a la lógica y conectivos lógicos

Introducción

La lógica es una herramienta fundamental para analizar razonamientos y tomar decisiones de forma correcta. En el ámbito de los sistemas y la programación, la lógica permite representar situaciones mediante condiciones que pueden ser verdaderas o falsas.

Un argumento lógico está formado por premisas (información inicial) y una conclusión (resultado). La validez de un argumento depende de su estructura, no necesariamente de su contenido.

En lógica proposicional se utilizan conectivos para relacionar proposiciones. Los principales son:

- **Conjunción (AND, \wedge):** Se cumple cuando ambas condiciones son verdaderas
- **Disyunción (OR, \vee):** Se cumple cuando al menos una condición es verdadera
- **Negación (NOT, \neg):** Invierte el valor de verdad
- **Condicional (\rightarrow):** “Si... entonces...”
- **Bicondicional (\leftrightarrow):** “Si y solo si...”

Estos conectivos permiten modelar decisiones que luego se aplican directamente en programación y sistemas reales.

Objetivos

- Comprender el concepto de lógica y su aplicación en la vida cotidiana
- Identificar premisas y conclusiones en un argumento
- Reconocer los principales conectivos lógicos
- Representar situaciones simples mediante expresiones lógicas
- Analizar el comportamiento de los conectivos en diferentes escenarios

Materiales

- Computadora
- Cuaderno

Procedimiento

Parte A: Identificación de elementos lógicos

- ✓ Analizar los siguientes enunciados:
 - “Si estudio, entonces apruebo el curso”
 - “Si llueve, llevo paraguas”
 - “Si tengo dinero y tiempo, voy al cine”
- ✓ Para cada enunciado, identificar:
 - Premisa(s)
 - Conclusión

Parte B: Clasificación de conectivos

- ✓ Identificar el conectivo lógico presente en cada caso:
 - “Estudio y hago tareas”
 - “Voy al cine o me quedo en casa”
 - “No voy a clases”
 - “Si trabajo, entonces gano dinero”
 - “Apruebo si y solo si estudio”
- ✓ Escribir el tipo de conectivo:
 - AND (\wedge)
 - OR (\vee)
 - NOT (\neg)
 - Condicional (\rightarrow)
 - Bicondicional (\leftrightarrow)

Parte C: Representación simbólica

- ✓ Definir proposiciones simples, por ejemplo:
 - p: “Estudio”
 - q: “Apruebo”
- ✓ Traducir los siguientes enunciados a lenguaje lógico:

- “Estudio y apruebo”
- “No estudio”
- “Si estudio, entonces apruebo”
- “Estudio o trabajo”
- “Apruebo si y solo si estudio”

Parte D: Aplicación en simulador (opcional)

- ✓ Utilizar un simulador lógico para representar:
 - Una compuerta AND
 - Una compuerta OR
 - Una compuerta NOT
- ✓ Cambiar los valores de entrada (verdadero/falso) y observar resultados

Hoja de trabajo 1

¿Qué diferencia existe entre una premisa y una conclusión?

¿Puede un argumento tener premisas verdaderas y una conclusión falsa? Explique.

¿Cuál es la diferencia entre AND y OR en una situación real?

¿Qué efecto tiene la negación (NOT) sobre una proposición?

¿En qué situaciones de la vida diaria utilizas condicionales sin darte cuenta?

¿Por qué es importante representar problemas con lógica antes de programar?

PRÁCTICA No. 02

Tablas de verdad, evaluación lógica y números binarios

Introducción

Las tablas de verdad son una herramienta fundamental en la lógica proposicional, ya que permiten analizar todas las combinaciones posibles de valores de verdad de una expresión lógica.

Cada proposición puede tomar dos valores:

- Verdadero (V)
- Falso (F)

En computación, estos valores se representan como:

- Verdadero = 1
- Falso = 0

Esto da origen al sistema binario, que es la base del funcionamiento de las computadoras. Todas las operaciones lógicas pueden representarse mediante combinaciones de 0 y 1.

El número de combinaciones en una tabla de verdad depende de la cantidad de variables:

- 1 variable → 2 combinaciones
- 2 variables → 4 combinaciones
- 3 variables → 8 combinaciones

Los principales conectivos lógicos son:

- AND (\wedge): verdadero solo si ambas proposiciones son verdaderas
- OR (\vee): verdadero si al menos una es verdadera
- NOT (\neg): invierte el valor
- Condicional (\rightarrow): falso solo cuando la premisa es verdadera y la conclusión es falsa
- Bicondicional (\leftrightarrow): verdadero cuando ambas proposiciones tienen el mismo valor

Estas herramientas permiten modelar decisiones que luego se utilizan directamente en programación y sistemas digitales.

Objetivos

- Construir tablas de verdad con una y dos variables
- Analizar el comportamiento de los conectivos lógicos
- Evaluar expresiones lógicas paso a paso
- Comprender la relación entre lógica proposicional y sistema binario
- Interpretar resultados lógicos como base de la computación

Materiales

- Computadora
- Cuaderno
- Hoja de cálculo (opcional)
- Simulador lógico (opcional)

Procedimiento

Parte A: Tablas de verdad básicas

- ✓ Construir la tabla de verdad de una variable:

P $\neg P$

- ✓ Completar los valores correspondientes

Parte B: Tablas de verdad con dos variables

- ✓ Construir las siguientes tablas:

- $p \wedge q$
- $p \vee q$

Parte C: Condicional y bicondicional

- ✓ Construir las tablas de verdad de:

- $p \rightarrow q$
- $p \leftrightarrow q$

Parte D: Evaluación de expresiones lógicas

- ✓ Construir la tabla de verdad de:

- $(p \wedge q) \vee \neg p$

Parte E: Introducción a números binarios

✓ Convertir los siguientes números a números binarios:

- 15
- 23
- 55
- 68
- 99

✓ Considerar:

$$p = 1$$
$$q = 0$$

✓ Evaluar:

- $p \wedge q$
- $p \vee q$
- $\neg p$

✓ Interpretar el siguiente caso:

“El sistema funciona si hay energía y señal”

Definir:

- Energía = 1 o 0
- Señal = 1 o 0

✓ Construir una tabla con todas las combinaciones posibles

Hoja de trabajo 2

¿Cuántas combinaciones existen con dos variables? ¿Por qué?

¿Qué relación existe entre los valores lógico (V/F) y binarios (1/0)?

¿En qué caso el condicional ($p \rightarrow q$) es falso? Explique con un ejemplo real.

¿Qué diferencia existe entre AND y OR al trabajar con valores binarios?

¿Por qué se utilizan números binarios en las computadoras?

¿Cómo ayudan las tablas de verdad a entender el funcionamiento de un programa?

PRÁCTICA No. 03

Lógica aplicada en programación (C++)

Introducción

La programación utiliza la lógica para tomar decisiones dentro de un sistema. Las computadoras evalúan condiciones que pueden ser verdaderas (1) o falsas (0), y en función de ello ejecutan diferentes acciones.

En el lenguaje C++, estas decisiones se implementan mediante estructuras de control como if, else y operadores lógicos.

Relación entre lógica y C++:

- AND (\wedge) \rightarrow `&&`
- OR (\vee) \rightarrow `||`
- NOT (\neg) \rightarrow `!`

Estas estructuras permiten crear programas que reaccionan a distintas condiciones, como validar datos, permitir accesos o ejecutar procesos específicos.

Una estructura básica de decisión en C++ es:

```
if (condición) {  
    // instrucciones  
} else {  
    // instrucciones alternativas  
}
```

También es posible evaluar múltiples opciones mediante switch, lo cual es útil cuando se tienen varias condiciones posibles.

El uso correcto de la lógica en programación es esencial para el desarrollo de sistemas funcionales y confiables.

Objetivos

- Comprender la relación entre lógica proposicional y programación
- Utilizar operadores lógicos en C++
- Implementar estructuras condicionales (if, else)
- Aplicar lógica en la resolución de problemas simples
- Desarrollar programas básicos funcionales

Materiales

- Computadora
- Compilador de C++ (Dev C++, Code::Blocks o similar)

Procedimiento

Parte A: Programa básico en C++

- ✓ Crear un programa simple:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hola mundo";
    return 0;
}
```

Parte B: Uso de if (condición simple)

- ✓ Crear un programa que evalúe la edad:

```
#include <iostream>
using namespace std;

int main() {
    int edad;
    cout << "Ingrese su edad: ";
    cin >> edad;

    if (edad >= 18) {
        cout << "Mayor de edad";
    }

    return 0;
}
```

Parte C: Uso de if – else

- ✓ Modificar el programa anterior:

```
if (edad >= 18) {
    cout << "Mayor de edad";
}
```

```
} else {  
    cout << "Menor de edad";  
}
```

Parte D: Uso de operadores lógicos

- ✓ Crear un programa que evalúe dos condiciones:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int edad;  
    bool tieneID;  
  
    cout << "Ingrese edad: ";  
    cin >> edad;  
  
    cout << "Tiene identificacion? (1=Si, 0=No): ";  
    cin >> tieneID;  
  
    if (edad >= 18 && tieneID) {  
        cout << "Acceso permitido";  
    } else {  
        cout << "Acceso denegado";  
    }  
  
    return 0;  
}
```

Parte E: Uso de OR y NOT

- ✓ Crear un programa que evalúe:

```
if (edad < 18 || !tieneID) {  
    cout << "No puede ingresar";  
}
```

Parte F: Uso de switch

- ✓ Crear un menú de opciones:

```
#include <iostream>  
using namespace std;
```

```
int main() {
    int opcion;

    cout << "1. Saludar" << endl;
    cout << "2. Despedirse" << endl;
    cout << "Seleccione una opcion: ";
    cin >> opcion;

    switch(opcion) {
        case 1:
            cout << "Hola!";
            break;
        case 2:
            cout << "Adios!";
            break;
        default:
            cout << "Opcion invalida";
    }

    return 0;
}
```

Parte G: Ejercicio práctico

- ✓ Desarrollar un programa que simule un sistema de acceso:
- ✓ Condiciones:
 - Usuario correcto
 - Contraseña correcta

Mostrar:

- "Acceso permitido"
- "Acceso denegado"

Hoja de trabajo 3

- ¿Cómo se relacionan los operadores lógicos con los vistos en lógica proposicional?
- ¿Qué ocurre si una condición en un if es falsa?
- ¿Cuál es la diferencia entre usar && y || en un programa?
- ¿Qué función cumple el operador NOT (!) en programación?
- ¿En qué casos es más conveniente usar switch en lugar de if?
- ¿Por qué es importante validar condiciones en un sistema?

PRÁCTICA No. 04

Semáforo con Arduino (lógica aplicada)

Introducción

La lógica no solo se utiliza en teoría o programación, sino también en sistemas reales. Los dispositivos electrónicos, como los sistemas de control de tráfico, funcionan a partir de condiciones lógicas que determinan su comportamiento.

Un semáforo es un ejemplo claro de aplicación de lógica, ya que sigue una secuencia de estados:

- Verde → avanzar
- Amarillo → precaución
- Rojo → detenerse

En Arduino, estas decisiones se programan mediante instrucciones condicionales y control de salidas digitales (encender y apagar LEDs).

Esta práctica integra los conocimientos adquiridos en el curso:

- Lógica proposicional
- Tablas de verdad
- Programación en C++
- Uso de condiciones

Objetivos

- Aplicar la lógica en un sistema físico real
- Comprender el uso de salidas digitales en Arduino
- Programar una secuencia lógica utilizando condiciones
- Implementar un sistema básico de semáforo
- Integrar conocimientos de lógica y programación

Materiales

- Arduino (UNO o similar)
- Protoboard
- LEDs (rojo, amarillo, verde)
- 3 resistencias (220Ω o 330Ω)
- Cables de conexión
- Computadora con Arduino IDE

Procedimiento

Parte A: Conexión del circuito

- ✓ Conectar los LEDs al Arduino:
 - LED rojo → Pin 8
 - LED amarillo → Pin 9
 - LED verde → Pin 10
- ✓ Conectar cada LED en serie con una resistencia
- ✓ Conectar el negativo a GND

Parte B: Programa básico

- ✓ Escribir el siguiente código en Arduino:

```
int rojo = 8;
int amarillo = 9;
int verde = 10;

void setup() {
  pinMode(rojo, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(verde, OUTPUT);
}

void loop() {
  // Verde encendido
  digitalWrite(verde, HIGH);
  digitalWrite(amarillo, LOW);
  digitalWrite(rojo, LOW);
  delay(3000);

  // Amarillo encendido
  digitalWrite(verde, LOW);
  digitalWrite(amarillo, HIGH);
  digitalWrite(rojo, LOW);
  delay(2000);

  // Rojo encendido
  digitalWrite(verde, LOW);
  digitalWrite(amarillo, LOW);
```

```
digitalWrite(rojo, HIGH);  
delay(3000);  
}
```

- ✓ Cargar el programa al Arduino
- ✓ Verificar que el semáforo funcione correctamente

Parte C: Análisis lógico

- ✓ Definir variables lógicas:
 - Verde = 1 cuando está encendido
 - Amarillo = 1 cuando está encendido
 - Rojo = 1 cuando está encendido

Parte D: Modificación del sistema

- ✓ Modificar el programa para que:
 - El LED rojo dure más tiempo (5 segundos)
 - El LED verde parpadee antes de cambiar a amarillo

Parte E: Aplicación de lógica

- ✓ Implementar una condición:

Ejemplo:

“Si el sistema está en modo noche, el semáforo solo parpadea en amarillo”

Simular este comportamiento agregando una variable y utilizando if.

Hoja de trabajo No. 4

- ¿Cómo se aplica la lógica en el funcionamiento del semáforo?
- ¿Qué representa el valor HIGH y LOW en Arduino?
- ¿Qué relación existe entre este sistema y las tablas de verdad?
- ¿Qué ocurriría si dos LEDs se encendieran al mismo tiempo?
- ¿Cómo se puede mejorar el sistema del semáforo usando más condiciones?
- ¿En qué otros sistemas reales se aplica este tipo de lógica?

REFERENCIAS BIBLIOGRÁFICAS

1. Gutiérrez, C. (2018). Lógica para ciencias de la computación.
2. Copi, I. M., Cohen, C., & McMahon, K. (2014). Introducción a la lógica (14.^a ed.). Pearson.