

# ALGORITMOS Y PSEUDOCÓDIGO

<b>INTRODUCCIÓN</b>	<b>2</b>
¿Qué es un dato?	2
<b>CONCEPTO DE INFORMACIÓN</b>	<b>2</b>
<b>CONCEPTO DE PROGRAMA</b>	<b>2</b>
<b>CONCEPTO DE ALGORITMO</b>	<b>3</b>
FASES DE DISEÑO DE UN ALGORITMO	5
ELEMENTOS DE UN ALGORITMO	5
<b>PARADIGMAS DE LA PROGRAMACIÓN</b>	<b>6</b>
Paradigma declarativo	6
<b>TIPOS DE LENGUAJES</b>	<b>7</b>
Lenguaje ensamblador	7
Lenguaje de alto nivel	7
Lenguaje de cuarta generación	8
<b>HERRAMIENTAS PARA EL DESARROLLO</b>	<b>8</b>
<b>ENTORNO DE DESARROLLO</b>	<b>9</b>
<b>Ciclo de la vida del software</b>	<b>10</b>

# INTRODUCCIÓN

## ¿Qué es un dato?

Es un conjunto de símbolos utilizados para representar un valor numérico. Un hecho, una idea, un objeto

Por ejemplo, un dato puede ser el número de la seguridad Social, el número de teléfono, la matrícula de un vehículo....

Los datos son

- Representaciones simbólicas
- Por sí, solos no tienen significado
- No pueden transmitir un mensaje
- Se derivan de la descripción de ciertos hechos

## CONCEPTO DE INFORMACIÓN

Se entiende por **información un conjunto de datos que han sido procesados, es decir, significativos para un determinado fin.**

*Ejemplo: nombre edad estudios salario son un conjunto de datos.*

*Por separado se trata de datos individuales . Sin embargo, los organizamos por edad y salario y tendremos un informe sobre la distribución del sueldo en función de la edad, es decir, serán datos significativos.*

**Características de la información:**

- Es la unión de datos que se han procesado y organizado
- Poseen un significado
- Pueden transmitir un mensaje
- Aumentan el conocimiento de una situación

Para que sea información ha de cumplir lo siguiente:

- **Ser útil:** ha de ajustarse a las búsquedas del usuario
- **Ser confiable:** que la información sea verídica, certera y confiable.
- **Ser oportuna:** de qué sirve saber que en EEUU llueve sí, soy de España.

## CONCEPTO DE PROGRAMA

Un **programa** es un conjunto de instrucciones correspondientes a cada paso hecho en el algoritmo. Se escribe en un **lenguaje de programación**.

La actividad de expresar un algoritmo en forma de programa se le denomina **programación** o **codificación** y es parte fundamental en el **desarrollo de aplicaciones**.

- Hardware: está formado por los componentes del ordenador.
- Software: es el conjunto de programas ejecutables por el ordenador:
  - Software de sistema
  - Software de programación
  - Software de aplicación

## CONCEPTO DE ALGORITMO

Un algoritmo consiste en una secuencia de pasos e instrucciones, cuyo objetivo principal es dar solución a un problema o llevar a cabo una tarea.

Para comprender en qué consiste un algoritmo , no es necesario que pensemos solo en términos informáticos , sino que bastará compensar en casi cualquier tarea diaria para obtener un ejemplo, *por ejemplo: como el algoritmo necesita para hacer pan o la secuencia de pasos que seguimos para llegar desde casa al trabajo.*

Por tanto, los algoritmos son una herramienta que permiten escribir un conjunto finito de instrucciones , pasos o procesos , ordenado secuencialmente , que tiene como resultado final la solución a un determinado problema . Las tres características principales de un algoritmo son:

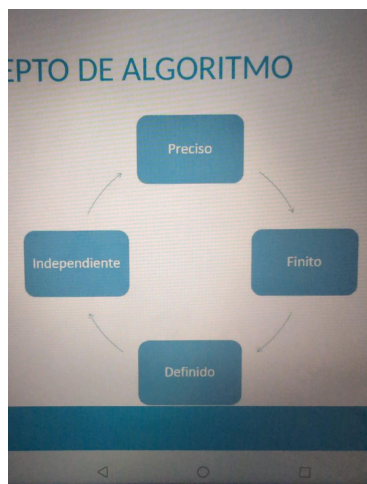
- **Realizable:** es decir, **todo algoritmo debe concluir después de ejecutar una cantidad finita de instrucciones o pasos, se debe obtener un resultado final, sea satisfactorio o no.** Diremos que es irrealizable si no se concluye la ejecución de pasos o estos son infinitos.
- **Comprensible:** las instrucciones y operaciones codificadas deben ser comprensibles , en este caso por el ordenador , para que la ejecución pueda llevarse a cabo .
- **Preciso:** el algoritmo debe ser preciso, es decir, antes una misma entrada o mismo proceso de ejecución. El resultado debe ser siempre el mismo. Cuando se habla de algoritmos , podemos hablar de tres tipos de pensamiento:
  - Computacional
  - Algorítmico
  - Procedimental.

Se expone a continuación la definición de estos, en base a lo expuesto por Moursund( 2006).

- **Pensamiento computacional**, relativo a la representación y solución de problemas a través de máquinas.
- **Pensamiento algorítmico**, desarrollo de algoritmos cuyo fin es la resolución de algún tipo de tarea o problema.
- **Pensamiento procedimental**, desarrollo de procedimiento que presentan como fin la resolución de algún tipo de tarea o problema que puede o no resultar exitoso.

Es importante para comprender el significado de la programación , saber que en el pensamiento computacional se define como el proceso por el cual nos enfrentamos a un problema. Aplicando habilidades propias de la computación , implica resolver problemas cotidianos haciendo uso de la programación informática.

Constituye un proceso mental utilizado para formular problemas, cuya solución puedan ser llevadas a cabo por un ordenador.



**Preciso**, diremos que un algoritmo es preciso cuando la instrucciones que lo conforman sean claras.

**Definido**, diremos que un algoritmo es definido, siempre que al ser ejecutado más de una vez se obtenga el mismo resultado . El resultado obtenido será el mismo independientemente del ordenador donde se ejecuta el algoritmo.

**Finito**, el algoritmo cuenta con un número determinado de pasos. Este consta de un inicio y de un fin.

**Independiente**, el algoritmo no dependerá del lenguaje en que se codifique.

El pensamiento computacional puede descomponerse en cuatro grandes bloques:

- Descomposición
- Abstracción
- Algoritmos
- Patrones.

La resolución de problemas se puede dividir en tres grandes bloques generales:

- Análisis del problema

- Diseño o desarrollo del algoritmo
- Resolución del algoritmo en el ordenador , estas a su vez se subdividen en otras.

## FASES DE DISEÑO DE UN ALGORITMO

En el primer paso , el **análisis del problema** requiere que en este sea definido y comprendido claramente para que pueda ser analizado con todo detalle.

Una vez analizado , se lleva a cabo el diseño **del algoritmo**, procedimiento paso a paso para solucionar el problema.

Para resolver más eficazmente cualquier tipo de problema, es conveniente recurrir a la división del problema central en subproblemas, a este método se le llama divide y vencerás.

**En el primer diseño del algoritmo, los pasos suelen ser más generales e incompletos.**

Tras la primera descripción estos se amplían y se detalla más , a este proceso. Se le denomina refinamiento del algoritmo.

Por último , para **resolver el algoritmo** en una máquina se necesita codificar el algoritmo en un lenguaje de programación , es decir , convertir el algoritmo en programa , ejecutarlo y comprobar que este soluciona verdaderamente el problema.

El programa que implementa el algoritmo debe ser escrito en un lenguaje de programación , siguiendo las reglas gramaticales o sintaxis del programa escogido , la fase de conversión del algoritmo en un lenguaje de programación se denomina codificación.

**Tras la codificación del programa , deberá ejecutarse en un ordenador y después de comprobar los resultados pasar a la fase final de documentación.**

## ELEMENTOS DE UN ALGORITMO

- **Entrada:**
  - Por pantalla
  - Por teclado
  - Voz
  - Datos a tratar
- **Proceso:** operaciones que transformarán los datos en información.
- **Salida:**
  - Pantalla
  - Alarmas
  - Ficheros
  - Movimiento de motores

*Ejemplo: define un algoritmo que solviente un problema que sea común. Para ello, define la entrada del algoritmo , en qué consistirá el procesamiento del algoritmo y define la salida del mismo.*

*“ Arreglar el pinchazo de la rueda de un coche”*

*Entradas: rueda nueva y herramientas.*

*Proceso: cambiar la rueda. :*

- 1. Elevar el coche*
- 2. Quitar la rueda*
- 3. Coger la rueda de repuesto*
- 4. Fijar la rueda nueva*
- 5. Bajar el coche.*

*Salida: vehículo reparado.*

## PARADIGMAS DE LA PROGRAMACIÓN

**Un paradigma de programación indica un método de realizar cálculos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa.**

Los paradigmas fundamentales están asociados a determinados modelos de cómputo.

También se asocian a un determinado estilo de programación

Los lenguajes de programación suelen implementar , a menudo de forma parcial , varios paradigmas.

### Paradigma declarativo

Describe qué se debe calcular sin explicar cómo.

Las variables son nombres asociados a definiciones y, una vez declarada son inmutables.

No existe la sentencia de **asignación**.

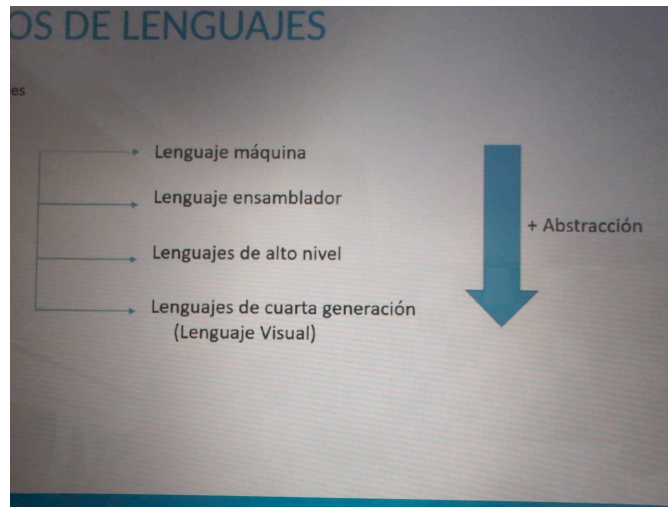
- Programación declarativa
  - Programación lógica
  - Programación funcional.

Permiten comunicarnos con el computador.

Existen diferentes tipos de lenguajes de programación y dentro de cada tipo diferentes lenguajes.

La clasificación de estos se realiza en función de su nivel de abstracción operativa , esto implica que sea necesario, detallar más o menos las operaciones para poder llevar a cabo una tarea completa.

Cuanto menos haya que indicar al ordenador, de mayor nivel será el lenguaje.  
La clasificación a la que se está haciendo referente es la siguiente.



## TIPOS DE LENGUAJES

### Lenguaje ensamblador

Ese lenguaje nativo de la CPU. Las instrucciones se han decodificar en código binario. Hay un lenguaje máquina distinto por cada tipo diferente de CPU.

¡Recuerda! Todo programa ha de ser traducido al lenguaje máquina del ordenador sobre el que va a ser ejecutado.

Tiene mayor independencia de la máquina y es más cercano al lenguaje humano

Hace uso de símbolos memotécnicos, palabras cortas, para hacer referencia a los datos del lenguaje y máquina en lugar de usar dígitos binario.

Un lenguaje ensamblador corresponde a un determinado lenguaje máquina.

### Lenguaje de alto nivel

También son llamados como lenguajes de tercera generación

Son lenguaje independientes de la máquina en la que se utiliza

Estos lenguajes no hacen referencia al funcionamiento de la CPU, sino a tareas enfocadas al usuario.

*Ejemplos: Java, COBOL o Fortran*

Cada instrucción de alto nivel se asocia a una o más instrucciones máquina

Utilizan instrucciones que se asemejan al lenguaje ordinario

De igual forma, un programa escrito en alto nivel necesita ser traducido al lenguaje máquina

Lenguaje sencillo de aprender

Es un lenguaje orientado a objetos

*Ejemplo Java ::*

*Dispone de un amplio catálogo de librería*

*Es el lenguaje robusto y portable*

*Se trata de un lenguaje de alto rendimiento*

*Permite la programación multihilo*

## Lenguaje de cuarta generación

Son lenguajes que trabajan a mayor nivel de abstracción

Suelen incluir capacidad para trabajar con base de datos BBDD

En función del estilo de programación se suelen clasificar en:

- Lenguaje imperativo
- O lenguaje declarativo

*Ejemplo: SQL o MatLab*

## HERRAMIENTAS PARA EL DESARROLLO

**GITHUB:** servicio que permite almacenar repositorios git en la nube. En otras palabras , es como un Google Drive para proyectos de software, donde puedes encontrar código .

Cuando abras tu proyecto, puedes elegir entre hacerlo público o privado. La comunidad es enorme y la base de proyectos es aún más grande. GitHub también es un sitio excelente para el desarrollo colaborativo , en el que desarrolladores. Pueden descubrir , compartir y construir el mejor software.

**Stack OverFlow:** es la comunidad online de programadores más grande del mundo. Cada mes , la visita más de 50.000.000 de desarrolladores. Este es el lugar donde aprenden comparten conocimientos e impulsan su carrera . A nuestros desarrolladores. Le encanta este foro y lo visitan casi cada día porque en él se puede encontrar la respuesta a casi todas las preguntas y dudas que tengas.

**Confluence:** es un software utilizado para la colaboración en equipo. Es un lugar fantástico para poner en común idea , requisito en los que el equipo de producto está trabajando y documentación para construirlo mejor. Es genial porque es muy fácil de utilizar y todos pueden contribuir.

**Kanban:** es una de las aplicaciones de gestión de proyectos más utilizadas y mejor conocida . Todos los proyectos se representan con tableros que contienen listas . Cada lista tiene tarjeta de proceso que tú creas y que puede moverse utilizando la función arrastrar y

soltar. Los usuarios relacionados con cada tablero. Pueden asignar las tarjetas a otro usuario. Kanban tiene muchas funciones útiles, como por ejemplo, escribir comentarios, añadir archivos adjuntos, establecer fechas de vencimiento, crear listados, añadir etiquetas de colores, espuma, integración con otras apps, etc.

## ENTORNO DE DESARROLLO

Conjunto de procedimientos y herramientas que son utilizadas para desarrollar un programa. Es comúnmente conocido como IDE: herramienta de desarrollo de software utilizado para escribir, generar, aprobar y depurar código en diferentes modos.

Para escoger el ID correcto, es necesario tener claro en qué lenguaje. Queremos programar.

### **Eclipse**

Fue desarrollado por IBM para cumplir alguna de las características básicas para satisfacer las expectativas del usuario.

Actualmente ha gestionado por una fundación independiente sin ánimo de lucro.

Permite instalar extensiones y plugins que se pueden añadir al IDE.

Permite editar, compilar, ejecutar y depurar. Código en C++, Java, JavaScript, PHP, Python, etc.

### **NetBeans**

IDE creado inicialmente para programar Java.

NetBeans es un programa libre y gratuito y sin restricciones de uso.

NetBeans soporta el desarrollo de los tipos de aplicaciones Java (J2SE, EJB, y aplicaciones móviles).

Posee un sistema de control de versiones.

### **IntelliJ IDEA**

IDE desarrollada por JetBrains anteriormente conocido como IntelliJ, y está disponible en dos ediciones: edición para la comunidad y edición comercial.

La versión 12.1 incluye soporte para Java 8, diseñador UI para desarrollo de Android.

### **Xcode**

IDE pensado para el desarrollo de apps iOS / OSX.

Puede compilar código C, C++, Swift, Objective-C, Objective-C++, Java y Applescript entre otros.

Actualmente se encuentra en la versión 11.

# Ciclo de la vida del software

Desde el punto de vista general, todo software pasa por tres fases: **desarrollos , uso y mantenimiento.**

LEER EN LA PLATAFORMA PUNTO 11