**INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE**

1. INTRODUCCIÓN.
2. HARDWARE Y SOFTWARE.
   1. ¿Qué es un ordenador?
   2. Partes de un ordenador.
   3. Hardware
   4. Software.
      1. Tipos de software.
   5. Sistemas Operativos.
      1. Funcionalidades del SO.
   6. Licencias Software.

**Pregunta: ¿Cuál crees que es el sistema operativo que más se usa en el mundo?**

1. DESARROLLO DE SOFTWARE
   1. Definición.
   2. Etapas.

**Pregunta: Si queremos construir una aplicación pequeña, y se prevé que no sufrirá grandes cambios durante su vida, ¿qué modelo de ciclo de vida sería el más recomendable? ¿Por qué?**

1. LENGUAJE DE PROGRAMACIÓN
   1. ¿Qué es un programa?
   2. Definición de Lenguaje de Programación.
   3. Elementos.
   4. Clasificación según nivel de abstracción.
   5. Clasificación según la forma de ejecución.
   6. Clasificación según la técnica de programación.
   7. Obtención de código ejecutable.
      1. Tipos de código.
      2. Ejecución de un programa.
2. ENTORNOS DE DESARROLLO (IDE).

5.1. Componentes de un IDE.

5.1.1. Editor de texto.

5.1.2. Compilador.

5.1.3. Intérprete.

5.1.4. Depurador o debbuger.

5.1.5. Control de versiones.

5.1.6. Interfaces graficas de usuario.

5.1.7. Refactorización.

1. MÁQUINAS VIRTUALES.

6.1. Definición.

6.2. Máquinas virtuales de proceso.

6.3. Máquinas virtuales de sistema.

6.4. Ventajas y desventajas de las máquinas virtuales.

**INTRODUCCIÓN**

La ingeniería del software es el diseño, desarrollo y mantenimiento del software, es decir, la *aplicación de los principios de la ingeniería al software*. Trata de acercarse a la programación de una forma estructurada  y disciplinada, utilizando medios sistematizados y herramientas preestablecidas de la forma más eficiente posible; no es tan solo la resolución de problemas, sino de encontrar la solución más eficiente para estos.

Al principio, debido a la falta de herramientas de desarrollo, muchos proyectos fracasaban a la hora de calcular el presupuesto y las fechas de entrega. Según se creaban diferentes herramientas se fue reduciendo el tiempo y facilitando el desarrollo, creando métodos que se han ido estableciendo con el tiempo.

**4 Lenguaje de Programación:**

Un lenguaje de programación es un una medio por el cual nos comunicamos con la máquina para darle una serie de instrucciones y así lograr hacer funciones. En otras palabras, es un idioma artificial que entienden las máquinas.

Cabe destacar que existen una variedad muy amplia de lenguajes de programación: los cuales se basan unos en otros simplificando sus estructuras y facilitando el uso del mismo, he aquí unos ejemplos: Perl, C++, Swift, C, Java, PHP, Python, etc.

**4.1 ¿Qué es un programa?**

Cuando nos referimos a un programa deducimos que se trata de un software de sistema o un software de aplicación, este tiene como uso el poder realizar una tarea concreta de una manera simple. Este suele contar con una interfaz de usuario, que viene a ser una manera de que los programadores podamos visualizar nuestro progreso y trabajo. Los programas están creados mediante lenguajes de programación previamente mencionados; y a su vez estos corren sobre un sistema operativo, como puede ser: Windows, Linux, Mac OS, Android, etc.

Cabe destacar que vienen programas preinstalados en los sistemas operativos, que tienen como función controlar operaciones internas del ordenador, o bien tratar con la persona que esté utilizando el terminal para ayudarle a trabajar, navegar por la web, o ver videos de gatitos.

**5.1.4 Depurador o Debugger:**

Un depurador(Debugger) es un programa informático creado con el fin de analizar, encontrar errores y repararlos en otro programa informático. Normalmente los  depuradores suelen estar dentro de los entornos de desarrollo integrados.

**5.1.5 Control de versiones:**

La control de versiones es gestionar los diversos cambios que se le van haciendo a un programa informático, para este fin se suelen usar diferentes programas que te ayudan a llevar un mejor control sobre las versiones, Un sistema de control de versiones muy utilizado actualmente es **Git**, el cual fue diseñado por Linus Torvalds  y en un primer momento fue pensado para ponerle un front end por encima, pero al final está siendo usado como un sistema de gestión de versiones completo.

**5.1.6 Interfaces Gráficas de Usuario:**

La interfaz gráfica de es un tipo de interfaz que permite que un programa sea más intuitivo para el usuario final, la interfaz gráfica de usuario utiliza iconos, botones, menús, desplegables, selectores… Para hacer el programa informático más amigable para el usuario final.

**5.1.7. Refactorización:**

**Bibliografía**

1.

<https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software>

4.

<https://blog.dinahosting.com/los-10-lenguajes-de-programacion-mas-usados/>

<https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico>

<https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-programa/>

<https://concepto.de/programa-informatico/>

5.1.4:

<https://www.glosarioit.com/Depurador>

<https://es.wikipedia.org/wiki/Depurador>

5.1.5:

<https://es.wikipedia.org/wiki/Control_de_versiones> <https://es.wikipedia.org/wiki/Programas_para_control_de_versiones>

<https://es.wikipedia.org/wiki/Git>

5.1.6:

<https://www.zorraquino.com/diccionario/marketing-digital/que-es-interfaz-grafica-de-usuario.html>

<https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario>