

Unidad 1: Evolución de los lenguajes de programación.

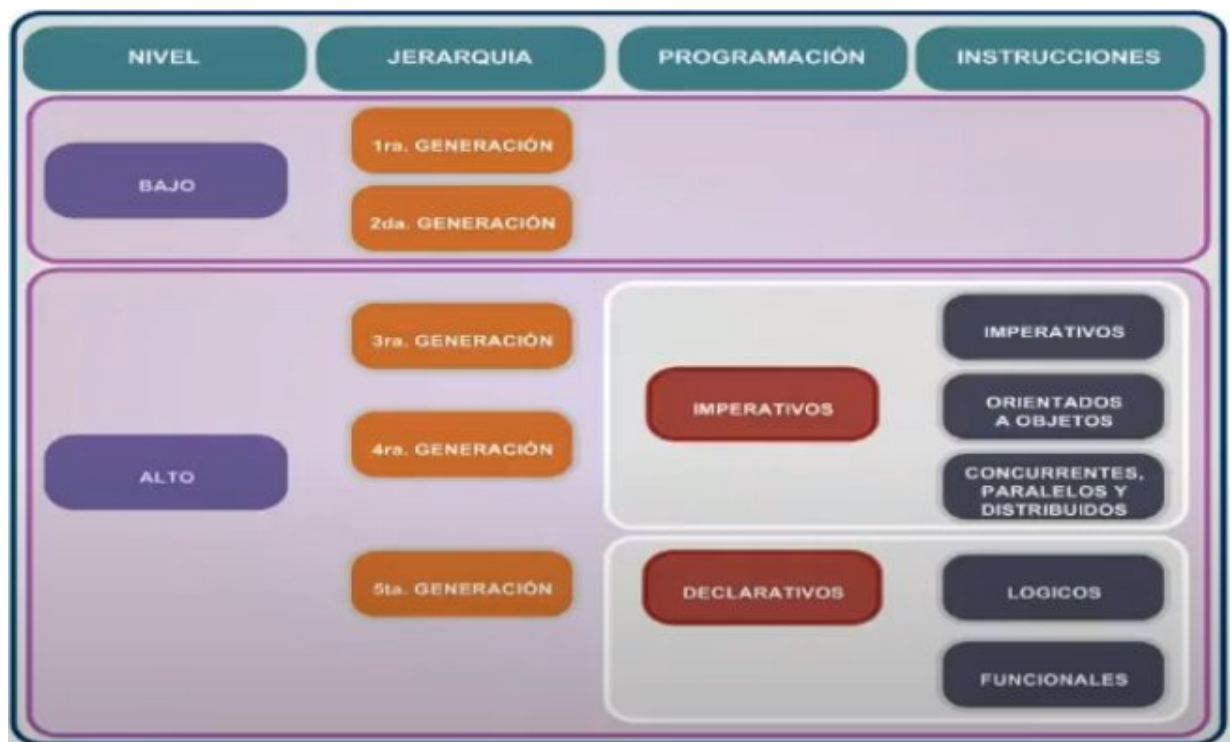
Definición de lenguaje de programación

Un lenguaje de programación es un sistema notacional para describir **computaciones/procesamientos** que puede ser realizado por una computadora en una forma **legible** tanto **para la maquina** como **para el ser humano**.

Legible para la maquina se dice cuando su estructura es adecuadamente simple tal que permite una traducción eficiente.

Legible para el ser humano se dice cuando dicho lenguaje proporcione abstracciones de todas aquellas operaciones llevadas a cabo por la maquina de forma que sean entendibles por el ser humano. Cuanto mas se acerca al lenguaje natural, mas legible es para el programador.

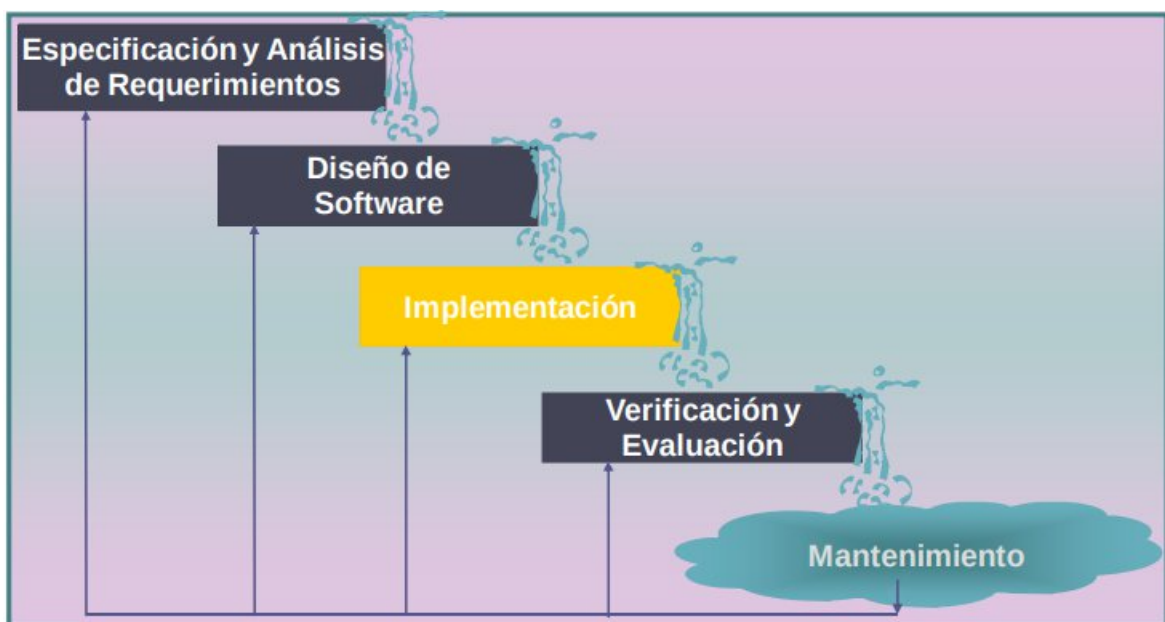
Clasificación:



- Lenguajes de bajo nivel:
 - o Directamente entendibles por la maquina.
 - o Jerarquía:
 - § 1ra. Generación: aquí tenemos al lenguaje maquina.
 - § 2da. Generación: aquí ubicamos a Assembler.
 - o Ej. Assembler o lenguaje maquina.
- Lenguajes de alto nivel:
 - o Permiten escribir sentencias con frases muy parecidas al lenguaje natural.
 - o Jerarquía:
 - § 3ra. Generación: lenguajes con habilidades procedurales, como Pascal o C.

- § 4ta. Generación: lenguajes para manipulación de bases de datos como PL/SQL de Oracle y lenguajes que proporcionan un entorno de programación visual.
- § 5ta. Generación: basados en lógica o matemáticas, utilizados para la inteligencia artificial como Miranda o ML.
- o Según la forma de programar tenemos:
 - § Lenguajes imperativos: ejecución paso a paso, cumpliendo ordenes que indican como se va a resolver/realizar el problema.
 - § Lenguajes declarativos: prevalece el que se va a hacer sin indicar como hacerlo.
- o Según las instrucciones que utilizan:
 - § Imperativos: uso de sentencias/ordenes que se le dan a la maquina para que realice una acción.
 - § Orientados a objetos: incorporan el concepto de clase, herencia, polimorfismo.
 - § Concurrentes, paralelos y distribuidos: proveen ciertas instrucciones que permiten sincronizar y paso de mensajes, necesario para este tipo de instrucciones.
 - § Lógicos: definición de predicados, especificar la solución.
 - § Funcionales: utilizan funciones que producen respuestas.
- o Ej. Pascal, C, Java, etc.

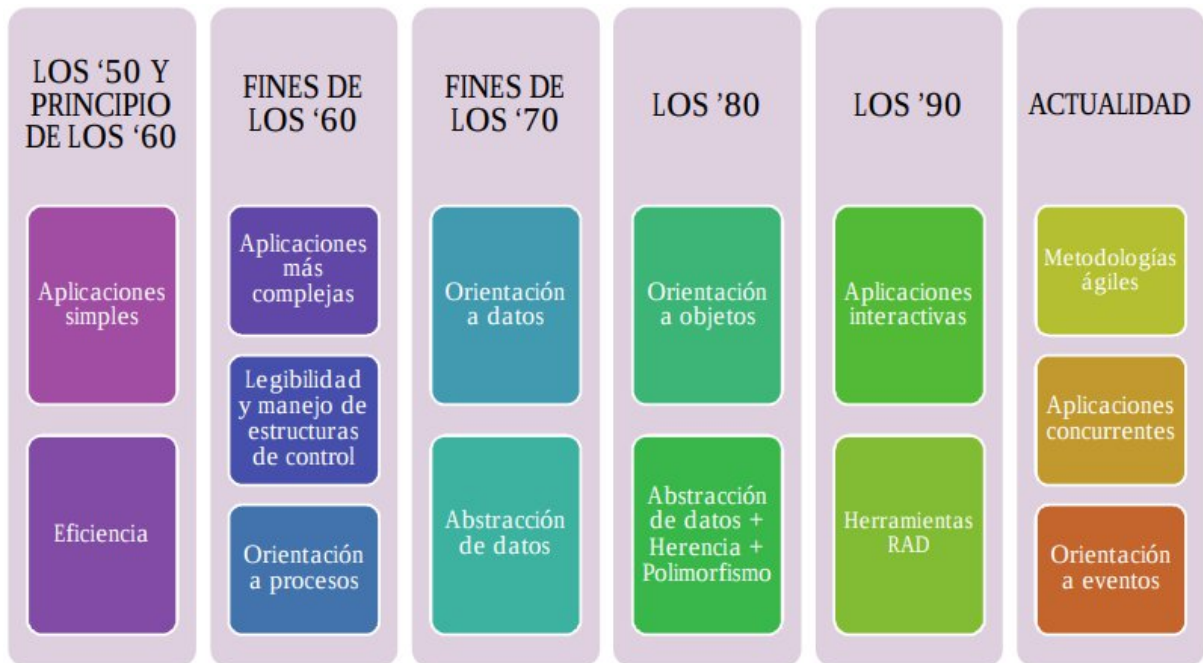
Un lenguaje de programación es una herramienta que nos permite desarrollar software, para desarrollarlo de una manera confiable y eficiente es necesario que se aborde de una manera sistemática.



- Especificación y Análisis de Requerimientos: se analiza cual es el problema a resolver, que es lo que debe realizar el sistema. Finalizada esta etapa se produce un documento de especificación de requerimientos.
- Diseño de software: se determina como se va a resolver el problema. Esta etapa da como resultado el documento de especificación de diseño. La metodología de diseño a utilizar va a afectar la elección del lenguaje de programación, es aquí donde se lo elije.

- Implementación: se elige la codificación que se ajusta al diseño del software, utilizando el lenguaje de programación elegido en la etapa previa para programar todos los módulos del sistema.
- Verificación y Evaluación: se realiza el testeo de los módulos y la integración entre ellos.
- Mantenimiento: se producen correcciones en el sistema ante un funcionamiento defectuoso o se añaden mejoras en la adaptación. Se dice que el **costo de mantenimiento** en relación a los costos de etapas anteriores es mayor en esta etapa.

Los lenguajes de programación surgen de la vinculación entre las distintas metodologías de diseño de software y también de la arquitectura de la computadora.

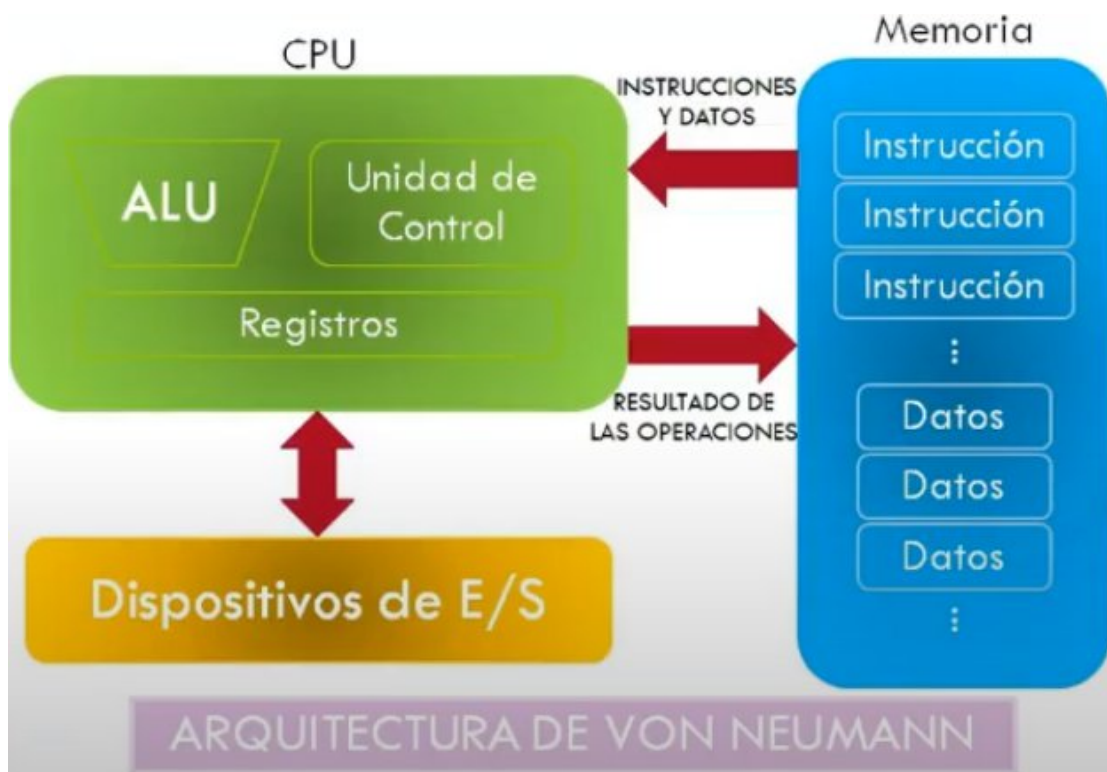


- Los '50 y principio de los '60:
 - o Lenguajes de programación no diseñados para seguir una metodología en particular.
 - o Aplicaciones simples.
 - o Se apuntaba a la eficiencia debido a los recursos de la época
- Fines de los '60:
 - o Aplicaciones mas complejas.
 - o Énfasis en la legibilidad y manejo de estructuras de control.
 - o Programación estructurada
 - o Diseño top down.
 - o Técnica de refinamiento sucesivo.
 - o Orientación a procesos.
- Fines de los '70
 - o Metodologías de orientación a procedimientos se transforman a metodologías de orientación a datos.
 - o Diseño de los datos utilizando el concepto de **tipo de dato abstracto** para la resolución de problemas.
 - o El lenguaje debe proveer las mecanismos para implementar estas **abstracciones de datos**.
- Los '80
 - o Surge la orientación a objetos.

- o Abstracción de datos + herencia + polimorfismo.
- Los '90
 - o Surgen las herramientas para automatizar y hacer mas rápida la producción de software.
 - o Aplicaciones interactivas.
- Actualidad
 - o Surgen metodologías ágiles como SCRUM o XP.
 - o Aplicaciones concurrentes.
 - o Orientación a eventos.

Las metodologías influyen en los lenguajes de programación dado que establecen requisitos para apoyar el proceso de desarrollo de software.

La arquitectura de la computadora restringe el diseño del lenguaje de programación para que se implementen de manera eficiente en las maquinas actuales.



Los lenguajes de programación convencionales como los imperativos o los orientados a objetos pueden verse como abstracciones de una arquitectura subyacente de Von Neumann.

Los desarrolladores de los nuevos lenguajes intentar elevar el nivel de abstracción del mismo incorporando procedimiento, funciones, tipos de datos, clases, manejadores de excepciones para que sean mas fáciles de utilizar por los usuarios pero con las restricciones impuestas por la arquitectura de la computadora, ya que todo programa que hagamos termina ejecutandose en una computadora y tiene que restringir su ejecución a lo que es la arquitectura de Von Neumann.



Ambientes de desarrollo

- **Conjunto integrado de herramientas y técnicas** que son utilizadas en todas las etapas del ciclo de vida del producto desarrollado. Facilitan el desarrollo de los programas. Cada herramienta es un programa en particular que se puede utilizar como ayuda durante varias etapas en el proceso de desarrollo, como por ejemplo en la compilación o en la depuración.
- Editores, depuradores, etc. Facilitan la tarea del programador. Netbeans o Visual Studio son ejemplos de editores (Entornos de programación).
- Los ambientes de desarrollo han influido en los diseños de los lenguajes de programación, en relación con los conceptos de compilación separada y ensamblado de un sistema a partir de sus distintos componentes, como también en aquellas características que influyen en la puesta a prueba.
- Compilación separada requiere un lenguaje estructurado de tal forma que los subprogramas hechos por distintos programadores se puedan ejecutar, compilar por separado y ensamblarse en el programa completo.

Dominios de aplicación

- Las computadoras se utilizan en gran cantidad de áreas, cada una de las áreas o **dominios de aplicación** posee ciertas particularidades, por lo que se han desarrollado distintos lenguajes con distintas características y objetivos para cada uno de los distintos dominios.
- Veremos 5 dominios:
 - o Aplicaciones científicas:
 - § El desarrollo de estas surgió en los primeros años de la historia de la computación, los primeros programas se escribieron durante la segunda guerra mundial para predecir la trayectoria de misiles, por ejemplo, usando una fórmula física.

- § Cálculos complejos se deben realizar de manera rápida, precisa, y eficiente.
- § Números de punto flotante.
- § Lenguajes de programación proveen estructuras tipo arreglos y matrices.
- § Fortran y MathLab son representativos de este dominio.
- o Aplicaciones de negocio:
 - § Cuando se vio que la computación podía avanzar a distintas áreas comienza a utilizarse a en aplicaciones relacionadas a la bancarizacion por ejemplo.
 - § Los programas diseñados se utilizan para la gestión de los sistemas de información de esas entidades.
 - § Sistemas mas numerosos que existen mundialmente.
 - § Sistemas de contabilidad, marketing, inventario, producción, ventas online, etc.
 - § Lenguajes que conforman este dominio se caracterizan por el uso de numero decimales, caracteres y de archivos.
 - § Lenguajes imperativos como Cobol, herramientas declarativas como RPG(generatoras de reportes), SQL(lenguaje de consulta) son de los mas representativos dentro de este dominio.
- o Inteligencia artificial:
 - § Programas que imitan el comportamiento de la inteligencia humana, deducción lógica, la cognición.
 - § Programación funcional como lógica.
 - § Lenguajes como Lisp, ML, Prolog.
- o Programación de Sistemas:
 - § Toda aquella programación de software de base que se utiliza cuando se ejecutan los sistemas como por ejemplo el sistema operativo, software de red, compilador, etc.
 - § Dominio estrictamente relacionado con la arquitectura de los distintos equipos.
 - § Lenguajes mas representativos son C, C++, Assembler, etc.
- o Aplicaciones Web:
 - § Área mas dinámica en el crecimiento y calidad junto con la inteligencia artificial.
 - § Utilizado en aplicaciones de comercio electrónicos, aplicaciones académicas, industriales, gubernamentales.
 - § Modelo interactivos, prevalece la programación guiada por eventos y programación orientada a objetos.
 - § Lenguajes de marcado como HTML, lenguajes de scripting como PHP, JS y de propósito general como Java.

La Abstracción

- Permite dar legibilidad a los programas.
- Todos los mecanismos de abstracción se diseñan para que los programadores puedan entenderlos.

La abstracción y los LP

10

ABSTRACCIONES DE DATOS

- Básicas.
 - Tipos primitivos.
 - Variables.
- Estructuradas.
 - Tipos de datos estructurados.
- Unitarias.
 - Encapsulamiento y ocultamiento.
 - Reutilización.

ABSTRACCIONES DE CONTROL

- Básicas.
 - Enunciados y sentencias de asignación.
 - Transferencia incondicional.
- Estructuradas.
 - Sentencias de selección condicionales.
 - Procedimientos y funciones.
- Unitarias.
 - Reutilización.



Lenguajes de Programación

marzo de 2021

- Dos tipos:
 - o Abstracciones de datos:
 - § Resumen las propiedades de los datos, de los números, de las cadenas de caracteres, etc.
 - § Según cantidad de datos:
 - Básicas:
 - o Abstraen la representación interna de valores de datos comunes.
 - o Ej. enteros almacenados en complemento a 2 y operaciones estándares como suma y multiplicación.
 - o Variables abstraen una dirección de memoria por medio de un nombre.
 - Estructuradas:
 - o Método principal para la abstracción de un conjunto de valores de datos relacionados entre si.
 - o Ej. Registros o arreglos.
 - o Una declaración de tipo define nuevas estructuras que son nuevos tipos de datos.
 - Unitarias:
 - o Surge de la necesidad de reunir código relacionado entre si y datos utilizando encapsulado de datos y ocultamiento de información.
 - o Ej. Módulos en ML. o paquetes en Java.
 - o Reutilización de abstracciones en diferentes programas, generalmente representan componentes o contenedores.

- o Abstracciones de control:
 - § Resumen las propiedades de las transferencias de control, de las sentencias condicionales, bucles, llamadas a procedimientos.
 - § Según cantidad de datos:
 - Básicas:
 - o Enunciados o sentencias que combinan varias instrucciones de maquina en una sentencia abstracta mas sencilla.
 - o Ej. Instrucción goto (transferencia incondicional).
 - Estructuradas:
 - o Dividen el programa en grupo de instrucciones que pueden estar anidadas.
 - o Ej. Instrucciones de selección(if,case,etc), repetición(for, while, etc), funciones y procedimientos.
 - Unitarias:
 - o Incluyen conjunto de procedimientos que pueden ser utilizados por otra parte del programa.
 - o Permiten entender la totalidad del programa sin obtener detalles de los servicios que brinda
- Se agrupan según la cantidad de información que contienen:
 - o Básicas: Información localizada de la maquina.
 - o Estructuradas: Información global de la estructura de un programa
 - o Unitarias: Información sobre una parte completa de un programa.

Perspectiva Histórica



- El desarrollo de software originalmente consistía solo en la implementación (codificación). La computadora era utilizada principalmente en aplicaciones científicas. Una aplicación era programada por una sola persona. Problema a resolver claro. Sin

especificación de diseño, análisis de requerimientos, mantenimiento. Lenguaje de programación solo necesitaba dar soporte al programador quien estaba programando lo que para los estándares de hoy en día sería una app simple.

- Con el paso del tiempo, surgió la necesidad de armar equipos de programadores por disciplinas y dar un enfoque más formal al desarrollo de software. A la fase de especificación de requerimientos y de diseño, también fue necesario incorporar un equipo de personas, con el consiguiente problema de comunicación. Cuestiones económicas obligaron a reforzar los sistemas existentes por lo que el mantenimiento resultó ser un problema importante.
- Posteriormente, los sistemas comenzaron a utilizarse en áreas críticas, la confiabilidad de los sistemas fue ganando importancia. Verificación y validación de los programas son de suma importancia para este tipo de sistemas.
- Paralelo a esto se produjo un salto exponencial en el hardware, lo cual no fue acompañado por el desarrollo de software (**Crisis del Software**). Dio como resultado la aparición de las primeras metodologías y de la Ing. De Software. Pautas a seguir para el desarrollo de software para optimizar el proceso y maximizar la calidad del resultado.
- Lenguajes de programación comienzan a agruparse bajo distintas reglas, patrones, estilos de programación dando origen a **Paradigmas de Programación**.