

Una proposición lógica es toda oración declarativa que pueda decirse si es verdadera o falsa.

Una proposición compuesta es la combinación de proposiciones simples mediante conectivos lógicos.

El conectivo lógico de negación invierte el valor de verdad de una proposición. Cuando aplicamos este conector a una proposición lógica verdadera, el resultado es falso. Cuando lo aplicamos a una proposición lógica falsa, el resultado es verdadero. Siempre cambia el valor.

p	$\neg p$
V	F
F	V

dadas dos proposiciones p y q, llamaremos conjunción de p y q a la proposición $p \wedge q$ (que se lee p y q), que es verdadera si ambas, p y q, lo son, o es falsa en todos los otros casos.

Tabla 6: Tabla de verdad de $p \wedge q$ resuelta

p	q	$p \wedge q$
V	V	V
F	F	F
V	F	F
F	V	F

Dadas dos proposiciones p y q, llamaremos disyunción de p y q a la proposición $p \vee q$ (que se lee p o q), que es falsa si ambas, p y q, lo son, o es verdadera en todos los otros casos.

Tabla 8: Tabla de verdad de $p \vee q$ resuelta

p	q	$p \vee q$
V	V	V
F	F	F
V	F	V
F	V	V

Ley de morgan:

- La negación de la conjunción es la disyunción de las negaciones.
- La negación de la disyunción es la conjunción de las negaciones.

Esto se interpreta de la siguiente manera. Dadas dos variables booleanas, A y B:

- $(A \text{ AND } B)'$ es igual a $A' \text{ OR } B'$.
- $(A \text{ OR } B)'$ es igual a $A' \text{ AND } B'$.

Operaciones de algebra booleana son: AND, NOT, OR

Hay mucho de preguntas de características

Características de los algoritmos

- Ordenados
- Finitos
- Definidos

Partes de un algoritmo

- Entrada
- Proceso

- **Salida**

Lenguajes de algoritmos

- **Cualitativos: se utilizan palabras**
- **Cuantitativos: se utilizan números**

Funciones de un algoritmo

- **Búsqueda**
- **Ordenamiento**
- **Encadenamiento**

Estrategia de un algoritmo

- **Determinista**
- **Probabilístico**
- **Veraz**

Pasos de una resolución

- 1. Identificación del problema**
- 2. Interpretación del problema. Análisis**
- 3. Abstracción**
- 4. Formular la estrategia de resolución**
- 5. Verificar la estrategia de resolución**
- 6. Ajustar la estrategia**

Lenguaje bajo nivel: lenguajes de más bajo nivel (lenguaje máquina).

Lenguaje alto nivel: son los que facilitan más la programación, contienen compilador que procesa los códigos, y los ejecuta a sistema binario.

Ciclo del desarrollo del software, fases:

- **Definición** de necesidades / Identificación del problema / Investigación preliminar
- **Análisis**/ Interpretación del problema / Determinación de requerimientos
- **Diseño** de la solución / Diseño del sistema
- **Codificación** / Desarrollo del Sistema
- **Prueba, depuración** y documentación
- **Implementación** / Validación / Implantación
- **Mantenimiento** / Evolución

Características que deben cumplir los programas

- Usabilidad
- Mantenibilidad
- Fiabilidad

¿Que son las condicionales?

Los condicionales son instrucciones que evalúan la veracidad de una sentencia, por lo tanto, utilizan valores booleanos (verdadero o falso) para decidir si ejecuta o no una determinada sección de código.

Algoritmos de ordenamiento

Los algoritmos de ordenamiento son un tipo especial de algoritmo que organiza los elementos de un arreglo, bajo un criterio específico

Por ejemplo, una complejidad $O(n)$ significa que es igual a la cantidad de elementos (n = cantidad de elementos por ordenar). Una complejidad de $O(2*n)$ significa que es el doble de la cantidad de elementos que ordenar, y una complejidad de $O(n^2)$ significa que es la cantidad de elementos al cuadrado.

1. Algoritmo 1: $O(n^2) = O(15^2) = O(225)$
2. Algoritmo 2: $O(n) = O(15)$
3. Algoritmo 3: $O(2*n) = O(2*15) = O(30)$

El orden de más simple a más complejo es: 2, 3, 1.

Compilador

Un compilador es una aplicación que lee un lenguaje de entrada, denominado fuente, y lo transforma en lenguaje objeto, el cual no tiene por qué ser necesariamente código máquina. Posteriormente, se usa un enlazador o *linker* para convertir los distintos ficheros objeto en un programa ejecutable.

Interpretado

Se considera un lenguaje interpretado a aquel que es traducido a instrucciones propias de la plataforma, línea por línea, mientras es ejecutado. Esto significa que la traducción y ejecución se hacen en el mismo momento.

Intérprete

Programa que realiza un análisis de una aplicación escrita en un lenguaje no-máquina (fácil de entender y trabajar con él) y lo convierte en lenguaje máquina, entendible por el ordenador. Es un procesador de lenguaje que, a diferencia de un compilador, analiza una línea de código y después realiza las acciones específicas, en lugar de producir una traducción al lenguaje de máquina para que se ejecute posteriormente.

Lenguajes estáticos

Un lenguaje estático es aquel que, si ya definimos una variable de un tipo determinado, no permite que cambiemos su tipo. Explicado de otra manera: si delimitamos una variable de un tipo específico, solamente podemos asignarle valores de ese tipo y solo de ese tipo.

Lenguajes dinámicos

Al contrario de los estáticos, los lenguajes dinámicos no tienen especificado el tipo explícitamente, lo infieren o deducen mediante el valor asignado a la variable en tiempo de ejecución. Otra diferencia es que una misma variable puede tener diferentes tipos de valores en diferentes momentos. Python es un ejemplo de este tipo de lenguaje.

¡Tipos de paradigmas (imperativo, declarativo, orientada a objetos)

Los paradigmas son un conjunto de reglas que permiten que la codificación sea más simple, ordenada y reutilizable. Existen diferentes paradigmas, entre los cuales, los más importantes son el imperativo, el declarativo y el orientado a objetos.

Paradigma imperativo

El paradigma imperativo se caracteriza por desarrollar **software** mediante un conjunto de instrucciones que indican cómo realizar la tarea.

Una subclasificación de este paradigma incluye los siguientes enfoques:

Programación estructurada: utiliza bucles anidados, condicionales y subrutinas.

Programación procedimental: utiliza funciones que son llamadas cada vez que se necesite ejecutar.

Programación modular: divide el desarrollo en módulos o subprogramas que se relacionan entre sí.

Paradigma declarativo

En el caso del paradigma declarativo, no definimos el conjunto de pasos o instrucciones, sino que determinamos directamente el estado final de lo que deseamos obtener y dejamos al computador que defina cuál es el mejor método para hacerlo.

Una subclasificación de este paradigma incluye los siguientes enfoques:

Programación lógica: es una especie de resolución tecnológica de *puzzles*, y tiene gran utilidad para resolver problemas en áreas concretas como los sistemas expertos, el procesamiento de lenguajes o la inteligencia artificial, entre otras.

Programación funcional: es un lenguaje de programación declarativo donde el programador especifica lo que quiere hacer, en lugar de lidiar con el estado de los objetos.

Paradigma orientado a objetos

El paradigma orientado a objetos establece que podemos modelar y codificar la solución de un problema identificando los objetos que participan en el proceso.

Algunos de los conceptos utilizados en este paradigma son:

Abstracción: La abstracción puede definirse como la extracción de las propiedades esenciales de un concepto. La abstracción nos ayuda a no enfocarnos en los detalles que no son importantes para lo que estamos realizando. Implica la identificación de los atributos y los métodos de un objeto. Podemos definirla también como la capacidad de encapsular y aislar la información que deseamos incluir en nuestros objetos.

Encapsulación: La información dentro de un objeto es privada a ese objeto y solo a ese objeto. La información es definida por el valor de sus atributos. Solo podemos acceder a los valores de los atributos (es decir, a la información) de un objeto mediante métodos, nunca directamente leyendo los atributos.

Pensemos en una casa, en su interior tiene luces. Si encapsulamos esta casa, estamos diciendo que solo las personas que están dentro pueden encender las luces.

Herencia: Podemos definir la herencia como la propiedad que permite a los objetos construirse a partir de otros objetos. Por ejemplo, las clases de animales se separan en mamíferos, anfibios, insectos, etcétera, las clases de vehículos se dividen en autos, camiones, colectivos, entre otros.

Polimorfismo: Esta propiedad establece que una misma operación puede

tener diferentes comportamientos en diferentes objetos, es decir, diferentes objetos pueden reaccionar de manera distinta al enviarles el mismo mensaje

Dentro de los principales conceptos de la programación orientada a objetos, podemos identificar los siguientes:

Clases

Objetos

Atributos

Métodos

Mensajes

Concepto	Definición
Objetos	Instancia de una clase, poseen atributos con sus valores y métodos.
Clases	Definen los atributos y métodos de un objeto. Se utilizan para crear nuevos objetos.
Atributos	Almacenan información de un objeto específico, determinan el estado del objeto.
Métodos	Ejecutan una acción determinada, de acuerdo con el estado del objeto; determinan el comportamiento del objeto.
Mensajes	Son peticiones que se envían de un objeto a otro para que este último ejecute una acción mediante uno de sus métodos.