

# IS-513 Lenguajes de Programación

---

Msc, MBA. Harold Coello



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# I Unidad – Intro. Lenguajes de programación.

---

- Conceptos y definiciones básicas
- Tipos de lenguajes de programación
- Paradigmas de lenguajes de programación
- Introducción al lenguaje de programación PHP
- Introducción al framework VueJs



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Conceptos y definiciones básicas

---

El objetivo principal de los lenguajes de programación es facilitar al usuario las actividades que realiza día con día; por tal razón como programadores es importante conocer los conceptos básicos de programación, los tipos de lenguaje que se utilizan para el desarrollo y su funcionamiento para la interpretación de algoritmos, así como para dar solución a los problemas que se pueden resolver con base en software.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Conceptos y definiciones básicas

---

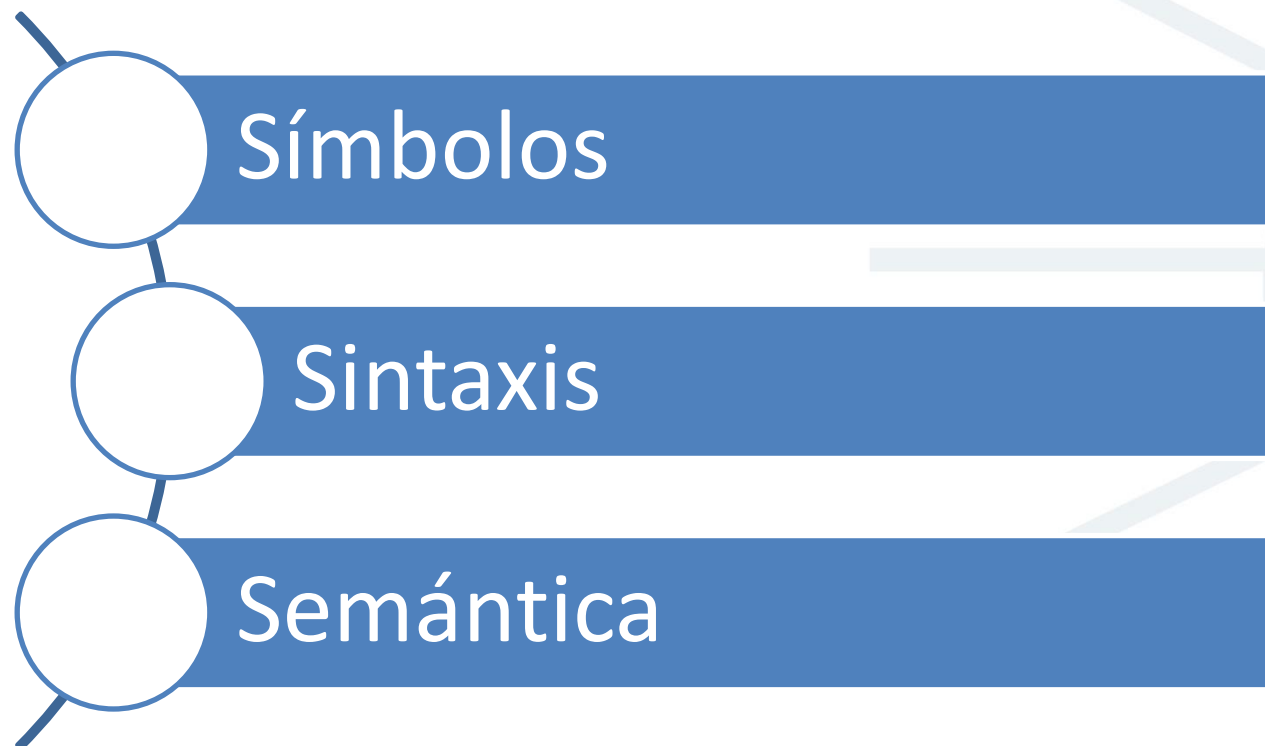
**Lenguaje de programación** es una herramienta que permite desarrollar software o programas para computadora. Estos son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Conceptos y definiciones básicas

---



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Conceptos y definiciones básicas

---

**La Programación** es el proceso de análisis, diseño, implementación, prueba y depuración de un algoritmo, a partir de un lenguaje que compila y genera un código fuente ejecutado en una computadora.

La función principal de un lenguaje de programación es escribir programas que permiten la comunicación usuario-máquina. Unos programas especiales (compiladores o intérpretes) convierten las instrucciones escritas en código fuente, en instrucciones escritas en lenguaje máquina (0 y 1).



# Conceptos y definiciones básicas

---

## Interprete

Leen las instrucciones líneas por línea y obtienen el código máquina correspondiente.

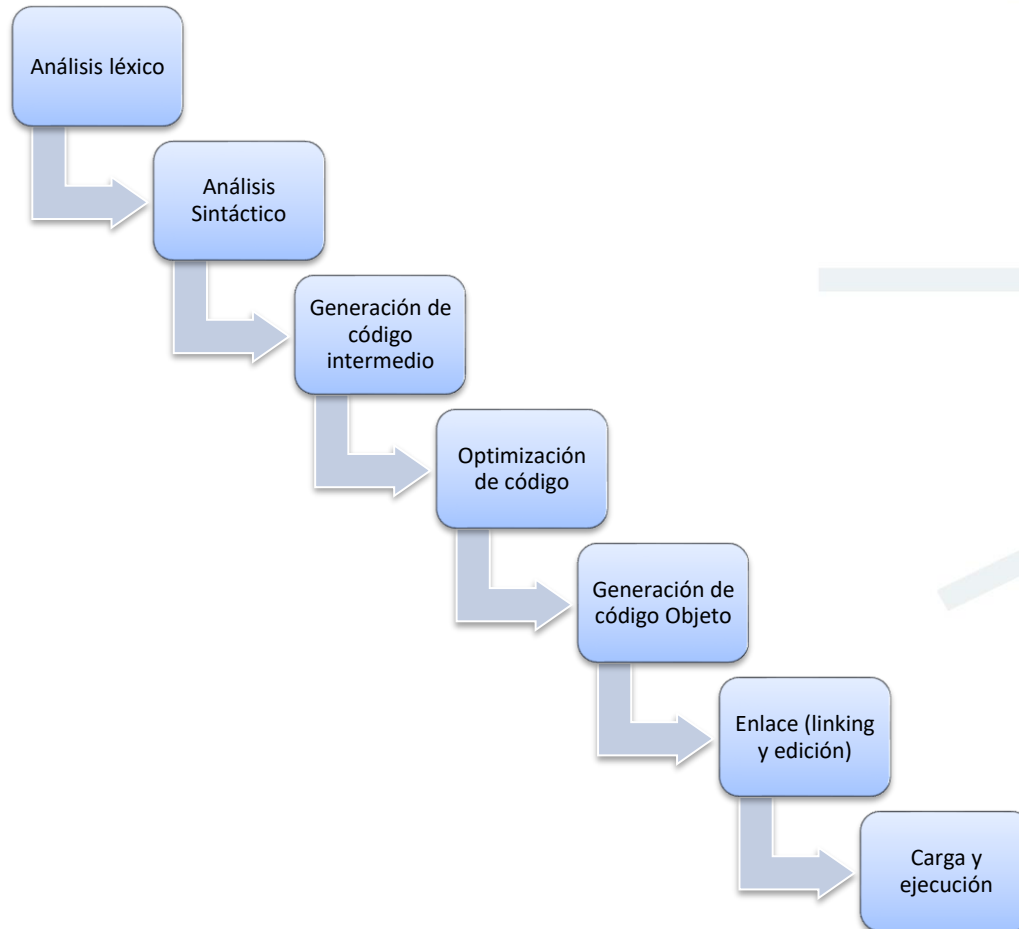
## Compilador

Traducen los símbolos de un lenguaje de programación a su equivalencia escrito en lenguaje de máquina (proceso conocido como compilar), por último se obtiene un programa ejecutable.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS



# Proceso de compilación

---

## Análisis Léxico

- Escaneo del código fuente para reconocer palabras clave, identificadores, constantes, operadores, etc.
- Creación de una secuencia de tokens



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación

---

## Análisis Sintáctico

- Utilización de reglas gramaticales para analizar como los tokens se combinan para formar declaraciones y estructuras más complejas.
- Creación de un árbol de sintaxis abstracta que representa la estructura del programa.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación

---

## Generación del código intermedio

- Traducción de la estructura del programa (representada en el árbol de sintaxis abstracta) a un formato intermedio.
- Esta etapa facilita la optimización antes de generar el código final.



# Proceso de compilación

---

## Optimización de código

- Aplicación de técnicas para reducir el tiempo de ejecución, minimizar el uso de recursos y mejorar la legibilidad del código intermedio.
- Optimización local y global del código.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación

---

## Generación de código objeto

- Transformación del código intermedio en instrucciones de código máquina.
- Creación de un archivo ejecutable o biblioteca que puede ser cargado y ejecutado por la máquina.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación

---

## Enlace (Linking) y Edición.

- Enlace de módulos para formar un programa ejecutable completo.
- Resolución de direcciones y referencias simbólicas.
- Creación del archivo ejecutable final.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de compilación

---

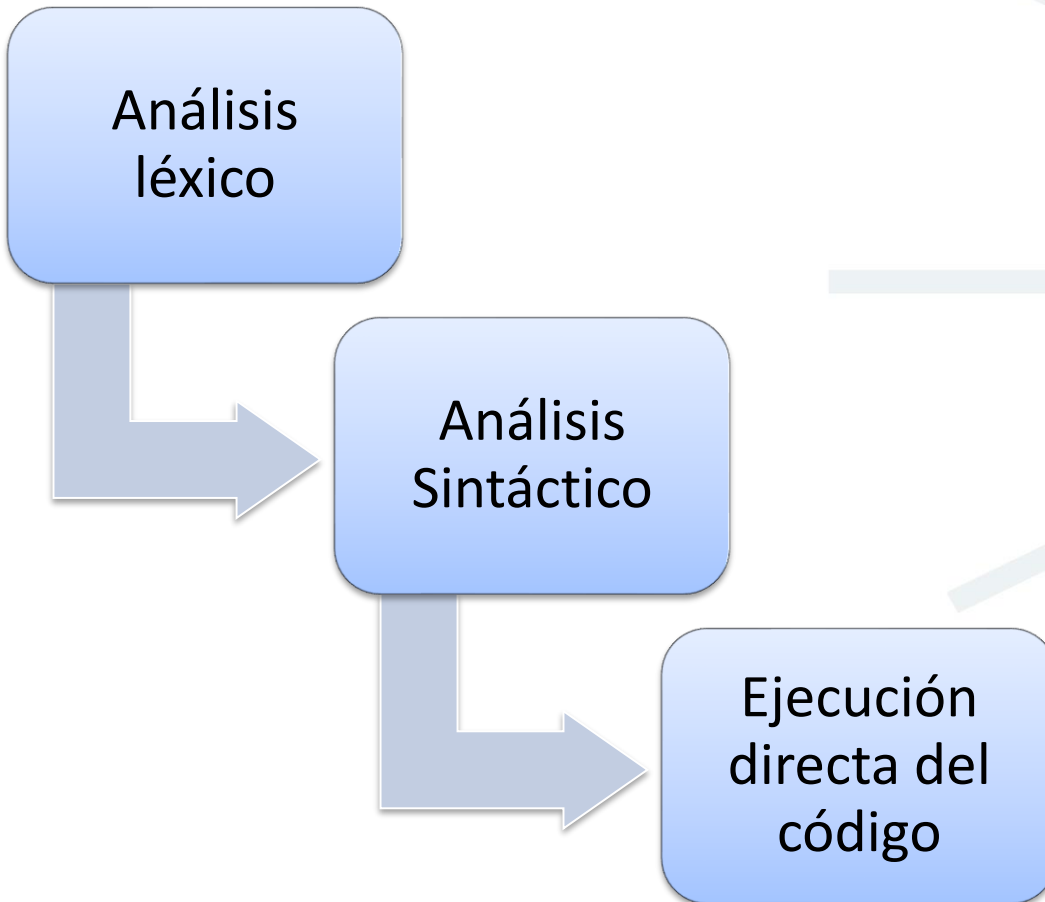
## Carga y ejecución

- Carga del programa en memoria.
- Inicio de la ejecución del programa por parte de la máquina.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de Interpretación



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS



# Proceso de Interpretación

---

## Análisis léxico y sintáctico

- Es similar a la compilación, el código fuente se analiza para identificar tokens y construir la estructura sintáctica del programa.
- No se genera un archivo ejecutable previo, en su lugar, el análisis se realiza a medida que se ejecuta cada línea de código.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Proceso de Interpretación

---

## Ejecución directa

- A diferencia de la compilación, donde se genera un código ejecutable separado, en la interpretación, el código fuente se ejecuta directamente.
- Las instrucciones se traducen y ejecutan una por una en tiempo real.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Tipos de lenguajes de programación

---

Los circuitos micro programables son sistemas digitales, lo que significa que trabaja con dos únicos niveles de tensión simbolizados con el cero (0) y el uno (1). Por eso el lenguaje de máquina utiliza solo dichos signos.

Un lenguaje de bajo nivel es trasladado fácilmente a lenguaje de máquina.

Los lenguajes de programación de alto nivel se caracterizan por expresar los programas de una manera sencilla.



# Tipos de lenguajes de programación

## Lenguaje máquina

- Interpretado directamente por un circuito micro programable.
- Instrucciones que determinan acciones que serán realizadas por la máquina.
- Es específico de cada máquina o arquitectura de la máquina.

## Lenguaje bajo nivel

- Proporciona poca o ninguna abstracción del micro procesador de una computadora.
- Su traslado a lenguaje máquina es fácil.

## Lenguaje de alto nivel

- Estructura semántica es muy similar a la forma como escriben los humanos.



# Lenguajes de máquina

## Instrucciones para una arquitectura hipotética:

0000: LOAD A, 100 ; Cargar el valor en la posición de memoria 100 en el registro A

0001: ADD A, B ; Sumar el contenido de los registros A y B y almacenar en A

0002: STORE 200, A ; Almacenar el contenido de A en la posición de memoria 200

0003: HALT ; Detener la ejecución



# Lenguajes de máquina

## Ejemplo de instrucciones MIPS:

add \$t0, \$t1, \$t2 ; Sumar el contenido de \$t1 y \$t2 y almacenar en \$t0

sub \$s0, \$s1, \$s2 ; Restar el contenido de \$s2 de \$s1 y almacenar en \$s0

lw \$t3, 100(\$s4) ; Cargar el valor en la posición de memoria 100 + contenido de \$s4 en \$t3

sw \$t3, 200(\$s5) ; Almacenar el contenido de \$t3 en la posición de memoria 200 + contenido de \$s5



# Lenguajes de bajo nivel

---

## Ensamblador:

- Utiliza mnemónicos para representar instrucciones de máquina.
- Cada instrucción de ensamblador se traduce directamente en una instrucción de máquina



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Lenguajes de bajo nivel - Ensamblador

## Ejemplo arquitectura x86:

MOV AX, 5 ; Mover el valor 5 al registro AX

ADD BX, AX ; Sumar el contenido de los registros BX y AX, almacenar en BX

JMP etiqueta ; Saltar a la etiqueta  
etiqueta:





# Lenguajes de programación de alto nivel

---

En esta categoría se encuentran los más utilizados dado que ofrecen un alto nivel de abstracción y están diseñados para ser más comprensibles y accesibles.

- Python
- Java
- JavaScript
- C#
- PHP



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Historia y evolución de los lenguajes de programación

---

- **Ada Lovelace:** Considerada la primera programadora. Creó el primer algoritmo informático mientras estudiaba la máquina de Babbage. Su visión acerca de las máquinas era que estas podían desarrollar operaciones más complejas que los simples cálculos numéricos.
- **Alan Turing:** Matemático e informático teórico que revolucionó la computación con la máquina de Turing, esta demostraba que con un algoritmo podían resolver cualquier problema informático.

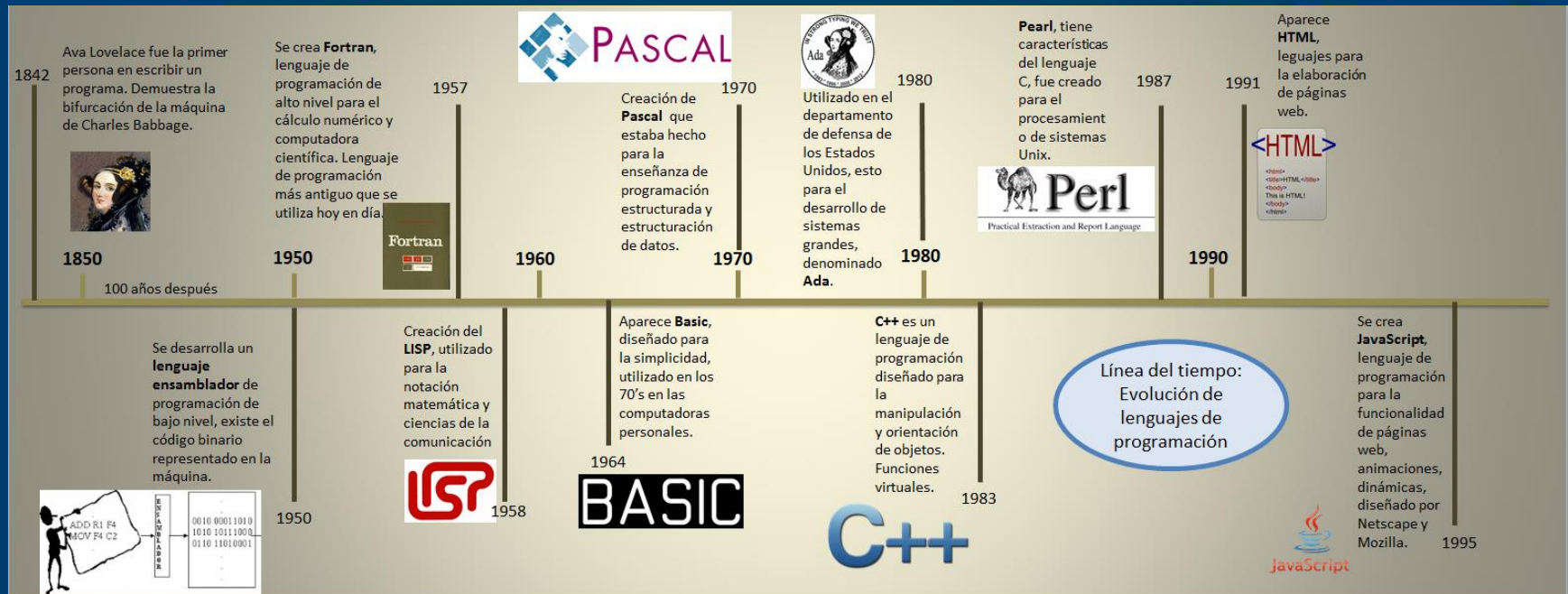


# Historia y evolución de los lenguajes de programación



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Evolución de los lenguajes de programación



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Componentes de lenguaje de programación

---

Los lenguajes de programación siguen siendo un conjunto de símbolos con una estructura gramatical, reglas de semántica y de sintaxis. Así mismo estos tienen una serie de componentes:

- **Tipos y estructuras de datos:** Elementos que permiten de manipular de forma eficiente las variables.
- **Instrucciones:** Estructuras gramaticales predefinidas, sirven para generar secuencias de acciones que conformen un programa.



# Componentes de lenguaje de programación

---

- **Control de flujo o estructuras de control:** Secuencia de acciones de un programa. Estas se clasifican en tres:
  - 1. Secuenciales
  - 2. Condicionales
  - 3. Cíclicas



# Componentes de lenguaje de programación

- **Estructuras secuenciales:** Permiten ejecutar una serie de instrucciones en un orden específico, de manera secuencial, es decir, una después de la otra.

Secuencias de instrucciones

instruccion1

instrucción2

instrucción3



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS



# Componentes de lenguaje de programación

- Estructuras secuenciales

Bloques de código

{

Instrucción1;

Instrucción2;

Instrucción3;

}



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS



# Componentes de lenguaje de programación

- **Estructuras condicionales:** Permiten que un programa tome decisiones basadas en la evaluación de condiciones. Estas estructuras controlan el flujo y ejecutan diferentes bloques de código según se cumplan ciertas condiciones.

Estructura If

Estructura If else

Estructura if else if

Operador ternario



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Componentes de lenguaje de programación

---

- **Estructuras cíclicas:** También conocidas como bucles o loops, permiten que un conjunto de instrucciones se ejecute repetidamente mientras se cumpla una condición.

Bucle while

Bucle for

Bucle do while

Bucle foreach



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Componentes de lenguaje de programación

- **Estructuras condicionales:** Permiten que un programa tome decisiones basadas en la evaluación de condiciones. Estas estructuras controlan el flujo y ejecutan diferentes bloques de código según se cumplan ciertas condiciones.

Estructura If

Estructura If else

Estructura if else if

Operador ternario



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Componentes de lenguaje de programación

---

## Gestión de la memoria

Este es un concepto crítico del desarrollo de software y varía según el tipo de lenguaje de programación. Dado que impacta directamente en la eficiencia y ejecución del programa.

La gestión de estos en algunos casos es directamente por el programador y en otros casos es gestionado propiamente por el lenguaje de programación.



# Componentes de lenguaje de programación

## Gestión de memoria

### Lenguajes de bajo nivel:

- En lenguajes como C, C++ los programadores tenemos control directo sobre la asignación y liberación de memoria.
- Se utilizan funciones como malloc y free para asignar y liberar memoria dinámicamente.

```
int *arr = (int *)malloc(5 * sizeof(int));  
free(arr);
```



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

# Componentes de lenguaje de programación

## Gestión de memoria

### Lenguajes de alto nivel:

- Este tipo de lenguajes gestionan automáticamente la asignación y liberación de memoria a través de un recolector de basura (GC).
- El GC identifica y libera automáticamente la memoria no utilizada.



**UNAH**  
UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS