



**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**  
**Modalidad de estudios En Línea**  
**Unidad de Organización Curricular: Básica**

**ESTUDIANTE:**

Montaño Rivera Ronny Duval

**TEMA:**

Actividad de Aprendizaje Individual Viaje a través de la Evolución de los Lenguajes de Programación

**FECHA:**

02 de junio de 2024

**MATERIA:**

Programación Orientada a Objetos

**DOCENTE:**

Mgs. Núñez Zamora Walter Rodrigo

**AÑO LECTIVO:**

2024

 [www.uea.edu.ec](http://www.uea.edu.ec)

 Km. 2. 1/2 vía Puyo a Tena (Paso Lateral)

 032892-118 / 032892-188 032892-098 / 032896-188 032896-476

**#UEAesExcelencia**



## 1. INTRODUCCIÓN

La evolución de los lenguajes de programación ha sido un viaje fascinante que ha moldeado el mundo de la informática y ha transformado la manera en que interactuamos con la tecnología. Desde los primeros lenguajes diseñados para simplificar cálculos complejos hasta los lenguajes modernos que facilitan el desarrollo de aplicaciones web, cada avance ha marcado un hito en la historia de la programación.

En este informe, exploraremos la evolución de tres lenguajes de programación emblemáticos: Fortran, C y Python. Cada uno de estos lenguajes ha dejado una huella indeleble en el desarrollo del software y ha influido en la adopción y evolución de la Programación Orientada a Objetos (POO). Analizaremos el origen, los hitos clave y la influencia de cada lenguaje en la programación moderna, y compararemos sus características distintivas para comprender mejor su impacto y relevancia en la actualidad.

A través de esta exploración, buscaremos no solo entender cómo estos lenguajes han evolucionado, sino también reflexionar sobre cómo este conocimiento puede enriquecer nuestras habilidades como programadores y mejorar nuestro enfoque hacia la POO. Esta perspectiva histórica nos permitirá apreciar la riqueza y diversidad del panorama de la programación y nos preparará para enfrentar los desafíos del desarrollo de software con una visión más informada y holística.



## 2. DESARROLLO

### 2.1. Fortran

- **Origen:** Fortran, abreviatura de "Formula Translation", fue desarrollado en 1957 por IBM. Es considerado el primer lenguaje de programación de alto nivel.
- **Hitos Clave:**
  - 1957: Lanzamiento de Fortran I, diseñado para cálculos científicos y de ingeniería.
  - 1966: Fortran 66, primera versión estándar.
  - 1978: Fortran 77, con mejoras en control de flujo y manejo de estructuras de datos.
  - 1991: Fortran 90, que introdujo conceptos de programación moderna como arrays dinámicos y la Programación Orientada a Objetos (POO).
- **Influencia:** Pionero en la automatización de cálculos numéricos y científicos, estableciendo la base para muchos lenguajes posteriores.

```
PROGRAM SumaEnteros
  IMPLICIT NONE
  INTEGER :: i, suma

  suma = 0
  DO i = 1, 10
    suma = suma + i
  END DO

  PRINT *, "La suma de los primeros 10 números enteros es:", suma
END PROGRAM SumaEnteros
```

*Imagen 1: Código Lenguaje Fortran*



Puedes saber que el programa está hecho en Fortran por varias razones específicas del lenguaje:

## 1. Sintaxis y Estructura:

- La palabra clave **PROGRAM** seguida del nombre del programa (**SumaEnteros** en este caso) es una convención en Fortran.
- El uso de **IMPLICIT NONE** es característico de Fortran, que requiere que todas las variables sean declaradas explícitamente.
- La declaración de variables con el tipo seguido de **::**, como **INTEGER :: i, suma**.
- El bucle **DO i = 1, 10 ... END DO** es una construcción típica de Fortran.
- **PRINT \*** se utiliza para imprimir salida en Fortran.

## 2. Declaración y Tipos de Datos:

- Fortran utiliza **INTEGER, REAL, DOUBLE PRECISION, COMPLEX**, entre otros, para declarar tipos de datos, lo cual es diferente de otros lenguajes de programación.

## 3. Comentarios:

- Aunque no hay comentarios en el ejemplo proporcionado, en Fortran, los comentarios se indican con un **“!”** al comienzo de la línea.

```
! Imprime el resultado
PRINT *, "La suma de los primeros 10 números enteros es:", suma
END PROGRAM SumaEnteros
```

*Imagen 2: Ejemplo de un comentario en Fortran:*



## 2.2. C

- **Origen:** Desarrollado por Dennis Ritchie en 1972 en los Laboratorios Bell, inicialmente para escribir el sistema operativo Unix.
- **Hitos Clave:**
  - 1972: Creación de C, que proporcionó una herramienta poderosa y flexible para la programación de sistemas.
  - 1989: ANSI C, estandarización del lenguaje.
  - 1999: C99, introducción de nuevas funcionalidades como tipos de datos booleanos y soporte para nuevas bibliotecas.
- **Influencia:** Base de muchos lenguajes modernos (C++, Java, C#), y fundamental para el desarrollo de sistemas operativos y software de alto rendimiento.

```
#include <stdio.h>

int main() {
    int i, suma = 0;

    for (i = 1; i <= 10; i++) {
        suma += i;
    }

    printf("La suma de los primeros 10 números enteros es: %d\n", suma);

    return 0;
}
```

*Imagen 3: Código Lenguaje C*



Puedes identificar que el programa está hecho en C por varias características específicas del lenguaje.

## 1. Bibliotecas Incluidas:

- La línea **#include <stdio.h>** es un claro indicativo de que el programa está escrito en C. Esta directiva de preprocesador incluye la biblioteca estándar de entrada y salida de C.

## 2. Función principal:

- La función **int main()** es otra señal clara. En C, el punto de entrada del programa es típicamente **main**, y la declaración **int** indica que la función retorna un entero.

## 3. Sintaxis de Bucle y Control de Flujo:

- El uso del bucle **for** con la estructura **for (i = 1; i <= 10; i++)** es una construcción típica de C.
- La manera de actualizar la suma **suma += i;** es una abreviatura común en C para incrementar el valor de **suma** por **i**.

## 4. Impresión en Consola:

- La función **printf** es específica de C para imprimir en la consola. El formato de cadena con el especificador **%d** es característico de la función **printf** en C.

## 5. Comentarios:

- Los comentarios de una sola línea en C se indican con **//**, y los comentarios de múltiples líneas se indican con **/\* ... \*/**.

```
// Imprime el resultado
printf("La suma de los primeros 10 números enteros es: %d\n", suma);

return 0; // Termina el programa
```

*Imagen 4: Ejemplo de comentarios en Lenguaje C*



### 2.3. Python

- **Origen:** Creado por Guido van Rossum en 1991, con el objetivo de ser fácil de leer y escribir.
- **Hitos Clave:**
  - 1991: Lanzamiento de Python 0.9.0, con soporte para POO, excepciones y módulos.
  - 2000: Python 2.0, introducción de recolección de basura y soporte para Unicode.
  - 2008: Python 3.0, mejoras significativas en la consistencia y legibilidad del código.
- **Influencia:** Amplia adopción en desarrollo web, ciencia de datos, inteligencia artificial y automatización, destacando por su simplicidad y versatilidad.

```
suma = 0
for i in range(1, 11):
    suma += i
print("La suma de los primeros 10 números enteros es:", suma)
```

*Imagen 5: Código Lenguaje Python*

Puedes identificar que el programa está hecho en Python por varias características distintivas del lenguaje. Aquí están algunos indicadores clave:

#### 1. Sintaxis y Estructura:

- **Indentación:** Python usa indentación para definir bloques de código en lugar de llaves {}. El cuerpo del bucle **for** está indentado.



- **Declaración de Variables:** En Python, no es necesario declarar el tipo de las variables explícitamente.
- **Bucle for:** La sintaxis **for i in range(1, 11):** es específica de Python.

## 2. Funciones y Construcciones Específicas:

- **Función range:** **range(1, 11)** es una función incorporada de Python que genera una secuencia de números del 1 al 10 inclusive.
- **Función print:** **print("La suma de los primeros 10 números enteros es:", suma)** es una función incorporada de Python para imprimir en la consola.

## 3. Concatenación de Cadenas y Variables:

- **Print con Variables:** En Python, puedes pasar múltiples argumentos a **print**, y se concatenan con un espacio entre ellos de forma predeterminada.

## 4. Comentarios:

- Los comentarios se crean utilizando el símbolo **#**. Todo lo que sigue a este símbolo en la misma línea es considerado un comentario y es ignorado por el intérprete de Python.

```
# Bucle que va del 1 al 10 (inclusive)
for i in range(1, 11):
    # Suma el valor de i a la variable suma en cada iteración
    suma += i
```

*Imagen 6: Ejemplo Código con comentarios en lenguaje C*





**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

### 3. CONCLUSIÓN

Comprender la evolución de los lenguajes de programación me ha permitido apreciar la diversidad y la especialización de cada lenguaje. Fortran me mostró la importancia de los cálculos científicos, C me demostró la eficiencia y control en la programación de sistemas, y Python me enseñó la versatilidad y simplicidad para aplicaciones modernas. Esta comprensión profundiza mi habilidad para elegir el lenguaje adecuado para cada tarea y mejora mi enfoque hacia la POO, resaltando la importancia de escribir código claro y modular.