Tema 1.1

La Ingeniería del Software estudia la creación de software ,rentable, confiable y de calidad

Son todas las teorías,métodos y herramientas que se utilizan en desarrollo de software.

El software(programas + datos) es un conjunto de programas,junto con la informacion necesaria para su operación,incluyendo documentos donde indican procedimientos de uso y funcionamiento

La programación es el método de confección de programas mediante lenguajes de programación,desde el codigo fuente hasta el codigo maquina pasanado por el codigo intermedio

El codigo objeto es un codigo bunario itermedio proximo al codigo maquina

El bytecode es un codgio binario intermedio mas abstracto e independiente del hardware que el codigo maquina interpretado por maquinas virtuales, con la ventaja de la portabilidad

Un programa es un conjunto de instruciones de procesamiento o tratamiento de datos correspondiendose con la implementacion de un algoritmo

Un Algoritmo es un conjunto de operaciones 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 ,representación de los elementos y estructuras basicas de programación

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

Software de sistema (software básico o de base).

-Software funcional:(base del sistema operativo) gestion interna de la computadora(o intermediario entre este y la computadora)

-Programas de Control (gestión de recursos + interface de usuario):

-Programas de Proceso (programas y herramientas de desarrollo de software)

-Software general

-Programas de Utilidad (utilidades o utillería)

-Programas de Servicio

Software de Aplicación (software ofimático y corporativo):

-Software a medida

-Software estándar

• Software Libre: software con libertad para ser usado, estudiado, modificado o distribuido. Puede ser gratuito (la mayoría) o de pago (una minoría). El software de código abierto u opensource disponibilidad del código fuente.

- Licencia Pública General o General Public License (GPL): licencia de software libre con código fuente bajo copyleft..

- Licencia Pública General Reducida GNU o GNU Lesser General Public License (LGPL): Licencia 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 con restricciones para ser usado, estudiado, modificado o distribuido, . El software propietario o privativo es software no libre comercial con restricciones impuestas por copyrigt.

- Freeware: software privativo que se distribuye, sin código fuente, gratuitamente y con funcionalidad completa, pero con restricciones .

- Shareware: software privativo gratuito de evaluación,con imitaciones operativas o temporales frente a la versión de pago,.

• Edición → código fuente.

• Traducción.

- Análisis (léxico, sintáctico, semántico).

- Compilación / Interpretación.

- Compilación → código intermedio (objeto, bytecode, pcode). - Generación de código.

- Optimización de código.

- Enlace: estático (librerías) / dinámico (DLL) → código ejecutable.

- Interpretación → código ejecutable.

Ejecución .

Depuración

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

• Estructura Gramatical

• Estructuras de datos .

- Tipos simples o básicos o elementales o fundamentales o primitivos

- Tipos compuestos, estructurados o derivados

• 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

- Procesamiento de Datos

- Control de Flujo de Programa.

• Funciones y/o Procedimientos.

• 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).

Una Máquina Virtual de sistema es un software que emula un ordenador por completo. 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),y se traducen a instrucciones en código nativo propio de una plataforma específica para su ejecución.

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.

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

PseInt (PSEudocódigo INTérprete) es un programa capaz de ejecutar la solución de un algoritmo implementada mediante pseudocódigo o diagramas de flujo