

Unidad 1

Introducción a los Lenguajes Formales y a los Compiladores



Lenguajes Formales y de Programación
Inga. Zulma Aguirre – zaguirre@ingenieria.usac.edu.gt

Enero 2022



U1. Introducción: Lenguaje Natural

Lenguaje Natural



Es el medio que utilizamos de manera cotidiana para establecer comunicación con las demás personas.





U1. Introducción: Lenguaje Natural

Lenguaje Natural



Es el medio que utilizamos de manera cotidiana para establecer comunicación con las demás personas.

Se caracterizan por las siguientes propiedades:

1. Desarrollados por enriquecimiento progresivo antes de cualquier intento de formación de una teoría.
2. La importancia de su carácter expresivo debido grandemente a la riqueza del componente semántico (polisemia).
3. Dificultad o complejidad de una formalización completa.



U1. Introducción: Lenguaje Formal

Lenguaje Natural

Es el lenguaje que el hombre ha desarrollado para expresar las situaciones que se dan en específico en cada área del conocimiento científico.

Lenguaje Formal



Características:

1. Se desarrollan de una teoría preestablecida.
2. Componente semántico mínimo.
3. Posibilidad de incrementar el componente semántico de acuerdo con la teoría a formalizar.
4. La sintaxis produce oraciones no ambiguas.
5. Completa formalización, por esto, el potencial de la construcción computacional.



U1. Introducción: Lenguaje Formal

Lenguaje Natural

Lenguaje Formal

Lenguaje de
Programación

Un lenguaje de programación es un Lenguaje Formal usado para escribir instrucciones que pueden ser traducidas a lenguaje máquina y ejecutadas en una computadora.





Introducción: Evolución de los Lenguajes de Programación

Paradigmas

Imperativos

Describe CÓMO se debe realizar el programa, no se especifica el por qué se realizará de esa forma. Se basa en dar instrucciones a la computadora de cómo hacer las cosas en forma de algoritmo.

Declarativos

Describe QUÉ se debe realizar en el programa, no se especifica cómo se realizará. Se basa en describir el problema declarando propiedades y reglas que deben cumplirse, en lugar de instrucciones.



Introducción: Evolución de los Lenguajes de Programación

Avance a los lenguajes de alto nivel

Generaciones de los Lenguajes de Programación

Primera Generación

Segunda Generación

Tercera Generación

Cuarta Generación

Quinta Generación

Lenguaje Máquina

Lenguaje Ensamblador

Lenguajes Estructurados

Lenguajes Orientados a Objetos

Lenguajes de Inteligencia Artificial

Lenguajes de Bajo Nivel

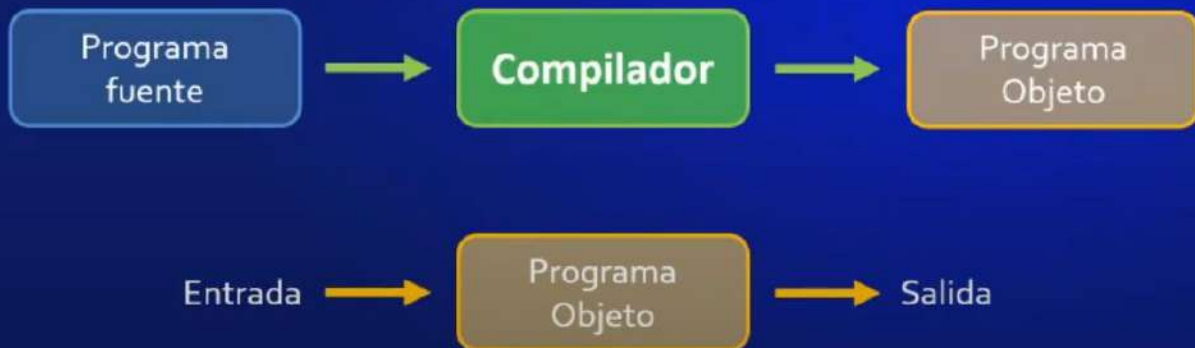
Lenguajes de Alto Nivel





Introducción: Procesadores de Lenguajes

Antes de que se pueda ejecutar un programa, primero debe ser traducido a una forma que se pueda ejecutar por una computadora. Los sistemas de software que realizan esta traducción se denominan **COMPILADORES**.





Introducción: Procesadores de Lenguajes

INTÉRPRETE es otro tipo de procesador de lenguajes. En lugar de producir un programa objeto o destino (como una traducción), un intérprete valida cada instrucción y la ejecuta tomando en cuenta las entradas proporcionadas por el usuario.





Introducción: Procesadores de Lenguajes

Procesadores de Lenguajes

Lenguajes Compilados

Un lenguaje compilado es aquel cuyo código fuente, escrito en un lenguaje de alto nivel, es traducido por un compilador a un archivo ejecutable entendible para la máquina. Se realiza una vez (siempre que no existan cambios en el código fuente).

Un lenguaje interpretado es aquel en el cual sus instrucciones[†] del código fuente, en un lenguaje de alto nivel, es traducido por el interprete a un lenguaje entendible para la máquina y se interpreta y ejecuta cada vez que se ejecuta el código, instrucción por instrucción.

Lenguajes Interpretados

Introducción: Evolución de los Lenguajes de Programación



Impactos del Compilador

- Los compiladores deben mejorar sus algoritmos y representaciones para traducir y respaldar las nuevas características de los lenguajes de programación.
- La arquitectura de las computadoras también ha evolucionado, donde los compiladores tienen que aprovechar al máximo las nuevas capacidades del hardware. Para que el rendimiento de las aplicaciones sean efectivas.
- Los compiladores pueden ayudar a promover el uso de lenguajes de alto nivel minimizando la sobrecarga de ejecución de los programas escritos en estos lenguajes.
- Las buenas técnicas de ingeniería de software son esenciales para crear y desarrollar procesadores de lenguajes modernos.