



Universidad Nacional del Litoral  
Facultad de Ingeniería y Ciencias Hídricas  
Departamento de Informática

# Tecnicatura en Informática Aplicada al Diseño Multimedia y Sitios Web

## Elementos de Programación

### Unidad 1: Introducción a la Programación - Parte 2

Docente: Ing. Risso, Oscar Luis  
Año: 2024

# Índice

<b>1. En esta segunda sesión de estudio</b>	<b>3</b>
<b>2. Objetivos</b>	<b>3</b>
<b>3. Clasificación de los Lenguajes de Programación</b>	<b>3</b>
<b>4. Documentación de Programas</b>	<b>4</b>
4.1. Documentación Interna . . . . .	4
4.2. Documentación Externa . . . . .	5
<b>5. Algunos Lenguajes de Programación de Alto Nivel</b>	<b>5</b>
<b>6. Resumen</b>	<b>6</b>

# Unidad 1: Introducción a la Programación

## Parte 2

### 1. En esta segunda sesión de estudio

Existe una gran cantidad de lenguajes de programación. Nosotros estudiaremos una en particular (JAVA). Sin embargo, en esta unidad veremos las características generales que distinguen a unos lenguajes de otros y las ventajas relativas.

También veremos algunos consejos acerca de un proceso que nunca debe faltar en la programación: la documentación.

### 2. Objetivos

Al finalizar este tema Ud. debe lograr:

- Conocer los paradigmas de programación existentes en el actual estado del arte de la ingeniería del software.

### 3. Clasificación de los Lenguajes de Programación

En base a la similitud de los lenguajes de programación de computadoras respecto de nuestro lenguaje natural podemos clasificarlos en:

#### ■ Lenguajes de Bajo Nivel

Si bien estos lenguajes pueden ser interpretados por un humano, esta tarea es altamente compleja. Los lenguajes de bajo nivel se utilizan en situaciones muy particulares, por ejemplo para la programación de sistemas electrónicos sencillos (algunos aparatos electrónicos de uso cotidiano, por ejemplo). Son lenguajes de de bajo nivel:

- Lenguaje Máquina
- Lenguaje Ensamblador

#### ■ Lenguajes de Alto Nivel

Los lenguajes de alto nivel son más accesibles a ser leídos por un humano. Generalmente consisten en un conjunto de palabras y símbolos que tienen una relación directa con su significado. Los lenguajes más ampliamente utilizados hoy en día son de alto nivel. Estos lenguajes facilitan el entrenamiento de recursos humanos (programadores) y permiten resolver más rápidamente problemas de gran complejidad. Los entornos de desarrollo para lenguajes de alto nivel poseen un amplio espectro de herramientas para la edición, compilación y depuración de los programas.

Según la metodología empleada para abordar el diseño del programa:

- **Procedimentales:** estos se basan en un conjunto de subprogramas o subrutinas que resuelven diferentes partes del problema. Por ejemplo, para resolver el problema del cálculo del volumen de un cilindro, podríamos tener un programa principal que pide los datos al usuario y muestra los resultados y un subprograma que realiza el cálculo.
- **Imperativos y Declarativos:** en los lenguajes imperativos las líneas del programa le dicen a la computadora que debe hacer. A lo largo de la evolución de los lenguajes de alto nivel se ha pasado desde los lenguajes imperativos hacia los lenguajes declarativos. En los lenguajes declarativos son aquellos en que en las líneas del programa encontramos las mismas abstracciones del problema en cuestión. Un buen ejemplo de lenguajes declarativos son los lenguajes orientados a objetos.

- **Orientados a objetos:** en este caso, las representaciones que se plasman en el programa tiene una relación directa con la realidad del problema que resuelven. Siguiendo el ejemplo anterior, nuestro programa contaría con un objeto denominado interfaz de usuario y otro cilindro. Estos objetos se encargaría de realizar las operaciones de carga de datos y muestra de resultados (el objeto interfaz de usuario) y de cálculo del volumen (el objeto cilindro).
- **Manejados a eventos:** los lenguajes manejados por eventos son aquellos en los que todas las acciones del programa son disparadas por algún suceso disparador o evento. Este evento puede provenir del usuario (como apretar un botón) o del sistema (como el aviso de que hay espacio insuficiente en el disco). En cualquier caso, el evento es enviados al programa y este podrá las acciones específicas para este evento.

Esta clasificación no es excluyente. Podríamos encontrar, por ejemplo, lenguajes orientados y manejados por eventos. De forma similar, algunos lenguajes son procedimentales y manejados por eventos. En particular, las versiones más recientes del lenguaje Pascal permiten programar según cualquiera de las clasificaciones anteriores e incluso combinarlas a todas en un solo programa. Sin embargo generalmente en una programa en particular se utiliza de forma consistente la metodología para abordar el problema. Es recomendable que cuando se programa orientado a objetos no se mezclen líneas de programas procedimentales.

## 4. Documentación de Programas

La documentación es un aspecto muy importante en la programación. Sin embargo, generalmente los programadores principiantes suelen dejar de lado el proceso de documentación por falta de dedicación o simple desconocimiento de su relevancia. La documentación es todo el material anexo a un programa que no constituye parte del código que se compila. Todo esto sirve para que el mantenimiento, la reparación y la actualización del programa sea rápida y segura. En el caso de que estas tareas deba desarrollarlas otro programador la documentación será imprescindible. Pero más aún, cuando hayan pasado varios meses de hecho el programa, para usted mismo será imprescindible la documentación porque ya no podrá recordar claramente lo que hizo y le constará mucho volver a entender el programa para poder hacer las modificaciones necesarias. La documentación puede realizarse en el mismo archivo de texto del programa mediante comentarios. Los comentarios poseen una sintaxis particular y son totalmente ignorados por el compilador. Esta documentación es denominada interna ya que se encuentra en el mismo archivo de texto del programa. Por otra parte, cualquier tipo de información que no se encuentre en el mismo programa se denomina documentación externa. La documentación externa puede consistir por ejemplo en gráficas, texto explicativo, manual de usuario, algoritmos en pseudocódigo, etc. A continuación podemos ver unas recomendaciones generales en cuanto a la documentación de programas:

### 4.1. Documentación Interna

- Objetivo del programa o módulo.
- Datos de autoría.
- Fechas: iniciación, finalización, modificaciones.
- Finalidad y uso de cada constante, estructura de datos y variable.
- Finalidad y modo de uso de cada procedimiento: entradas, procesos y salidas.
- Comentarios aclaratorios en las líneas de código y bloques.
- Buen uso de la tipografía: mayúsculas y minúsculas.
- Usar correctamente la indentación y separación de bloques lógicos.
- Utilizar nombres significativos para los identificadores.
- Documentar a medida que se realiza la implementación: ¡NO AL FINALIZARLA!

## 4.2. Documentación Externa

- Información general: propósitos, modo de uso.
- Estructura modular
- Árboles jerárquicos de módulos y clases Documentación de las módulos y clases
- Nombre y archivo que la contiene
- Diagrama de herencia con atributos y métodos
- Breve descripción de la clase.
- Atributos: nombre, estructura de datos
- Funcionalidades: nombre, cabecera, finalidad, modo de uso, método y algoritmo
- Fechas y datos de autoría

Puede que algunos términos de esta lista le resulten poco familiares pero los comprenderá a medida que avancemos en el curso.

## 5. Algunos Lenguajes de Programación de Alto Nivel

Existen muchos lenguajes de alto nivel. Cada cual tiene su propio campo de aplicación. Sin embargo, existen lenguajes de propósitos generales como el Pascal, el Basic o el C++. A continuación podemos ver una lista con los lenguajes más difundidos en la actualidad, listados en orden de aparición aproximadamente

- **FORTRAN**: FORmula TRANslator. Lenguaje para cálculos científicos. Posee especiales características para el tratamiento de números. nació a mediados de los 50 y todavía se lo emplea.
- **COBOL**: COmmon Business Oriented Language. Lenguaje orientado a los negocios comunes. Útil para tareas administrativas. Fue creado a fines de los 50.
- **BASIC**: Lenguaje para todo propósito de gran sencillez de aprendizaje. Fue creado a mediados de los 60 y fue muy popular con la aparición de las primeras PCs a principios de los 80.
- **ALGOL**: Lenguaje Algorítmico. De uso científico.
- **PASCAL**: Primer lenguaje estructurado. Nació a principios de los 70 y fue popular por su modularidad, elegancia y sencillez. Respondía al paradigma de la programación estructurada, aunque hoy existen versiones orientadas a objetos.
- **C, C++**: El lenguaje C nació a principios de los 70. Es algo críptico (difícil) para principiantes, pero genera programas muy eficientes. C++ es una extensión de C para el paradigma de objetos. Es muy popular a nivel profesional. En 1998 los comités ANSI e ISO publicaron un standard de este lenguaje de gran potencia, que facilita el desarrollo de aplicaciones.
- **PROLOG**: Programación lógica. Empleado en inteligencia Artificial.
- **SMALTALK**: lenguaje empleado exclusivamente para el modelo de objetos.
- **VISUAL C, VISUAL BASIC, VISUAL PASCAL (Delphi)**: lenguajes de programación visual. Facilitan el diseño de interfaces gráficas a través de herramientas visuales, el código se escribe automáticamente.
- **JAVA**: lenguaje multiplataforma de estructura similar a C++. Empleado para aplicaciones en Internet.
- **HTML**: lenguaje de hipertexto empleado en el diseño de páginas web.

En esta materia se estudiará el lenguaje Java.

## 6. Resumen

1. Al programar se suelen emplear varios tipos de lenguajes. Los de alto nivel se denominan así pues poseen una sintaxis similar a la de cualquier lenguaje natural empleado por las personas para comunicarse. Un programa escrito en un lenguaje de programación se denomina programa fuente.
2. Al ejecutar un algoritmo estamos realizando la prueba del mismo (6ta etapa). Si detectamos errores debemos corregirlos o depurarlos (7ma etapa). Por último es importante documentar la solución propuesta (8va etapa).
3. Los programas fuente escritos en lenguajes de alto nivel no son interpretables por una computadora, por ello debemos compilarlos para lograr un programa ejecutable. Es posible usar intérpretes en lugar de compiladores, pero estos últimos son más eficientes.
4. Los lenguajes de alto nivel se denominan así pues poseen una sintaxis similar a la de un lenguaje natural empleado por personas para comunicarse. Esto nos facilita la legibilidad e interpretación de los programas.
5. Existen varios tipos de lenguajes de programación de alto nivel. Algunos fueron creados para facilitar la resolución de cierto tipo de problemas, aunque predominan los de aplicación general. A su vez, estos lenguajes suelen responder a ciertos modelos o paradigmas de diseño de soluciones.