

Matemáticas Discretas

Lógica proposicional

Clase 1



La excelencia no se improvisa



INTRODUCCIÓN DE LA CLASE

Imagina que estás diseñando una regla para un firewall: «Si el tráfico no viene de una IP confiable y además intenta acceder a un puerto restringido, entonces debe bloquearse». ¿Cómo podríamos representar esta condición de manera precisa y sin ambigüedades? En esta primera clase de la asignatura de Matemáticas Discretas, comenzaremos a desarrollar una de las herramientas más poderosas para estructurar el pensamiento lógico y riguroso: la lógica proposicional. Este sistema nos permitirá expresar, analizar y verificar enunciados lógicos mediante símbolos formales, algo fundamental tanto para la programación como para la ciberseguridad.

A lo largo de la clase exploraremos qué son las proposiciones, cómo se combinan mediante conectivos lógicos como la negación, conjunción o implicación, y cómo podemos traducir situaciones del lenguaje natural -como condiciones de acceso, reglas de decisión o políticas de seguridad- en fórmulas lógicas precisas. Aprenderás también a interpretar su sintaxis correctamente y a aplicar estos conceptos a través de ejemplos prácticos. Al finalizar, estarás en capacidad de reconocer y formular proposiciones lógicas, lo cual constituye una base esencial para todo el razonamiento formal que encontrarás más adelante en el curso.

Clase 1: Lógica proposicional

RDA 1: Comprender los conceptos fundamentales de la Matemática Discreta, incluyendo lógica matemática, teoría de conjuntos, relaciones, algoritmos y teoría de grafos, destacando su importancia y aplicabilidad en el análisis de problemas discretos.

1. Lógica proposicional

Desde la antigüedad, la humanidad ha buscado herramientas para razonar correctamente. Aristóteles fue uno de los primeros en formalizar este esfuerzo mediante un conjunto de reglas para el razonamiento deductivo, cuya intención era establecer una base sólida para todas las ciencias. Mucho más tarde, durante el siglo XVII, Gottfried Leibniz propuso una idea revolucionaria: usar símbolos para representar el razonamiento, del mismo modo en que la notación algebraica había transformado el

manejo de los números (Lenzen 2025). Esta visión simbólica tomó forma en el siglo XIX gracias a George Boole y Augustus De Morgan, quienes establecieron las bases de la lógica moderna o lógica simbólica.

Figura 1

Retratos de Gottfried Leibniz, Augustus De Morgan y George Boole



Nota: Imágenes recuperadas de Gottfried Leibniz (s.f.), Augustus De Morgan (s.f.) y George Boole (s.f.).

En el siguiente enlace encontrarás un video que presenta la vida y obra de George Boole, destacando sus principales aportes a la lógica simbólica y su impacto en el desarrollo del pensamiento lógico-formal:

- **Título del enlace:** George Boole: biografía y aportaciones
- **Descripción del enlace:** Este video expone aspectos relevantes de la trayectoria de George Boole, así como su influencia en la lógica moderna.

Enlace: <https://youtu.be/iSoFsrCqQ5s?si=RTABxLuUNLfPP3ai>

Hoy en día, la lógica proposicional es indispensable en múltiples campos de la computación, como el diseño de circuitos digitales, las bases de datos relacionales, la teoría de autómatas o la inteligencia artificial. Pero también tiene un valor universal: nos permite razonar con precisión, sin ambigüedad, evaluando no el contenido de las afirmaciones, sino su estructura lógica. Por ejemplo, si aceptamos como verdaderas las premisas «Todos los matemáticos usan sandalias» y «Cualquiera que use sandalias es un algebrista», entonces, sin importar lo extraño que suene, la conclusión lógica «Todos los matemáticos son algebristas» debe aceptarse como verdadera dentro de ese sistema de razonamiento.

La lógica proposicional también es fundamental para asegurar la corrección de programas y sistemas. Consideremos el desarrollo de un algoritmo que calcule rutas óptimas entre ciudades. Para pocos datos, podemos verificar los resultados manualmente; pero cuando el número de entradas crece, ya no es viable comprobar todo a mano. ¿Cómo podemos entonces confiar en que el programa funciona siempre? La respuesta es: necesitamos un argumento lógico que demuestre su validez.

Incluso en contextos cotidianos, entender la lógica puede evitar confusiones. Pensemos en este decreto de la ciudad de Naperville: «Será ilegal que una persona tenga más de tres perros y tres gatos en su propiedad» (Johnsonbaugh 2018). ¿Qué sucede si alguien tiene cinco perros y ningún gato? Este tipo de afirmación ambigua se puede analizar y desambiguar mediante la lógica proposicional, revelando cómo una interpretación clara puede cambiar completamente la aplicación de una norma.

1.1. Propositiones y conectivos lógicos

Antes de hablar sobre lógica matemática, necesitamos responder una pregunta fundamental: ¿todas las oraciones que usamos al hablar o escribir son útiles para hacer lógica? La respuesta es no. No toda oración es útil en lógica, y para entender por qué, tenemos que fijarnos en qué función cumple esa oración.

En general, cuando alguien dice algo, lo hace con una intención: informar, expresar emociones o pedir algo. Esto se llama función del lenguaje. Mira estos tres ejemplos:

- **Informar:** «La Tierra gira alrededor del Sol». Aquí, quien habla está dando un dato, algo que puede ser verdadero o falso.
- **Expresar:** «¡Qué emoción este concierto!». Aquí, la persona está mostrando cómo se siente.
- **Pedir o dirigir:** «Pásame la sal, por favor». Aquí, se espera que el otro haga algo.

De estos tres tipos, solo nos interesa uno en lógica: el primero. Solo nos sirven las oraciones que informan algo que puede ser verdadero o falso. Entonces, ¿cómo se llama una oración de ese tipo en lógica? Se llama **proposición**. De esta manera, una proposición es una oración que afirma algo y puede ser verdadera o falsa, pero no ambas al mismo tiempo.

Por ejemplo, observa estas oraciones:

- El ECU 911 es el sistema nacional de emergencias del Ecuador.
- El presidente del Ecuador en 2020 fue Gustavo Petro.
- Todo sistema conectado a Internet puede ser vulnerable a un ciberataque.
- El volcán Cotopaxi hará erupción este año.
- Configura un firewall en todos los servidores antes del viernes.

Solo las oraciones a), b), c) y d) son proposiciones, porque todas afirman algo que puede ser verdadero o falso, aunque en el caso de la d) no sepamos actualmente cuál es su **valor de verdad**. En cambio, la oración e) no es una proposición, ya que no afirma nada que pueda ser evaluado como verdadero o falso: es una orden. En lógica, solo trabajamos con oraciones declarativas que tienen un valor de verdad bien definido.

Para trabajar con proposiciones de manera más clara, vamos a representarlas con letras, igual que en álgebra usamos letras para representar números. Así, en lugar de escribir frases largas una y otra vez, usamos símbolos como p , q , r , etc.

¿Por qué hacemos esto? Porque en lógica no nos interesa el contenido de las frases, sino cómo se relacionan entre ellas. Esto es fundamental para analizar argumentos.

Veamos como ejemplo este argumento tomado del texto de Epp (2020):

Si la sintaxis del programa es defectuosa o si los resultados de la ejecución del programa dan como resultado una división entre cero, entonces la computadora va a generar un mensaje de error. Por tanto, si la computadora no genera un mensaje de error, entonces la sintaxis del programa es correcta y la ejecución del programa no da como resultado una división entre cero.

Aquí hay muchas palabras, pero si lo analizamos, todo se puede reducir a proposiciones. Para manejarlo mejor, vamos a dar un nombre a cada una:

- p : La sintaxis del programa es defectuosa.
- q : La ejecución del programa da como resultado una división entre cero.
- r : La computadora genera un mensaje de error.

Ahora que tenemos identificadas las proposiciones individuales, queremos escribirlas de forma que conserven la estructura del argumento original. En el texto, notamos palabras como «o», «entonces», «no», «y», que conectan las distintas partes del razonamiento.

Esas palabras son claves porque nos indican cómo se relacionan las proposiciones entre sí. En lógica, usamos símbolos especiales para representar esas conexiones, y a esos símbolos los llamamos **conectores lógicos**.

Los principales conectores lógicos que utilizaremos en este curso son:

Tabla 1

Conectores lógicos básicos

Nombre	Significado	Escritura	Símbolo
Conjunción	Ambas proposiciones ocurren	y	$\wedge$
Disyunción	Al menos una proposición ocurre	o	$\vee$
Negación	Se niega una proposición	no	$\neg$
Implicación	Si una ocurre, entonces ocurre la otra	si... entonces...	$\rightarrow$
Doble implicación	Ocorre la una si y solo si ocurre la otra	... si y solo si...	$\leftrightarrow$

Nota: creación propia (Merino, A., 2025).

Con estos conectores, podemos tomar proposiciones simples como p , q y r , y combinarlas para formar proposiciones más complejas que reflejen argumentos reales. Por ejemplo, el argumento anterior puede ser representado así:

$$((p \vee q) \rightarrow r) \rightarrow (\neg r \rightarrow (\neg p \wedge \neg q)).$$

1.2. Sintaxis de fórmulas

Hasta ahora hemos hablado de proposiciones y cómo se conectan entre sí usando conectores lógicos. Pero si queremos trabajar de forma rigurosa, necesitamos definir con precisión qué se considera una expresión válida en lógica proposicional.

En otras palabras: ¿cómo sabemos si una fórmula está bien escrita? Para eso, vamos a introducir una noción fundamental: la *forma proposicional*.

Definición 1: Forma Proposicional

Una forma proposicional es una cadena de caracteres que se genera mediante las siguientes reglas:

1. Las letras minúsculas del alfabeto (como p , q , r , etc.) son formas proposicionales. A estas se les llama *letras proposicionales*.
2. Si P y Q son formas proposicionales, entonces también lo son las siguientes:

$$\neg P, \quad P \wedge Q, \quad P \vee Q, \quad P \rightarrow Q, \quad P \leftrightarrow Q.$$

3. Nada más es una forma proposicional salvo lo que se construye usando las reglas anteriores.

A las formas proposicionales las llamaremos directamente *proposición*. Esta definición nos da un marco claro para construir fórmulas lógicas: empezamos con letras (como p , q) y las combinamos usando conectores para formar proposiciones más complejas. La lógica, en este sentido, funciona como un lenguaje formal con reglas de formación bien definidas.

En la próxima clase veremos cómo evaluar si una de estas proposiciones es verdadera o falsa: eso nos llevará al concepto de tablas de verdad.

Para entender mejor cómo funciona la sintaxis de las fórmulas, veamos algunos ejemplos concretos. Algunas expresiones están bien formadas y siguen las reglas que definimos para las formas proposicionales. Otras, en cambio, parecen razonables a simple vista, pero no cumplen con las reglas formales del lenguaje lógico. En la siguiente tabla comparamos ambas situaciones e indicamos qué errores aparecen en los casos incorrectos:

Tabla 2*Ejemplos de formas proposicionales válidas e inválidas*

Expresión	Justificación / Error
Expresiones que sí son fórmulas proposicionales	
p	Es una letra proposicional.
$\neg p$	Aplicación válida de la negación.
$p \wedge q$	Conjunción de dos formas proposicionales.
$(p \vee \neg q)$	Combinación válida de proposiciones con conectores.
$\rightarrow r$	
$\neg(p \vee q)$	Negación aplicada a disyunción, luego doble implicación.
$\leftrightarrow s$	
Expresiones que no son fórmulas proposicionales	
$\wedge pq$	Conector aparece sin operandos previos; orden inválido.
$p \vee \rightarrow q$	Faltan operandos para el conector de implicación.
$\neg \wedge p$	Conectores mal aplicados: negación seguida de conjunción.
$\neg(p \vee)$	Faltan elementos en la disyunción: expresión incompleta.
$pq \rightarrow$	Faltan paréntesis o conectores entre p y q ; sintaxis incorrecta.
$p \rightarrow q \vee r$	Ambigüedad: no hay paréntesis para determinar si es $(p \rightarrow q) \vee r$ o $p \rightarrow (q \vee r)$.

Nota: creación propia (Merino, A., 2025).

1.3. Ejemplos básicos y ejercicios

Ahora que ya conoces los elementos fundamentales de la lógica proposicional —las proposiciones, los conectores lógicos y la forma correcta de escribir fórmulas— es momento de ponerlos en práctica. En esta sección trabajaremos la siguiente tarea: **Traducción del lenguaje natural al lenguaje lógico**, lo que significa identificar las proposiciones simples y expresar oraciones más complejas usando conectores lógicos.

Para realizar una traducción correcta, seguimos estos pasos:

4. **Identificar las proposiciones simples:** Detecta las partes básicas del enunciado que afirman algo y asocia una letra a cada proposición.

5. **Reconocer los conectores:** Determina si en el enunciado hay palabras como «y», «o», «no», «si..., entonces...», «si y solo si», y tradúcelas a símbolos lógicos.
6. **Escribir la expresión lógica:** Usa las letras proposicionales y los conectores para construir la fórmula correspondiente, respetando el uso de paréntesis si es necesario.

Ejemplo 1.

Queremos traducir la siguiente frase del lenguaje ordinario al lenguaje lógico formal:

«Si estudio lógica o resuelvo ejercicios, entonces aprobaré el curso.»

- **Paso 1.** Identificamos las proposiciones simples que componen el enunciado:
 - p : Estudio lógica.
 - q : Resuelvo ejercicios.
 - r : Aprobaré el curso.
- **Paso 2.** Reconocemos los conectores del lenguaje natural:
 - «o» se traduce como disyunción: $\vee$
 - «si..., entonces...» se traduce como implicación: $\rightarrow$

- **Paso 3.** Escribimos la fórmula compuesta respetando la estructura lógica:

$$(p \vee q) \rightarrow r.$$

Este primer ejemplo muestra cómo los paréntesis son cruciales para agrupar la condición completa («estudio lógica o resuelvo ejercicios») y distinguirla del resultado.

Ejemplo 2.

Queremos traducir la siguiente frase del lenguaje ordinario al lenguaje lógico formal:

«No es cierto que si llueve, entonces no saldremos a caminar.»

- **Paso 1.** Identificamos las proposiciones simples que componen el enunciado:
 - p : Llueve.

- q : Saldremos a caminar.
- **Paso 2.** Reconocemos los conectores del lenguaje natural:
 - «no es cierto que» se traduce como negación: $\neg$
 - «sí..., entonces...» se traduce como implicación: $\rightarrow$
 - «no» (antes de «saldremos») se traduce como negación: $\neg$
- **Paso 3.** Escribimos la fórmula compuesta respetando la estructura lógica:

$$\neg(p \rightarrow \neg q).$$

En este caso, se introducen dos negaciones. Es importante notar que la primera negación afecta a toda la implicación, mientras que la segunda solo afecta a la proposición q , lo que se refleja en el uso de los paréntesis.

Ejemplo 3.

Queremos traducir la siguiente frase del lenguaje ordinario al lenguaje lógico formal:

«Si se detecta una intrusión, entonces el acceso no se autoriza y se activa una alerta.»

- **Paso 1.** Identificamos las proposiciones simples que componen el enunciado:
 - p : Se detecta una intrusión.
 - q : El acceso se autoriza.
 - r : Se activa una alerta.
- **Paso 2.** Reconocemos los conectores del lenguaje natural:
 - «no» se traduce como negación: $\neg$
 - «y» se traduce como conjunción: $\wedge$
 - «sí..., entonces...» se traduce como implicación: $\rightarrow$
- **Paso 3.** Escribimos la fórmula compuesta respetando la estructura lógica:

$$p \rightarrow (\neg q \wedge r).$$

Aquí vemos una implicación donde el resultado es una conjunción. La coma después de «se autoriza» nos indica que ambas condiciones siguientes («el acceso no se autoriza» y «se activa una alerta») son consecuencias de la intrusión, lo que se representa correctamente con la conjunción entre paréntesis.

Ejemplo 4.

Queremos traducir la siguiente frase del lenguaje ordinario al lenguaje lógico formal:

«Si estudio matemáticas y no pierdo clases, o si realizo todas las tareas, entonces aprobaré el examen.»

- **Paso 1.** Identificamos las proposiciones simples que componen el enunciado:
 - p : Estudio matemáticas.
 - q : Pierdo clases.
 - r : Realizo todas las tareas.
 - s : Aprobaré el examen.
- **Paso 2.** Reconocemos los conectores del lenguaje natural:
 - «y» se traduce como conjunción: $\wedge$
 - «no» se traduce como negación: $\neg$
 - «o» se traduce como disyunción: $\vee$
 - «sí..., entonces...» se traduce como implicación: $\rightarrow$
- **Paso 3.** Escribimos la fórmula compuesta respetando la estructura lógica:

$$[(p \wedge \neg q) \vee r] \rightarrow s.$$

Este ejemplo es más complejo y demuestra el uso de paréntesis anidados. El uso de los corchetes grandes no es obligatorio pero ayuda a leer la fórmula, dejando claro que la implicación principal tiene como antecedente todo el conjunto de la disyunción.

1.3.1. Jupyter Notebooks en el curso

Durante este curso, utilizaremos Jupyter Notebooks como herramienta para explorar conceptos de Matemática Discreta mediante programación en Python.

Un Jupyter Notebook es un entorno interactivo que permite escribir texto, insertar ecuaciones, ejecutar código y visualizar resultados en un mismo documento. Esta herramienta combina la claridad de una libreta de apuntes con el poder computacional de Python, facilitando el aprendizaje activo y la experimentación directa.

No necesitas instalar nada en tu computadora para utilizarlo: puedes acceder a todos los notebooks del curso directamente desde tu navegador utilizando Google Colab, una plataforma gratuita que permite ejecutar código Python en la nube.

Antes de la próxima clase, te invito a explorar el siguiente enlace. En él encontrarás una introducción amigable al uso de Jupyter Notebooks, herramienta que emplearemos para realizar actividades prácticas a lo largo del curso:

- **Título del enlace:** IA4LEGOS - Introducción práctica a Python con Jupyter
- **Descripción del enlace:** Explicación sobre qué es un Jupyter Notebook, cómo funciona y cómo se puede comenzar a usar Python mediante Google Colab.
- **Enlace:** <https://andres-merino.github.io/MatematicasDiscretas-Virtual-05-N0279/pages/asistentes/IA4LEGOS-Python.html>

Referencias

Augustus De Morgan. (s.f.). En Wikipedia. [Recuperado el 19 de julio de 2025].

https://es.wikipedia.org/wiki/Augustus_De_Morgan

Epp, S. S. (2020). Discrete Mathematics with Applications (5.a ed.). Cengage Learning.

George Boole. (s.f.). En Wikipedia. [Recuperado el 19 de julio de 2025].

https://es.wikipedia.org/wiki/George_Boole

Gottfried Leibniz. (s.f.). Gottfried Leibniz. EnWikipedia. [Recuperado el 19 de julio de

2025]. https://es.wikipedia.org/wiki/Gottfried_Leibniz

Johnsonbaugh, R. (2018). Discrete Mathematics (8.a ed.). Pearson Educación.

Lenzen, W. (2025). Leibniz: Logic [Recuperado el 19 de julio de 2025]. Internet Encyclopedia of Philosophy. <https://iep.utm.edu/leib-log/>

Términos

- **Conectores lógicos:** son símbolos que se usan para relacionar proposiciones entre sí.
- **Valor de verdad:** es el estado que indica si una proposición es verdadera o falsa.

Profundización

Recurso 1

- **Título del recurso:** Traducción del lenguaje natural a la lógica proposicional
- **Descripción del enlace:** Incluye guías para traducir conectores como «y», «o», «no», «si..., entonces...» y «si y solo si».
- **Enlace:** <https://andres-merino.github.io/MatematicasDiscretas-Virtual-05-N0279/pages/clases/clase01/traducciones.html>

Recurso 2

- **Título del recurso:** Ejercicios resueltos de traducción
- **Descripción del enlace:** Este recurso presenta una serie de ejercicios resueltos paso a paso sobre traducción de lenguaje natural a la lógica proposicional.
- **Enlace:** https://andres-merino.github.io/MatematicasDiscretas-Virtual-05-N0279/pages/clases/clase01/traducciones_ejercicios.html

Recurso 3

- **Título del recurso:** Ejercicios propuestos de traducción
- **Descripción del enlace:** Este recurso presenta una serie de ejercicios propuestos sobre traducción de lenguaje natural a la lógica proposicional.
- **Enlace:** https://andres-merino.github.io/MatematicasDiscretas-Virtual-05-N0279/pages/clases/clase01/ejercicios_propuestos.html



La excelencia no se improvisa

síguenos

