

Informática I

La Documentación de Programas

En la ejecución de un proyecto informático o un programa software se deben de seguir una serie de pasos desde que se plantea el problema hasta que se dispone del programa o de la aplicación funcionando en el ordenador.

Los pasos son los siguientes:

- Análisis de factibilidad
- Análisis de requerimientos
- Diseño del sistema
- Implementación
- Validación y pruebas
- Explotación
- Mantenimiento

La Documentación de Programas

- ▶ Cada uno de estos pasos debe de llevar asociado un documento. Estos documentos son muy importantes ya que van a regir las fases del ciclo de vida del software y se recogen los pasos seguidos en cada fase para su ejecución.
- ▶ No es viable la solución mostrada por algunos programadores de ir directamente a la implementación sin antes pararse en la fases 1, 2 y 3. Un trabajo deficiente en estas fases supone una mala definición del problema y por tanto el sistema no cumplirá seguramente con todos los requisitos. El diseño del sistema no será efectivo y los errores serán de difícil solución.
- ▶ Por lo tanto en la realización de las prácticas será obligado a complementar un formulario que guiará al alumno en la fase de análisis de requisitos y de diseño.

El Documento de Especificaciones

- ▶ Este documento tiene como objeto asegurar que tanto el desarrollador como el cliente tienen la misma idea sobre las funcionalidades del sistema. Es muy importante que esto quede claro ya que si no el desarrollo software no será aceptable.

¿Qué es una ERS en ingeniería del software?

- ▶ La ERS es un contrato entre la empresa desarrolladora y la empresa cliente. Ambas partes deben comunicarse muy estrechamente para establecer los requisitos de la aplicación. Una buena ERS ayudará a la empresa cliente a describir qué es lo que quiere, y también servirá a los desarrolladores para comprender qué es, exactamente, lo que le están pidiendo. De la ERS depende el resultado final del producto software. Por tanto, es muy importante que en ella se describa lo más detalladamente posible "todo" lo que se espera que haga dicho software.

Según la norma IEEE 830, un ERS debe contener los siguientes puntos:

I. Introducción (Se definen los fines y los objetivos del software)

- ▶ Referencia del sistema.
- ▶ Descripción general.
- ▶ Restricciones del proyecto.

II. Descripción de la **información** (Descripción detallada del problema, incluyendo el HW y SW necesario)

1. Representación del flujo de la información.

- a. Flujo de datos
- b. Flujo de control

2. Representación del contenido de la información.

3. Descripción de la interfaz del sistema.

Según la norma IEEE 830, un ERS debe contener los siguientes puntos:

III. Descripción **funcional** (Descripción de cada función requerida, incluyendo diagramas).

A. Partición funcional

B. Descripción funcional

- ▶ Narrativa de procesamiento
- ▶ Restricciones/Limitaciones.
- ▶ Requisitos de rendimiento.
- ▶ Restricciones de diseño
- ▶ Diagramas de soporte

C. Descripción del control

- ▶ Especificación del control
- ▶ Restricciones de diseño



Según la norma IEEE 830, un ERS debe contener los siguientes puntos:

IV. Descripción del **comportamiento** (comportamiento del SW ante sucesos externos y controles internos)

- a) Estados del sistema.
- b) Sucesos y acciones.

V. Criterios de validación.

- a) Límites de rendimiento
- b) Clases de pruebas
- c) Respuesta esperada del SW
- d) Consideraciones especiales

El documento de diseño

- ▶ En la fase de diseño se toman aquellas decisiones relativas a la futura implementación, se decide la estructura de datos a utilizar, la forma en que se van a implementar las distintas estructuras, el contenido de las clases (sus métodos, los atributos, ...), los objetos. También se definen las funciones, sus datos de entrada y salida, que tarea realizan, para alguna de especial interés el algoritmo que soluciona el problema.
- ▶ El flujo del programa se define mediante una serie de gráficos que permiten visualizar cual es la evolución del sistema software, en caso de orientación de objetos existen el diagrama de clases, el diagrama de importante tenerlo claro para ello existen una serie de diagramas que permitan clarificar este asunto.

La documentación del código fuente

- ▶ Durante la fase de implementación, cuando se está programando, es necesario comentar convenientemente cada una de las partes que tiene el programa. Estos comentarios se incluyen en el código fuente con el objeto de clarificar y explicar cada elemento del programa, se deben de comentar las clases, las variables, los módulos y en definitiva todo elemento que se considere importante.

La documentación del código fuente

- ▶ Esta documentación tiene como objeto hacer más comprensible el código fuente a otros programadores que tengan que trabajar con él, ya sea porque forman parte del grupo de desarrollo, el programa va a ser mantenido o modificado por otra persona distinta al programador inicial. También resulta muy útil durante la depuración y el mantenimiento del programa por el propio programador, al paso del tiempo las decisiones se olvidan y surgen dudas hasta en el propio programador de porqué se hicieron las cosas de una determinada manera y no de otra.

Formulario de prácticas

- ▶ Antes del comienzo de cada práctica es necesario haber realizado un primer estudio del problema a resolver durante la sesión, para ello es obligatorio el complimentar este formulario. El objetivo del documento es asegurar que el alumno ha analizado la práctica y ha madurado suficientemente el problema como para estar capacitado para afrontar la codificación del programa.

Estructura de programa

► ¿Cómo funciona un programa informático?

El comportamiento de un programa depende del tipo de lenguaje usado para escribirlo...

En cualquier caso, casi todos los lenguajes de programación se basan en el mismo principio: el programa consiste en un conjunto de comandos que el equipo debe ejecutar. El equipo ejecuta los comandos a medida que va leyendo el archivo (de arriba hacia abajo) hasta que alcanza un comando (al que a menudo se hace referencia como un comando ramificado) que ordena al equipo ir a una ubicación específica del programa. En cierto modo, se parece a la búsqueda del tesoro, donde el equipo debe seguir las pistas y ejecutar los comandos hasta alcanzar el final del programa y detenerse.

El concepto de variables


- ▶ Las variables se utilizan en la mayoría de los lenguajes, donde un nombre se asocia a un contenido. Por ejemplo, podemos llamar a una variable "maniquí" y asignarle el valor 8.

Tipos de datos

- ▶ Algunos lenguajes permiten asociar cualquier tipo de dato con una variable (ya sea un número completo o un carácter), dicho lenguaje se denomina **lenguaje sin tipos**.
- ▶ El tipo de datos establece el número de bytes en que el dato está codificado, es decir, la cantidad de memoria que este dato ocupa, así como el formato en el que está representado. Por esta razón, los lenguajes evolucionados (C, Java) son **lenguajes con tipo**, lo que significa que una variable no se asocia solamente con un nombre sino también con un tipo de datos que debe especificarse cuando se declara la variable; es decir, cuando el nombre de la variable se escribe por primera vez, debe especificar en el compilador el tipo de datos que esta variable contendrá (la manera de declarar la variable dependerá del lenguaje utilizado).

Sintaxis

- ▶ Los lenguajes de programación requieren una sintaxis estricta, por lo tanto, no es posible escribir de la manera que se desea.
- ▶ Así, algunos lenguajes **distinguen entre minúsculas y mayúsculas**, lo que significa que un nombre escrito en minúscula no es considerado el equivalente del mismo nombre escrito en mayúscula. En este caso, la variable "Maniquí" sería diferente a la variable "maniquí".

- 
- ▶ Los nombres de variables requieren comúnmente una longitud máxima (dependiendo del lenguaje) y de un determinado conjunto de caracteres. Generalmente, se utilizan los siguientes caracteres:

**abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890_**

- ▶ De esta forma, un espacio (" ") se considera un tipo de carácter diferente, llamado un carácter especial. Por lo tanto, no es muy común que un lenguaje permita el uso de caracteres especiales en nombres de variables.

Palabras reservadas

- ▶ En la mayoría de los lenguajes, existe un puñado de palabras que no deben utilizarse como variables de nombres, llamadas **palabras reservadas**.

Constantes

- ▶ Las constantes son datos cuyo valor no puede modificarse. Por lo general, se definen al comienzo del programa. El valor de la constante puede ser de cualquier tipo permitido por el lenguaje de programación.

Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos.

Lenguajes de bajo nivel:

- ▶ Están basados directamente en la máquina de Von Neumann, por lo que están a un nivel muy cercano a la máquina
- ▶ Las instrucciones del lenguaje son las instrucciones del microprocesador del ordenador, que normalmente son demasiado simples.
- ▶ Es exclusivo de cada ordenador.
- ▶ Es difícil y costoso de programar.

Lenguajes de programación

Lenguaje máquina:

- ▶ Instrucciones reconocidas por los circuitos del procesador.
- ▶ Se codifican en binario.
- ▶ Los datos se referencian por su posición de memoria.

Lenguaje ensamblador:

- ▶ Codificación mnemotécnica del lenguaje máquina.
- ▶ Necesita un traductor.
- ▶ Se pueden utilizar etiquetas en vez de posiciones de memoria.

Lenguajes de programación

Lenguajes de alto nivel:

- ▶ Están basados en máquinas abstractas, que facilitan la comprensión por personas.
- ▶ Instrucciones más flexibles y potentes.
- ▶ Necesita un traductor para convertir el programa a lenguaje máquina, que es el que entiende el ordenador.
- ▶ No depende del procesador, por lo que el mismo programa sirve para diferentes Ordenadores.
- ▶ Al tener que traducirlo, es más lento e ineficiente que el lenguaje de bajo nivel.

¿Preguntas?

