PROGRAMACIÓN.
U3 - INTRODUCCIÓN A LA PROGRAMACIÓN.

ÍNDICE

I. INTRODUCCIÓN

- ¿ Qué es un sistema informático y cómo identificar los bloques que componen ?
- ¿ Qué es un programa y un algoritmo, los lenguajes de programación que existen, junto a sus características y su organización ?
  - Definición de los conceptos de datos, algoritmo y programa.
  - ¿ Cómo expresar un algoritmo ?
  - Diferentes tipos de lenguajes de programación.
  - Herramientas y entornos para el desarrollo de un programa. Ejemplos de entornos de desarrollo.

II. CONCEPTOS

CONCEPTOS: DATOS, ALGORITMO, PROGRAMA Y PROGRAMACIÓN.

DATOS:

- Los datos son la representación simbólica, bien sea mediante números o letras, de una información que facilita la deducción de una investigación o un hecho. En definitiva, son la información que se introduce en los computadores.

ALGORITMO:

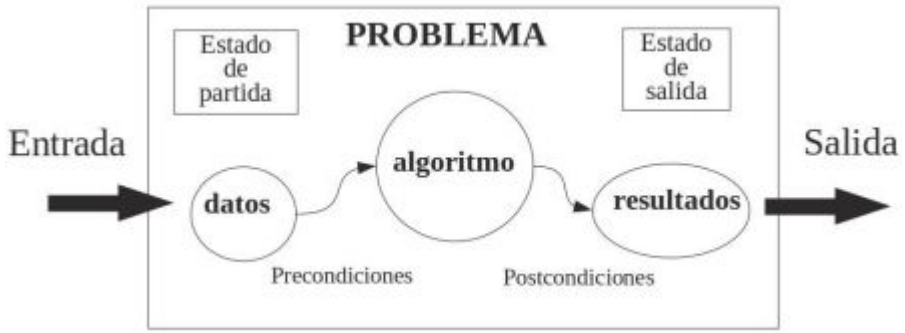
- Un algoritmo es un conjunto ordenado y finito de operaciones o reglas que permite hallar la solución de un problema.
- Son las instrucciones o pasos consecutivos con el fin de que la máquina realice la actividad definida.
- EXPRESIONES:
  - LENGUAJE NATURAL.
  - DIAGRAMA DE FLUJO.
  - PSEUDOCÓDIGO.
  - LENGUAJE DE PROGRAMACIÓN
- CARACTERÍSTICAS:
  - **PRECISOS**: Establecen una serie de reglas que detallan paso a paso lo que hay que hacer.
  - **MISMOS RESULTADOS**. Dado los mismos datos y premisas se deben obtener los mismos resultados.
  - **MÉTODO SISTEMÁTICO**. Desarrollo sistemático (unión de elementos dispersos formando un todo).
  - **INICIO Y FIN**. Los pasos definidos tienen un principio y un fin bien acotado (nº finito de pasos).

II. CONCEPTOS

CONCEPTOS: DATOS, ALGORITMO, PROGRAMA Y PROGRAMACIÓN.

PROGRAMA. Conjunto de elementos implicadas en:

- **La descripción. ¿** Qué datos utilizar de entrada y salida ? ¿ Qué pasos ha realizar ?
- **Desarrollo.** Construcción del algoritmo.
- **Implementación.** Secuenciación de las acciones definidas en el algoritmo y su interacción con los datos.

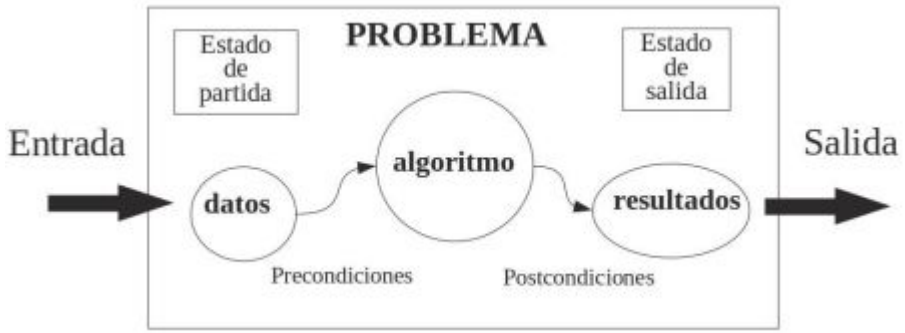


II. CONCEPTOS

CONCEPTOS: DATOS, ALGORITMO, PROGRAMA Y PROGRAMACIÓN.

PROGRAMA. Conjunto de elementos implicadas en:

- **La descripción.** ¿ Qué datos utilizar de entrada y salida ? ¿ Qué pasos ha realizar ?
- **Desarrollo.** Construcción del algoritmo.
- **Implementación.** Secuenciación de las acciones definidas en el algoritmo y su interacción con los datos.



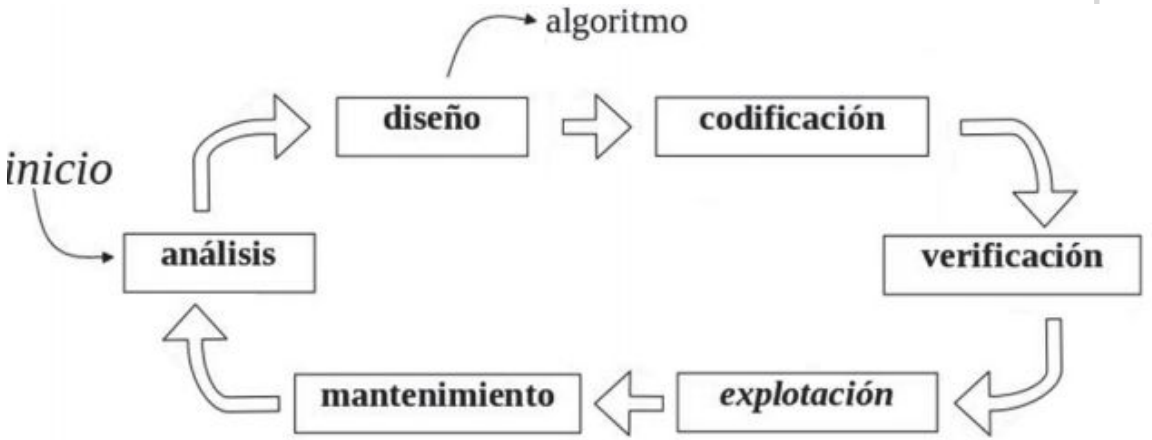
II. CONCEPTOS

CONCEPTOS: DATOS, ALGORITMO, PROGRAMA Y PROGRAMACIÓN.

PROGRAMACIÓN. Establece la normas a seguir para la creación del código fuente de un programa.

Los pasos que se han de seguir para obtener un buen programa se pueden ver en la imagen que aparece a continuación, donde se distinguen:

1. Análisis de problema.
   a. Identificar la forma de resolverlo.
   b. Establecer objetivos que se quiere alcanzar.
   c. Definición correcta de los elementos que intervienen.
2. Diseñar el algoritmo: Definir los pasos ha realizar para alcanzar los objetivos.
3. Codificación.
   a. Escoger el lenguaje de programación más adecuado.
   b. Convertir los pasos definidos en instrucción interpretables por el PC.
4. Ejecución: Ejecución de los pasos y verificación de su funcionamiento.
5. Explotación y mantenimiento.
   a. Fase de explotación o uso por parte del cliente.
   b. Fase de mantenimiento ante ampliaciones, modificaciones o corrección de bugs.



III. PARADIGMAS DE PROGRAMACIÓN.

PARADIGMA DE PROGRAMACIÓN.

Un paradigma de programación indica un método de realizar cómputos 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, necesariamente, se encuadran en uno o varios paradigmas.

TIPOS:

1. **Programación imperativa o por procedimiento.** Serie de instrucciones o sentencias que definen cambios en el estado de un programa. Las sentencias se agrupan en procedimientos que tiene una tarea específica. El programador codificará los algoritmos mediante secuencias de pasos bien definidos. **C, Basic, Pascal...**
2. **Programación orientada a objetos.** Se utilizan clases a partir de las cuales se obtienen los objetos y de las interacciones entre ellos. Cada clase contiene una estructura de datos y algoritmos propios que determinan las operaciones que puede realizar. **C++, Java, Python**
3. **Programación dirigida por eventos.** Paradigma de programación en el que tanto la estructura como la ejecución de los programas están determinados por los sucesos que ocurran en el sistema. Estos eventos pueden activarse por el usuario o por ellos mismos. **JavaScript, NodeJS**
4. **Declarativo.** Basada en describir el problema declarando propiedades y reglas que deben cumplirse, en lugar de instrucciones. Es un paradigma enfocado a la descripción del problema, no cómo resolverlo. **Prolog.**
5. **Funcional**. Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida. **Lisp**

IV. LENGUAJES DE PROGRAMACIÓN.

LENGUAJES DE PROGRAMACIÓN.

Los lenguajes de programación son el medio de comunicación con el computador para poder expresar un algoritmo y solucionar el problema.

Tipos:

- **Lenguaje máquina.** Es el único que entiende directamente el computador. Su codificación se realiza mediante cadenas binarias (de ceros y unos), que especifican una serie de operaciones y las posiciones de memoria implicadas en la operación.
  - **PROS**: Posibilidad de cargar un programa en memoria sin traducción previa.
  - **CONTRAS:**
    - Dificultad y lentitud de la codificación.
    - Dificultad para verificar y poner a punto (depurar) el código.
    - Dependencia del procesador con el cuál se comunica y fue desarrollado. PORTABILIDAD NULA.
- **Lenguaje ensamblador o de bajo nivel.** Su programación consiste en indicar al computador las operaciones que ha de realizar mediante unos códigos nemotécnicos (un número limitado de códigos) : MOVE 1, n : Cargar el valor 1 en la variable “n”.
  - **PROS**: Más fácil de programar & legible que lenguaje máquina y ejecución más rápida que con lenguajes de alto nivel.
  - **CONTRA: PORTABILIDAD REDUCIDA.** Dependencia a la procesador con la que se comunica el programa original - intercambio de operaciones-

IV. LENGUAJES DE PROGRAMACIÓN.

LENGUAJES DE PROGRAMACIÓN.

Los lenguajes de programación son el medio de comunicación con el computador para poder expresar un algoritmo y solucionar el problema.

Tipos:

- **Lenguaje de alto nivel. S**on los más utilizados y están orientados al problema.
  - PROS:
    - Se basa en reglas sintácticas similares a los lenguajes humanos. Facilita la documentación y depuración.
    - Independientes de la máquina (PORTABILIDAD). Necesitando traducción al lenguaje del procesador:
      - Compilador. Analiza que este correctamente escrito.
      - Montador. Combina librerías con el programa a compilar.
      - Ensamblador. Genera código máquina.
  - CONTRA:
    - Pérdida de eficiencia frente lenguaje máquina. Supondrá un aumento significativo de la ocupación de memoria.
    - Incremento tiempo de ejecución/procesamiento.

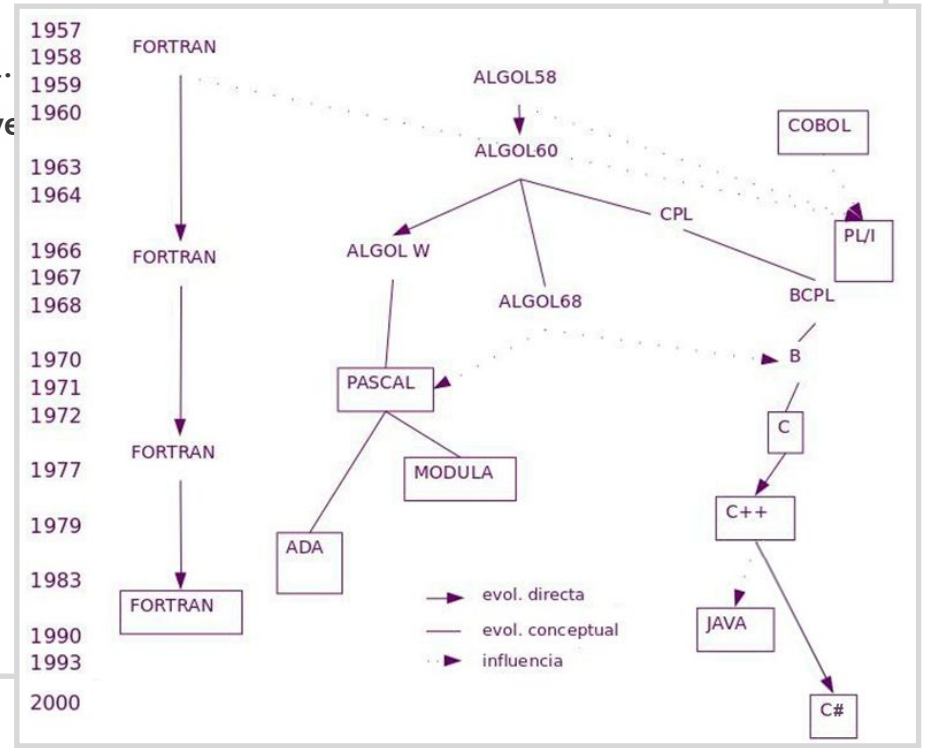
IV. LENGUAJES DE PROGRAMACIÓN.

LENGUAJES DE PROGRAMACIÓN.

Los lenguajes de programación son el medio de comunicación con el computador para poder expresar un algoritmo y solucionar el problema.

Tipos:

- Lenguaje interpretado de alto nivel. Tienen las mismas características que los de alto nivel.
  - **La principal diferencia es que van ejecutando el programa a la vez que lo van leye**
- Ejecuta el programa a la vez que lo va leyendo.
- **PROS:**
  - Multiplataforma.
  - No hay que compilarlo.
  - Ocupan menos memoria.
- **CONTRA:**
  - Su ejecución es más lenta.
  - Son menos eficientes que los lenguajes compilados.



IV. LENGUAJES DE PROGRAMACIÓN.

LENGUAJES DE PROGRAMACIÓN. GENERACIONES

1. **Primera generación.** La primera generación son los lenguajes máquina. Programas en código binario.
2. **Segunda generación.** La segunda generación son los lenguajes simbólicos o ensambladores. Al igual que los de primera generación, también dependen del ordenador que se está utilizando, aunque el código es más legible.
3. **Tercera generación.** La tercera generación son los lenguajes de alto nivel. Los códigos utilizados ya no dependen de la máquina y son parecidos al lenguaje humano o matemático.
4. **Cuarta generación.** La cuarta generación son los lenguajes de tipo SQL para acceso a base de datos. Presentan una alta portabilidad.
5. **Quinta generación.** La quinta generación son los lenguajes simbólicos dirigidos a la inteligencia artificial (IA)..

IV. LENGUAJES DE PROGRAMACIÓN.

LENGUAJES DE PROGRAMACIÓN. Requisitos de los lenguajes de programación.

1. Debe ser sencillo, claro, coherente.
2. Debe ser eficiente, rápido en ejecución, consumiendo la menor cantidad posible de memoria.
3. Debe disponer de un entorno de programación cómodo e informar de los errores del compilador de forma clara y concisa.
4. Debe tener librerías con cantidad de situaciones ya resueltas y con alta fiabilidad.
5. Universalidad, todo problema computable debe poderse codificar.
6. Naturalidad, cercano al lenguaje natural empleado en el campo de aplicación.
7. Capacidad de representar los datos, asignación y operaciones fundamentales.
8. Capacidad de controlar que un paso se ejecute de forma correcta.

V. HERRAMIENTAS Y ENTORNOS PARA EL DESARROLLO DE PROGRAMAS.

1. INTÉRPRETES Y COMPILADORES.

a. INTÉRPRETE.

Un intérprete es un traductor que toma un programa fuente, lo traduce y lo ejecuta a continuación. Además, no se genera un programa escrito en lenguaje máquina.

b. COMPILADOR.

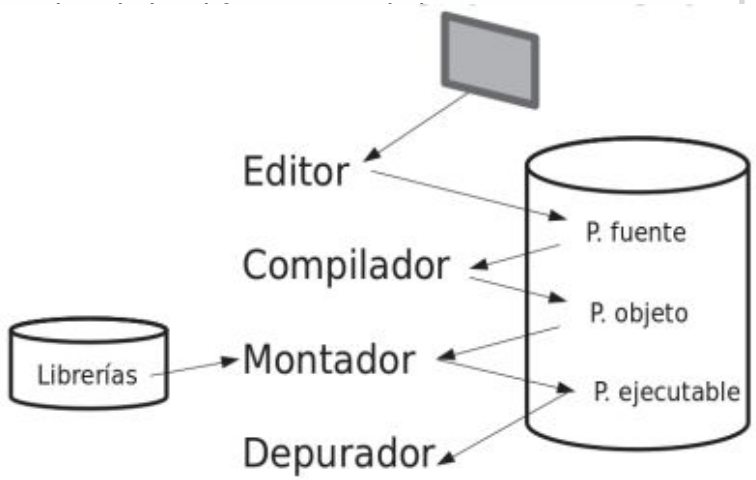
Un compilador es un programa que traduce los programas fuentes escritos en lenguaje de alto nivel a lenguaje máquina. La velocidad de ejecución de programas compilados es superior a la de los programas interpretados.

2. HERRAMIENTAS Y ENTORNOS PARA EL DESARROLLO DE PROGRAMAS.

El entorno de desarrollo es el banco de trabajo del programador. Las actividades soportadas son la codificación y las programa, es decir, da soporte a las actividades de la fase de codificación (preparación del código y prueba de unida

2.1. ELEMENTOS QUE COMPONEN UN ENTORNO DE DESARROLLO:

- **Editor.** Editor de texto donde escribir el código.
- **Compilador.** Traduce el código fuente a código objeto.
- **Montador.** Combina el fichero objeto con ficheros objeto de las librerías utilizadas. Genera el fichero ejecu
- **Depurador (debugger).** Comprueba que el ejecutable es correcto y información de la línea que presenta una en caso de fallo.



VI. HERRAMIENTAS Y ENTORNOS PARA EL DESARROLLO DE PROGRAMAS.

1. **INTÉRPRETES Y COMPILADORES.**
2. **HERRAMIENTAS Y ENTORNOS PARA EL DESARROLLO DE PROGRAMAS.**

2.2. TIPOS DE ENTORNOS:

- **Entornos centrados en un lenguaje.** Específicos para un lenguaje de programación con el que están fuertemente integrados.
- **Entornos orientados a estructura.**
  - El editor de código no es un editor de texto, sino un editor de estructura (sintáctico) y se basa en representar internamente el código fuente como una estructura.
  - La presentación externa del código es en forma de texto.
- **Entornos colección de herramientas.** Tienen una integración débil y un conjunto de elementos relativamente heterogéneos. Son fáciles de ampliar o adaptar mediante nuevas herramientas.
- **Entornos multilenguaje.** Entornos genéricos (NetBeans, Eclipse) pero no se combinan lenguajes en un mismo programa.
- **Entornos específicos.** Combinar módulos de lenguajes, como Ada y C++.

3. **Máquinas virtuales**

- Permiten el desarrollo y ejecución de programas sobre plataformas que pueden ser distintas a la utilizada por el usuario.
- En el caso de Java, cualquier programa escrito en este lenguaje podrá ser ejecutado desde cualquier ordenador, independientemente de su arquitectura, siempre que se tenga instalada la correspondiente máquina virtual.

VII. ERRORES Y CALIDAD DE LOS PROGRAMAS

LENGUAJES DE PROGRAMACIÓN.

Las pruebas de software son la técnica principal de verificación y validación. Esto implica la ejecución del programa con datos similares a los reales. Los defectos se descubren analizando la salida del programa y, una vez detectados los errores, viene la fase de depuración, que es el proceso que localiza y corrige los errores detectados.

- Se comprueba que hace lo esperado y no se comprueben situaciones anormales.
- Se comprueba la fiabilidad del programa a través de los resultados impresos o almacenados en diferentes archivos.
- Se interviene el código con sentencias de impresión para ver si las variables van tomando los valores que se supone deben tomar.
- No se prueban condiciones inválidas.
- No se piensa la prueba ya que se recurre a GUI, Debuggers... para que ayuden a realizar la prueba.
- Cuando se corrige un fallo, solo se prueba lo corregido.
- No cometer errores en el momento de la realización de las pruebas.

VIII. ACTIVIDADES.

ACTIVIDAD 1 : LOCALIZA LOS CONCEPTOS DE PROGRAMACIÓN.

ACTIVIDAD 2: RELACIONA LOS LENGUAJES DE DE PROGRAMACIÓN.

VIIII. EJERCICIOS OBLIGATORIOS.

ENUNCIADO (2 cuestiones):

- **EN EL CASO DE LENGUAJE COMPILADO.**
  ¿Qué es un compilador? ¿Qué es semántica?. Analizador léxico ¿Qué es?¿Para qué sirve?¿Cuál es su misión en un compilador?. Analizador sintáctico ¿Qué es?¿Para qué sirve? ¿Cuál es su misión en un compilador?
- **DIFERENCIAS ENTRE LENGUAJES DE PROGRAMACIÓN.**
  Da unos ejemplos significativos de ellos: Lenguaje de alto nivel, lenguaje de bajo nivel, lenguaje máquina.

CARACTERÍSTICAS:

- Ambos textos unificados en único documento que se envía al tutor para su calificación.

DUDAS/ACLARACIONES:

- **¿ Es necesario referenciar y citar y/o utilizar alguna norma (APA, Vancouver..) concreta ?** Es obligatorio indicar los fuentes utilizadas y evitar en todo momento las citas textuales, en su lugar utilizar usar “parafraseo”. En el caso de norma, cualquier normativa será válida, siendo el objetivo principal que se identifique la fuente utilizada.
- **¿ De donde debemos obtener la bibliografía ?** Google Scholar, Researchgate, libros relacionados con la materia disponibles por internet.

¿ DUDAS ?

¿ Dudas ?


IMF
Formación
Profesional
MUCHAS GRACIAS