

Programación de Software Informático

TEMA 1.1

Ingeniería del Software.

La **Ingeniería del Software** es una de las disciplinas de las ciencias de la computación, que estudia la creación de software rentable, confiable y de calidad, mediante la aplicación de principios, métodos y técnicas de desarrollo, operación, y mantenimiento, en base a un enfoque sistemático, disciplinado y cuantificable. En resumen, son todas las teorías, métodos y herramientas que se utilizan en el desarrollo del software. El producto software desarrollado debe ser eficaz (hacer correctamente su trabajo) y eficiente (optimizando el consumo de recursos del sistema).

El **Software** (programas + datos) es un conjunto de programas de tratamiento de datos a procesar, junto con la información adicional necesaria para su operación, incluyendo los documentos que describen los procedimientos de uso y funcionamiento.



```
#include <stdio.h>
void main () { // main (int argc, char[] argv) [
    printf ("Hola y Adios");
}
```



```
#include <iostream>
main (int argc, char **) {
    std::cout << "Hola y Adios\n";
}
```



```
public class HolayAdios2 {
    public static void main (String args[]) {
        System.out.println ("hola y Adios");
    }
}
```

Ingeniería del Software.

La **Programación** es el proceso de confección (edición, traducción, ejecución, depuración) de **programas**, mediante determinados **lenguajes de programación**, desde la codificación en código fuente, hasta la generación del código máquina, que es el lenguaje de la computadora, para que pueda ser ejecutado, pasando por otras formas de código intermedio.

El código objeto es un código binario intermedio próximo al código máquina, pero a falta del montaje de ciertas partes enlazadas como módulos aparte.

El bytecode es un código binario intermedio más abstracto e independiente del hardware que el código máquina, generalmente interpretado por máquinas virtuales, con la ventaja de la portabilidad entre distintas plataformas (equipo informático con una arquitectura de ordenador y sistema operativo con unas librerías de rutinas).

Un **Programa** es un conjunto de instrucciones encaminadas a resolver un problema determinado de procesamiento o tratamiento de datos de entrada y/o salida, que se corresponde con la implementación de un **algoritmo**, por una máquina o equipo con capacidad de computación.

Los programas se almacenan en archivos, uno o varios, según la complejidad del programa, de distinto tipo, según la fase de construcción en la que se encuentre.

Un **Algoritmo** es un conjunto de operaciones destinadas a realizar una tarea con un determinado nivel de abstracción o de concreción, siguiendo un flujo de pasos sucesivos según una lógica de decisión que permita alcanzar la solución o estado final resultante a partir del estado inicial de partida.

Un algoritmo debe ser preciso (orden de pasos bien definido), regular (misma respuesta ante las mismas condiciones) y finito (número limitado de pasos con un final).

La **Metodología de la Programación** estudia la forma de resolver y representar **algoritmos** para obtener **programas** mediante técnicas de **programación** basadas en reglas formales, según distintos formatos de representación (gráfico, simbólico, lingüístico, nemotécnico, ...) de los diferentes tipos de instrucciones, para los elementos (inicio, fin o parada de programa; procesamiento o tratamiento; condición o decisión; líneas de flujo, conectores, enlaces y comentarios) y estructuras básicas de programación (secuencia; alternativa simple, doble y múltiple; iteración mientras, hasta y para).

Un **Lenguaje de Programación** es el conjunto de elementos y reglas que permite especificar, de forma concreta y precisa, las acciones a realizarse desde programa.

Los lenguajes de programación definen la estructura gramatical (sintaxis), y el alfabeto (léxico), que determinan las construcciones válidas y su significado (semántica).

Software.

Clasificación

- **Software de sistema** (software básico o de base).
 - **Software funcional**: programas (base del sistema operativo) encargados de la gestión interna de la computadora para su correcto y óptimo funcionamiento obteniendo el mayor rendimiento posible de la computadora y proporcionando al usuario un medio fácil de manejo de los recursos disponibles (como intermediario entre este y la computadora).
 - **Programas de Control** (gestión de recursos + interface de usuario): programas para la explotación del sistema y la sincronización de operaciones.
 - **Programas de Proceso** (programas y herramientas de desarrollo de software).
 - **Software general**: programas complementarios del sistema operativo que permite hacer uso, de la manera más fácil y cómoda posible, de las posibilidades que ofrece la computadora aprovechando al máximo sus prestaciones.
 - **Programas de Utilidad** (utilidades o utillería): programas que realizan funciones útiles para el sistema o el usuario facilitando el trabajo.
 - **Programas de Servicio**: programas encargados de realizar funciones para otros programas
- **Software de Aplicación** (software ofimático y corporativo): software específico constituido por programas, junto con los datos gestionados por los programas, interrelacionados y encargados de resolver de forma automática por medio de la computadora problemas asociados a cualquier tipo de trabajo.
 - **Software a medida**: conjunto de programas desarrollados para resolver problemas concretos relacionados con trabajos de carácter específico, bien, poco adaptables (sistema de control de tráfico, sistema de monitorización de pacientes, ...) o adaptables (nóminas, almacén, facturación, contabilidad, ...) a diferentes entornos.
 - **Software estándar**: conjunto de programas desarrollados para resolver problemas genéricos relacionados con trabajos de carácter general y adaptable a distintos entornos y configuraciones buscando dar servicio al mayor número de usuarios posibles de manera que se pueden utilizar de forma independiente como unidad operativa autónoma (**aplicaciones estándar**) o como conjunto de aplicaciones estándar relacionadas y agrupadas de forma conjunta como unidad operativa integrada (**paquetes de software integrado**) tendiéndose a aplicaciones que se pueden usar indistintamente como unidad operativa autónoma o integrada (**software integrable**) permitiendo la interrelación y compartición de datos (siendo el vínculo de unión generalmente el entorno operativo Windows).

Software.

Clasificación

- **Software Libre** (desde los 80): software con libertad para ser usado, estudiado, modificado o distribuido, sin necesidad de autorización, y sin restricción alguna. Puede ser **gratuito** (la mayoría) o **de pago** (una minoría). El **software de código abierto** u **opensource** (desde los 90) es software libre, en el que se hace hincapié y cobra especial relevancia la disponibilidad del código fuente para poderse modificar o redistribuir.
 - **Licencia Pública General** o **General Public License (GPL)**: licencia creada en el marco del proyecto GNU, de software libre con código fuente bajo copyleft. Es la licencia de software libre más común, con diversas versiones y variantes.
 - **Licencia Pública General Reducida GNU** o **GNU Lesser General Public License (LGPL)**: licencia creada en el marco del proyecto GNU, de software libre con código fuente, y con posibilidad de módulos privados.
- **Software no Libre**: software, gratuito o no, con o sin código fuente disponible (generalmente solo incluye los archivos necesarios para la ejecución), con restricciones para ser usado, estudiado, modificado o distribuido, no permitiéndose alguna o ninguna de las libertades del software libre. El **software propietario o privativo** es software no libre comercial con restricciones impuestas por copyright, aunque con posibilidad de variantes demo con menos limitaciones.
 - **Freeware**: software privativo que se distribuye, sin código fuente, gratuitamente y con funcionalidad completa, pero con restricciones que impiden usarse libremente como el software libre.
 - **Shareware**: software privativo gratuito de evaluación, básicamente para prueba antes de su posible adquisición, con imitaciones operativas o temporales frente a la versión de pago, con funcionalidad completa durante un tiempo limitado, o con funcionalidad incompleta durante un tiempo ilimitado.

La licencia de software establece los términos y condiciones de uso mediante contrato, con varias modalidades: copyright, protegido por derechos de autor; copyleft, con licencia solo para que siga siendo copyleft en caso de modificarse o redistribuirse; dominio público, sin licencia ni restricciones, aunque pudiendo las copias ser de pago.

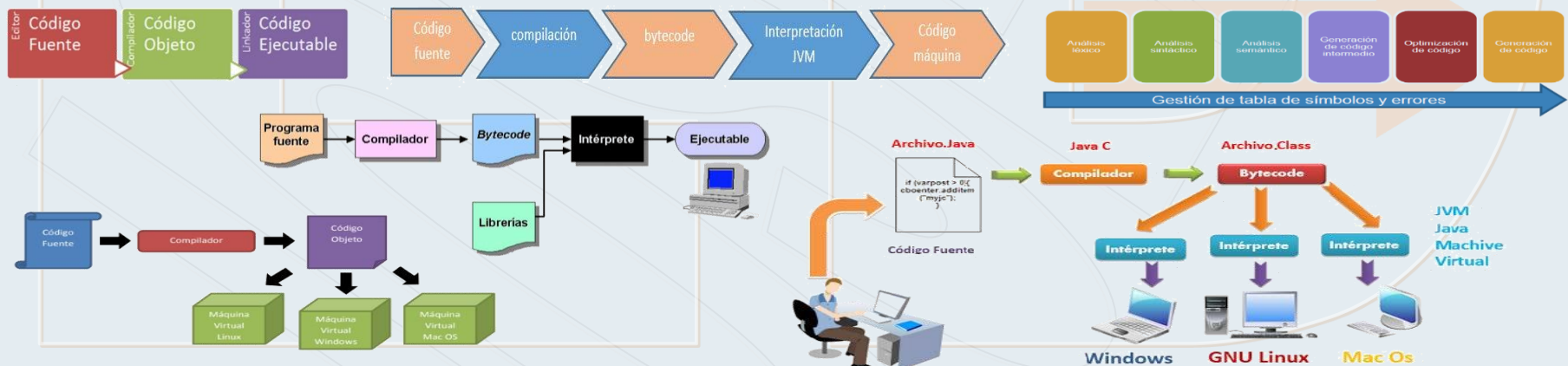
El movimiento de software libre tiene su origen en el proyecto GNU (GNU's Not Unix), impulsado desde sus inicios en la década de los 80 (1983) por un grupo precursor liderado por Richard Stallman, y auspiciado por la fundación de software libre o Free Software Foundation (FSF), con motivación basada en principios éticos.

Programación.

Fases

- **Edición** (estudio especificaciones, representación algoritmo, codificación en el lenguaje) → código fuente.
- **Traducción.**
 - **Análisis** (léxico, sintáctico, semántico).
 - **Compilación / Interpretación.**
 - **Compilación** → código intermedio (objeto, bytecode, pcode).
 - Generación de código.
 - Optimización de código.
 - Enlace (todo o parte, una vez o cada vez): estático (librerías) / dinámico (DLL) → código ejecutable.
 - **Interpretación** → código ejecutable.
- **Ejecución** (pruebas parciales de módulos y prueba final de programa).
- **Depuración.**

Cada una de las fases de la programación se lleva a cabo por medio de software creado para dichas fases, que se puede utilizar, de manera más o menos automática o manual, independientemente, o conjuntamente a través de entornos integrados de desarrollo (IDE), que incluyen todos los elementos necesarios.



Lenguajes de programación.

Evolución

- **1ª Generación:** lenguajes de bajo nivel (lenguaje máquina).
- **2ª Generación:** lenguajes de bajo nivel con mejoras (lenguaje ensamblador).
- **3ª Generación:** lenguajes de alto nivel. Programación estructurada y modular.
- **4ª Generación:** lenguajes de alto nivel con mejoras (orientación a objetos, con código y datos integrados).
- **5ª Generación:** lenguajes de muy alto nivel (inteligencia artificial, con capacidad de aprendizaje, y sistemas expertos, con reconocimiento de patrones, realidad virtual, realidad aumentada, metaverso, etc).

Elementos

- **Estructura Gramatical:** léxica (palabras clave reservadas, identificadores, delimitadores, comentarios, ...), sintáctica (reglas de combinación de elementos), y semántica (significado de las construcciones).
- **Estructuras de datos** (estáticas / dinámicas, contiguas / no contiguas, lineales / no lineales).
 - **Tipos simples o básicos o elementales o fundamentales o primitivos:** escalares o atómicos o indivisibles, y no compuestos por otros tipos: (enteros / reales / lógicos / caracteres / conjuntos o enumerados / subrango o subcampo).
 - **Tipos compuestos, estructurados o derivados:** están formados por otros tipos (cadenas de caracteres / fechas / matrices o arreglos o arrays (vector, tabla, ...) / estructuras o registros / archivos o ficheros).
- **Expresiones y Operadores** (unarios / binarios / ternarios, asignación / aritmético / lógico / relacional o de comparación / bits / signo / condicional / tamaño / puntero). Precedencia y Asociatividad.
- **Sentencias o Instrucciones.**
 - **Declaración y Definición:** asignación de tipo, reserva inicial de espacio de memoria, posibles restricciones sobre valores,
 - **Procesamiento de Datos:** asignación, cálculo, (lectura, escritura, ...), ...
 - **Control de Flujo de Programa:** secuencia, alternativa (simple, doble, múltiple), iteración (mientras, hasta, para).
- **Funciones y/o Procedimientos.**

Lenguajes de programación.

Clasificación

- **Bajo / Medio / Alto nivel.**
 - Bajo nivel: máquina, ensamblador.
 - Alto nivel: C, C++, C#, Objective-C, Java, JavaScript, Python, Swift,
 - Medio nivel o híbrido: C, C++, C#, Objective-C, y derivados, u otros similares.
- **Imperativo / Declarativo / Concurrente.**
 - Imperativo (como hacer).
 - Procedimental (clásico): Basic, Logo, Cobol, Fortran, Pascal, C, C++, C#, Objective-C, Java, JavaScript, Python, Swift,
 - Orientado a Objetos (abstracción, encapsulación, herencia, polimorfismo): Visual Basic, Smalltalk, Eiffel, Ada. Modula, Object Pascal, C++, C#, Objective-C, Java, JavaScript, Python, Ruby, Swift, ...
 - Declarativo (que hacer).
 - Funcional: Logo, Lisp, Racket, Clojure, Scheme, Miranda,
 - Lógico: Absys, Prolog,
 - Concurrente.
 - Basados en semáforos: PL/I, Algol, C, Java, Python, ...
 - Basados en monitores: Concurrent Pascal, Modula, Python, ...
 - Basados en buzones de mensajes: Elixir, Erlang, ...
 - Basados en cerrojos y memoria compartida: Java, ...
 - Basados en variables atómicas: Java, ...
- **Interpretado / Compilado / Virtual (mixto).**
 - Interpretado: Basic, JavaScript, PHP, Perl, Python, Ruby,
 - Compilado: C, C++, C#, Objective-C, Java, Rust, Go,
 - Virtual (mixto): Java, Kotlin,
- **Gestión o Comercial** (Cobol) / **Científico** (Fortran) / **Multipropósito** (C, Java) / **Especializado** (Unity, matlab) / **Educativo** (Logo).

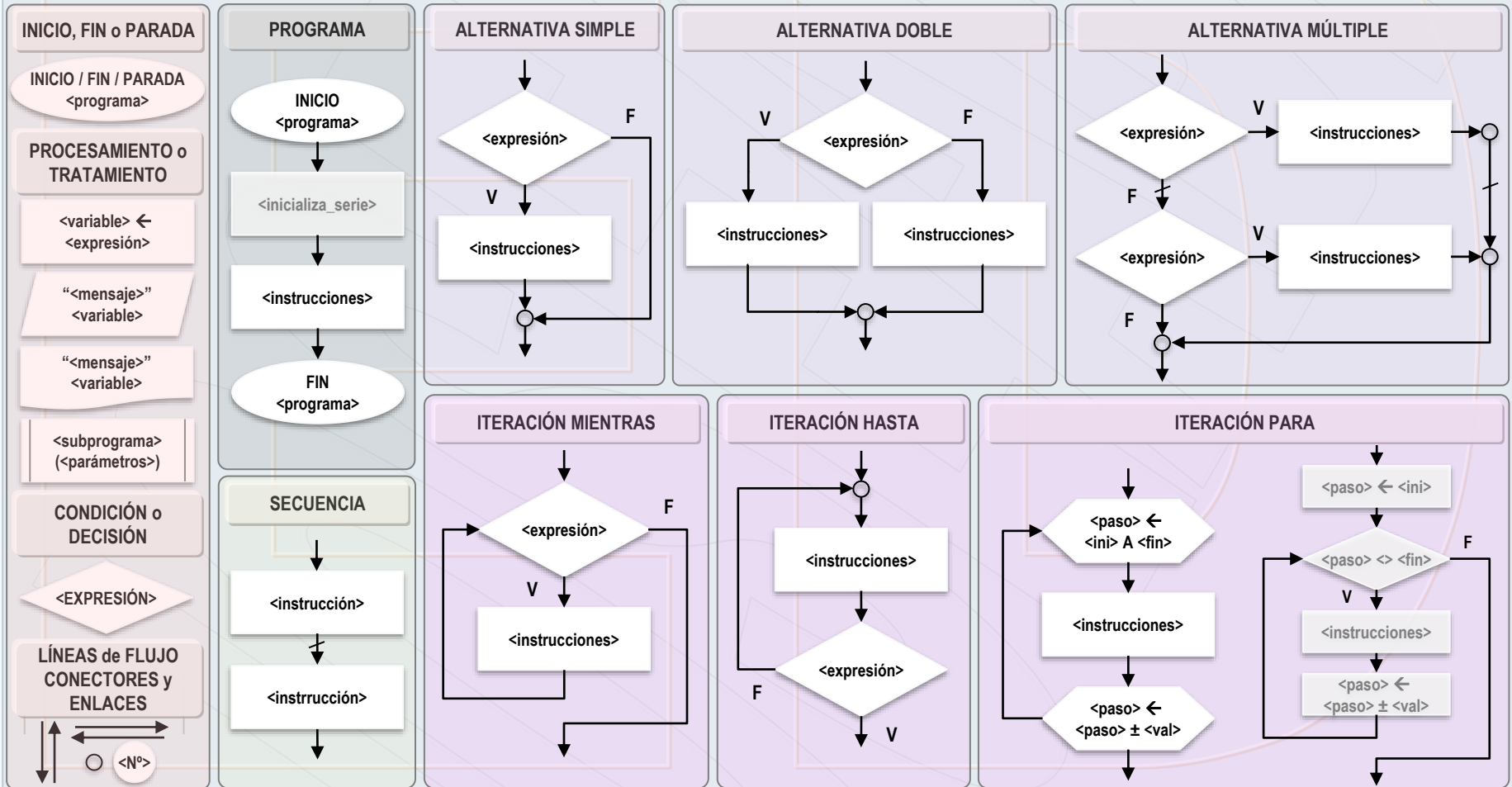
Lenguaje	Características
Lenguaje C	Medio nivel, compilado, procedural
Ensamblador	Bajo nivel, compilado, procedural
HTML	Lenguaje de marcas, no es de programación
PHP	Alto nivel, interpretado, procedural
Javascript	Alto nivel, interpretado, procedural
JAVA	Alto nivel, compilado, POO
JSP	Alto nivel, compilado, POO
Prolog	Muy alto nivel, compilado-interpretado, declarativo



Una **Máquina Virtual de sistema** es un software que emula un ordenador por completo, de forma que se puede instalar en su interior otro sistema operativo con su propio disco duro, memoria, tarjeta gráfica, etc (VirtualBox, Vmware Workstation, ...). Una **Máquina Virtual de proceso** es un software que proporciona un entorno portable de ejecución de un proceso, con independencia de la plataforma. La **Máquina Virtual Java (JVM)** es una máquina virtual de proceso que proporciona el entorno en el que se interpretan instrucciones expresadas en un código intermedio de bytes (bytecode de Java), que es generado por el compilador del lenguaje Java, y se traducen a instrucciones en código nativo propio de una plataforma específica para su ejecución.

Metodología de la Programación.

El **Diagrama de Flujo u Ordinograma** es una técnica de representación algorítmica, mediante lenguaje gráfico a base de símbolos geométricos, de los elementos y estructuras básicas de programación para describir algoritmos.



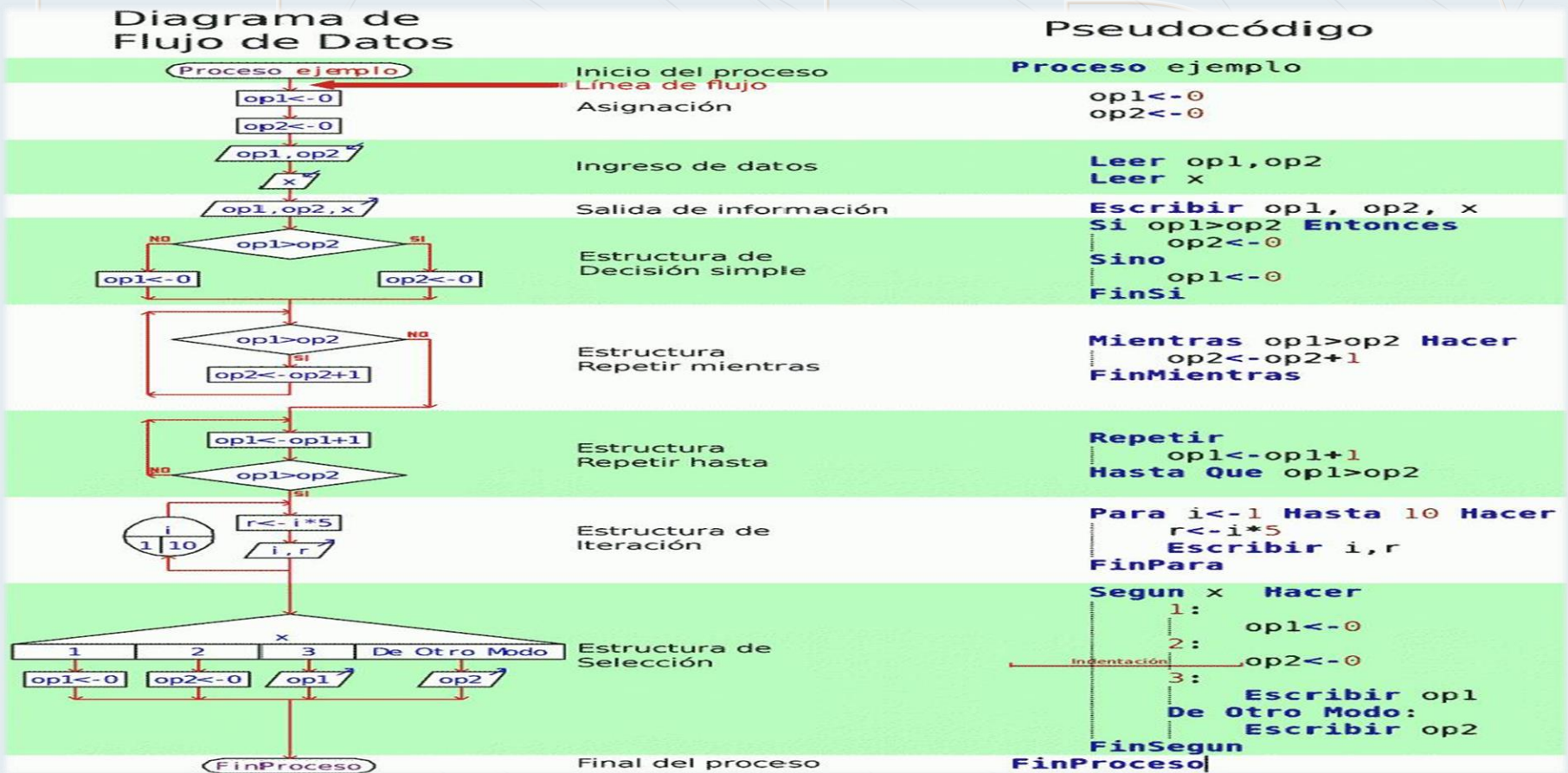
Metodología de la Programación.

El **Pseudocódigo** es una técnica de representación de los elementos y estructuras básicas de programación para describir algoritmos, mediante lenguaje pseudo-natural, con convenciones formales pero entendible por los humanos, de formato más o menos libre o flexible.

INICIO, FIN O PARADA	PROGRAMA	ALTERNATIVA SIMPLE	ALTERNATIVA DOBLE
INICIO / FIN / PARADA <programa>	INICIO <programa> <inicializa_serie> <instrucciones> FIN <programa>	SI <expresión> ENTONCES <instrucciones> FINSI	SI <expresión> ENTONCES <instrucciones> SINO ENTONCES <instrucciones> FINSI
PROCESAMIENTO O TRATAMIENTO	SECUENCIA	ALTERNATIVA MÚLTIPLE	
<variable> ← <expresión>	<instrucción> ... <instrucción>	SI <EXPRESIÓN> ENTONCES <instrucciones> SINO SI <EXPRESIÓN> ENTONCES <instrucciones> ... SINO ENTONCES <instrucciones> FINSI	SEGÚN <expresión> SI <VALOR>: <instrucciones> ... SINO ENTONCES: <instrucciones> FINSEGÚN
LEER "mensaje", <variable>, ...			
ESCRIBIR "mensaje", <variable>, ...			
IMPRIMIR "mensaje", <variable>			
<subprograma> (<parámetros>)			
CONDICIÓN O DECISIÓN		ITERACIÓN MIENTRAS	ITERACIÓN HASTA
<expresión>		MIENTRAS <expresión> <instrucciones> REPETIR	REPETIR <instrucciones> HASTA <expresión>
COMENTARIOS		ITERACIÓN PARA	
* <comentario>		PARA <paso> ← <ini> A <fin> EN <val> <instrucciones> REPETIR	paso ← <ini> MIENTRAS paso <> <fin> <instrucciones> paso ← paso ± <val> REPETIR

Metodología de la Programación.

PselInt (PSEudocódigo INTérprete) es un programa capaz de ejecutar la solución de un algoritmo implementada mediante pseudocódigo o diagramas de flujo. Es una herramienta educativa ampliamente utilizada principalmente en entornos educativos para aprender los fundamentos de la programación y el desarrollo de la lógica.



Referencias.

Qué es un algoritmo informático: características, tipos y ejemplos

<https://profile.es/blog/que-es-un-algoritmo-informatico/>