

# USO DE UN LENGUAJE DE PROGRAMACIÓN DE ALTO NIVEL PARA LA IMPLEMENTACIÓN DE SOLUCIONES COMPUTACIONALES

**Maria C. Torres Madroñero**  
**Profesora asociada**  
**Departamento de Ciencias de la Computación y la Información**

## Objetivos

- Recordar los fundamentos básicos de programación: variables, expresiones, métodos, estructuras de control (condicionales y ciclos), para la solución de problemas.
- Identificar las etapas para la implementación de una solución computacional empleando un lenguaje de programación de alto nivel.

## Recursos requeridos

- PC
- IDE para JAVA o Python – el estudiante deberá seleccionar un lenguaje de programación para el desarrollo de las prácticas de laboratorio

## Actividades preliminares al laboratorio

- Lectura de la guía
- Instalación de IDE seleccionado (opcional)

## Marco teórico

A continuación, se presentan la terminología y definiciones de conceptos y herramientas que se emplean para el diseño, implementación y prueba de soluciones computacionales.

**Algoritmo:** un algoritmo es una secuencia ordenada de pasos lógicos que permiten transformar datos de entrada en nuevos datos de salida. Para la descripción de un algoritmo se deben identificar al menos tres elementos: los parámetros de entrada, los parámetros de salida, y el paso a paso que permite manipular los datos de entrada para llegar a la salida. Una forma más fácil de entender que es un algoritmo es asimilarlos a una receta de cocina, donde los ingredientes en nuestro caso son los parámetros de entrada y el producto esperado son los parámetros de salida. Como en una receta de cocina, el paso a paso debe seguir una secuencia lógica para obtener el resultado esperado, esto quiere decir que el orden en que se ejecutan las instrucciones afectará el producto final. La descripción, presentación y diseño de un algoritmo usualmente se presenta en forma de pseudocódigo.

**Seudocódigo:** el pseudocódigo es una descripción de alto nivel de los pasos lógicos que hacen parte de un algoritmo. En un pseudocódigo se identifican los datos de entrada, los datos de salida, y el paso a paso que permite la transformación de los datos de entrada a la salida. Dado que son descripciones de alto nivel usualmente no se incluyen declaraciones de tipos de variables y algunas funciones o métodos que corresponden a operaciones ampliamente conocidas (ejemplo: la función *length* para retornar la longitud de un vector). Los pseudocódigos emplean palabras reservadas que son ampliamente usada en diferentes lenguajes de programación, tales como las palabras FOR y WHILE reservadas para ciclos para y mientras, o la palabra IF reservada para secuencias de control condicional. Sin embargo, los pseudocódigos son independientes del lenguaje de programación que se empleará para la implementación de la solución computacional; este debe incluir la descripción suficiente y necesaria para que un programador pueda codificar la solución en cualquier lenguaje de programación. Se recomienda la escritura del pseudocódigo en inglés, esto permite que este sea entendible por cualquier programador alrededor del mundo.

**Paradigma de programación:** Los paradigmas de programación definen el estilo para diseñar una solución computacional para resolver un problema. Entre los paradigmas de programación de mayor uso se encuentran la programación imperativa, la programación lógica, la programación funcional y la programación orientada a objetos.

Un error común es confundir un paradigma de programación con un lenguaje de programación. Los lenguajes de programación permiten la implementación de programas que siguen uno o más paradigmas.

**Lenguaje de programación:** define el conjunto de símbolos, instrucciones y reglas gramaticales que permiten implementar una solución computacional, para que este sea interpretado y ejecutado en nuestro computador. Los lenguajes de programación se pueden clasificar como lenguajes de alto y bajo nivel de acuerdo con el nivel de abstracción respecto a la arquitectura del computador. Por otra parte, los lenguajes también se pueden clasificar de acuerdo con el paradigma, aunque un lenguaje puede soportar uno o más modelos de cómputo. Entre otros elementos, los lenguajes de programación definen como declarar, inicializar y manipular variables y arreglos, como establecer estructuras de control como condicionales y ciclos, así como la declaración y uso de funciones.

**Entorno de desarrollo integrado:** un IDE (por sus siglas en inglés *Integrated Development Environment*) es una aplicación o software que facilita la implementación de una solución computacional. Un IDE incluye al menos un editor de código fuente, un compilador y un depurador. El editor de código fuentes es un editor de texto diseñado específicamente para la escritura de códigos de un programa. Los editores integrados en IDE están diseñados para simplificar la escritura de código, con herramientas como la identificación de palabras reservadas, indexación, autocompletar y verificadores de coincidencia de corchetes. Por otra parte, el compilador traduce el código escrito en determinado lenguaje de programación a otro lenguaje usualmente de más bajo nivel, para construir un programa ejecutable. Un depurador es un programa que permite probar y depurar otros programas. La principal tarea de un depurador es correr un programa bajo condiciones controladas evaluando el progreso de las operaciones y el uso de recursos.

## Actividades

**Programa 1 – Variables, expresiones y estructuras de control:** Diseñe un programa que le pida al usuario ingresar  $n$  números enteros, donde  $n$  también es ingresado por el usuario. El programa debe calcular (i) el valor máximo, (ii) el valor mínimo, (iii) la suma de los enteros ingresados y (iv) el valor promedio.

**Programa 2 – Archivos y cadenas:** Diseñe un programa que lea el archivo de texto *test\_pr2.txt* (archivo adjunto a esta guía) y cuente cuantas veces se repite la palabra “en” en el párrafo leído. Tenga en cuenta que el párrafo esta dividido en varias líneas. (No modifique el archivo de texto).

**Programa 3 – Tablas y diccionarios:** Diseñe un programa que realice el control de acceso a una aplicación. El programa almacena los usuarios y contraseñas mostrados en la siguiente tabla. El programa solicita al usuario ingresar su nombre y contraseña. Si el usuario introduce de forma correcta los datos, el programa muestra en pantalla el mensaje “Acceso permitido”, en caso contrario indicara “Datos incorrectos”. Si el usuario se equivoca tres veces consecutivas el programa indicara “Lo siento, su acceso no es permitido”.

Nombre Usuario	Contraseña
Juan1223	J12an*.
Maria2345	M23a*.
Pablo1459	P14o*.
Ana3456	A34a*.

## Instrucciones de entrega

- La solución de los problemas debe desarrollarse en JAVA o Python. Los estudiantes tendrán la libertad de seleccionar el lenguaje de programación y plataforma para presentar la solución de los problemas.

- La solución debe emplear librerías nativas y se invita a los estudiantes a no usar código descargado de internet. Los laboratorios están diseñados para practicar los fundamentos teóricos; entre más código escriba el estudiante más fácil será su comprensión de los temas de clase.
- La solución se puede presentar en grupos de hasta 3 estudiantes.
- La solución de los problemas debe entregarse y sustentarse en el aula de clase o en la hora de asesoría a estudiantes. No se reciben soluciones por correo electrónico.