

INTRODUCCIÓN

Esteban Álvarez

Programa

¿Qué es un programa?

- Desde el punto de vista de un usuario
- Desde el punto de vista de un programador

Programa

- Generalmente, la primera razón que mueve a una persona hacia el aprendizaje de la programación es utilizar **el ordenador** como herramienta para resolver **problemas** concretos.

Programa

- Resolución de un problema en la vida real:

En la vida real...
Observación de la situación o problema.
Pensamos en una o varias posibles soluciones.
Aplicamos la solución que estimamos más adecuada.

Programa

- Resolución de un problema en programación:

En Programación...
Análisis del problema: requiere que el problema sea definido y comprendido claramente para que pueda ser analizado con todo detalle.
Diseño o desarrollo de algoritmos: procedimiento paso a paso para solucionar el problema dado.
Resolución del algoritmo elegido en la computadora: consiste en convertir el algoritmo en programa, ejecutarlo y comprobar que soluciona verdaderamente el problema.

Algoritmo

- Palabra clave: algoritmo
- ¿Qué es un algoritmo?
 - [Vídeo 1](#)

Algoritmo

- Escribe un algoritmo para calentar una taza de café en el microondas
- Analiza si puedes detallar más ese algoritmo
- ¿Es posible que existan varios algoritmos diferentes para un mismo problema?

Algoritmo y programa

- Un algoritmo es independiente de un lenguaje de programación al igual que puedes escribir el algoritmo para lavarte los dientes en cualquier idioma y utilizar una libreta, bolígrafo, lápiz o un documento de texto electrónico
- Un programa aplica un algoritmo utilizando un lenguaje de programación con el objetivo de resolver un problema
- El lenguaje de programación es solo un medio para expresar un algoritmo y el ordenador un procesador para ejecutarlo

Algoritmo

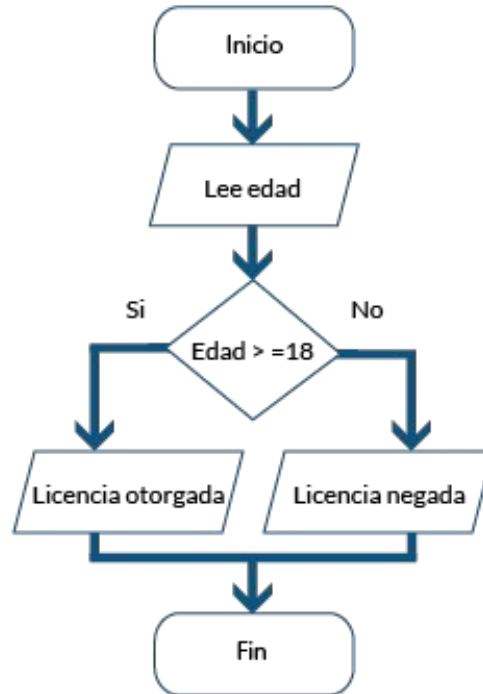
- En esencia, todo problema se puede describir por medio de un algoritmo y sus características fundamentales son:
 - Debe ser **preciso** e indicar el orden de realización paso a paso.
 - Debe estar **definido**, si se ejecuta dos o más veces, debe obtener el mismo resultado cada vez.
 - Debe ser **finito**, debe tener un número finito de pasos.

Algoritmo

- Si los problemas son muy complejos, se necesita descomponerlos en problemas más pequeños
- Métodos:
 - Diseño descendente. Descomposición del problema en problemas más sencillos de resolver
 - Diseño modular (top-down). Dividir un problema en módulos más pequeños o sub-programas.
- Se basan en la teoría de “divide y vencerás”

Algoritmo. Representación

- Para representar gráficamente un algoritmo podemos usar:
 - Diagramas de flujo



Algoritmo. Representación

- Para representar gráficamente un algoritmo podemos usar:
 - Pseudocódigo. Es lo que hemos hecho en el algoritmo para calentar en el microondas. Es el método más utilizado

En realidad se adaptan las palabras a acciones que se pueden realizar con los lenguajes de programación:

```
1  Proceso sin_titulo
2      Definir num Como Entero;
3      Escribir 'Ingrese un número';
4      Leer num;
5      Si num>0 Entonces
6          Escribir 'El número es positivo';
7      Sino
8          Escribir 'El número es negativo';
9      FinSi
10 FinProceso
```

Algoritmo. Representación

- Para representar gráficamente un algoritmo podemos usar:
 - Tablas de decisión. En una tabla son representadas las posibles condiciones del problema con sus respectivas acciones.

Suele ser una técnica de apoyo al pseudocódigo cuando existen situaciones condicionales complejas

Algoritmos. Conclusión

- [Vídeo 2](#)

El algoritmo es **el elemento más importante** de la programación

EJERCICIOS

2.1. Diseñar una solución para resolver cada uno de los siguientes problemas y tratar de refinar sus soluciones mediante algoritmos adecuados:

- a)* Realizar una llamada telefónica desde un teléfono público.
- b)* Cocinar una tortilla.
- c)* Arreglar un pinchazo de una bicicleta.
- d)* Freír un huevo.

2.2. Escribir un algoritmo para:

- a)* Sumar dos números enteros.
- b)* Restar dos números enteros.
- c)* Multiplicar dos números enteros.
- d)* Dividir un número entero por otro.

2.3. Escribir un algoritmo para determinar el máximo común divisor de dos números enteros (MCD) por el algoritmo de Euclides:

- Dividir el mayor de los dos enteros positivos por el más pequeño.
- A continuación dividir el divisor por el resto.
- Continuar el proceso de dividir el último divisor por el último resto hasta que la división sea exacta.
- El último divisor es el mcd.

2.4. Diseñar un algoritmo que lea y visualice una serie de números distintos de cero. El algoritmo debe terminar con un valor cero que no se debe visualizar. Visualizar el número de valores leídos.

Paradigmas de programación

- Son los principios fundamentales de la programación del software



Fases de la programación

1. Fase de resolución del problema
2. Fase de implementación
3. Fase de explotación y mantenimiento

Fases de la programación

1. Fase de resolución del problema
 - a. Análisis. Ya hemos visto que se necesita analizar el problema viendo las necesidades del cliente y valorando las posibles soluciones.
 - b. Diseño. En base al análisis realizado, se desarrolla el algoritmo.

Fases de la programación

2. Fase de implementación

- Esta etapa consiste en traducir los resultados obtenidos a un determinado lenguaje de programación
- Pero para que nuestro programa comience a funcionar, antes debe ser traducido a un lenguaje que la máquina entienda (código máquina o código objeto)
 - Es aquí donde aparecen los términos compilación, compilador e interpretación



Fases de la programación

3. Fase de explotación y mantenimiento

- Cuando el programa ya está instalado en el sistema y está siendo de utilidad para los usuarios, decimos que se encuentra en fase de explotación.
- Periódicamente será necesario realizar evaluaciones y, si es necesario, llevar a cabo modificaciones para que el programa se adapte o actualice a nuevas necesidades, pudiendo también corregirse errores no detectados anteriormente. Este proceso recibe el nombre de mantenimiento del software.

Ciclo de vida del software

- En ingeniería del software se utiliza el término ciclo de vida para referirse a los estados por los que pasa una aplicación a lo largo de su “vida”
- Existen varios tipos de ciclo de vida. Busca información sobre ellos

Lenguajes de programación

- Hemos visto que es la forma que tenemos de dar a la máquina las instrucciones para aplicar un algoritmo
- Se puede establecer una clasificación dependiendo de lo cerca que estén del lenguaje humano o de los ordenadores:
 - Lenguaje máquina
 - Lenguaje ensamblador
 - Lenguajes de alto nivel
 - Compilados
 - Interpretados
 - Pseudo-compilados o pseudo-interpretados

Lenguajes de programación

Lenguaje máquina

- Este es el lenguaje utilizado directamente por el procesador
- Consta de un conjunto de instrucciones codificadas en binario (ceros y unos)
- Fue el primer lenguaje de programación
- Tiene muchas limitaciones y es muy complicado

Lenguajes de programación

Lenguaje ensamblador

- Es la evolución del lenguaje máquina
- Las secuencias de ceros y unos se sustituyen por códigos de operación (ADD, MUL, DIV)
- Sigue siendo dependiente de la máquina en que se ejecute
- Aún de muy bajo nivel

Lenguajes de programación

Lenguajes de alto nivel

- Se acercan al lenguaje humano
- Son independientes del procesador en el que se ejecutan
 - **Lenguajes compilados.** Se escriben en alto nivel y se compilan (traducen) a lenguaje máquina. Pascal, Fortran, C, C++
 - **Lenguajes interpretados.** Se habla de interpretación en lugar de compilación porque se crea un ejecutable que no se guarda en memoria, sino que se ejecuta directamente. Perl, PHP, Python, Javascript
 - **Lenguajes pseudo-compilados.** Se genera un código intermedio (bytecodes) que debe ser interpretado por una aplicación. Es el caso de java y la máquina virtual de java



El lenguaje Java

Características principales:

- El código generado por el compilador Java es independiente de la arquitectura
- Está totalmente orientado a objetos
- Su sintaxis es similar a C y C++
- Es distribuido, preparado para aplicaciones TCP/IP
- Dispone de un amplio conjunto de bibliotecas
- Es robusto, realizando comprobaciones del código en tiempo de compilación y de ejecución
- La seguridad está garantizada, ya que las aplicaciones Java no acceden a zonas delicadas de memoria o de sistema.

Estructura de un programa Java

```
public class Hola {  
    public static void main(String[] args) {  
        System.out.println("Hola, ");  
        System.out.println("me llamo Angel");  
        System.out.println("Hasta luego");  
    }  
}
```

Estructura de un programa Java

```
public class Hola { Hola es una clase
    public static void main(String[] args) {
        System.out.println("Hola, ");
        System.out.println("me llamo Angel");
        System.out.println("Hasta luego");
    }
}
```

Estructura de un programa Java

```
public class Hola {
```

```
    public static void main(String[] args) {  
        System.out.println("Hola, ");  
        System.out.println("me llamo Angel");  
        System.out.println("Hasta luego");  
    }
```

```
}
```

Estructura de un programa Java

```
public class Hola {  
    public static void main(String[] args) {
```

Main es el programa principal. Todo nuestro código va a ir aquí.

System.out.println("me llamo Angel");
System.out.println("Hasta luego");

```
}
```

```
}
```

Estructura de un programa Java

```
public class Hola {  
    public static void main(String[] args) {  
        System.out.println("Hola, ");  
        System.out.println("me llamo Angel");  
        System.out.println("Hasta luego");  
    }  
}
```

Dentro de main hay tres **instrucciones**

Cada una de ellas acaba con punto y coma (;)

Estructura de un programa Java

- Cuidado!! Java distingue entre mayúsculas y minúsculas.

Instalación

- Instalaremos el entorno **Eclipse** por ser más popular en el mundo empresarial
 - Pero existen otras opciones igual de buenas como NetBeans que es el oficial de Oracle, IntelliJ, Visual Studio Code...
- El proceso es sencillo
 - Descargar el instalador de la página oficial de Eclipse
 - Instalar y listo