Software

* Sistema operativo
* Software de programación
* Software

Programa informatico

* Conjunto de ordenes con un propósito concreto

CLASIFICACIONES

* Lenguaje de bajo nivel
  + Pueden ser en binario o hexadecimal
* Lenguaje ensamblador
  + Instrucciones básicas
  + Sobre posiciones de memoria
* Lenguajes de alto nivel
* Lenguajes visuales

Lenguajes según su interpretación

* Compilados
  + Se ejecuta todo el bloque del programa
* Interpretados
  + Se va lullendo por filas
* Lenguajes virtuales
  + Es necesario una traducción intermedia
  + Aporta universalidad
    - Como los programas los transforma una maquina virtual

Tipos de lenguaje según su forma

* Orientado a objeto
* Estructurado

Tipos de código

* Código fuente
  + Los que hace un programador
* Código objeto
  + Es una traducción que aun no es binario pero aquí ya es ensamblador
* Código ejecutado

LENGUAJES MAS USADOS

* PHP
  + No se usa para manejar gran cantidad de datos
* Java
  + Para aplicaciones wev que tratan una cantidad grande de datos pero no excesiva
  + Permiten multihilos
  + Es orientado a objetos
  + Lenguaje virtual
* Python
  + Big Data
  + Soporte para interfaces graficas
  + Lenguaje virtual
  + Es más lento
* C++
  + Muy potente
  + Tiene punteros
  + Gran uso
  + Orientado a objetos y estructurada
* Javascript
  + Mayormente usado para desarrollo web
  + Interpretado por el navegador

CICLO DE VIDA DE UNA APLICACIÓN

* En cascada
  + Se establecen requisitos
  + analisis
  + Se diseña
  + Se modifica
  + pruebas
* Modelo en cascada con realimentación
  + Da opción a volver para atrás, pero con bloques muy grandes
* Modelos evolutivos
  + Se intenta acortar y flexibilizar los procesos
  + Iterativo incremental
    - Se hacer por partaes mas compartimentadas
  + Modelo en espiral
* Fases básicas
  + Fase inicial
    - Planificación
    - Hacer estimaciones
  + Análisis
    - Requisitos
      * Funcionales
        + Características concretas
      * No funcionales
        + Tiempos de respuesta
        + Hardware
        + Legislación
  + Especificación de requisitos
    - Detallar lo necesario
      * Objetivos
      * Requisitos funcionales y no funcionales
      * Temporización del proyecto
  + Implementación
    - Modalidad y sencillez
      * Buenas practicas
    - Corrección
      * Que haga lo que se pide
      * Comentarios sencillos
      * Fácil de mantener
      * Eficiencia
  + Pruebas
    - Pruebas funcionales
      * Comprueban que hacen lo que el cliente pidió
    - Pruebas unitarias
      * Comprobar las funciones por separado
    - Pruebas de integración
      * Prueba del código completo
    - Pruebas de regresión
      * Pruebas que todo funciones
        + Lo antiguo y lo nuevo
    - Pruebas de aceptación
      * El cliente prueba la aplicación y verifica que es lo que necesita
    - Pruebas estructurales
      * De carga real
  + Documentación
    - Comentar todos los pasos
      * Documentos de especificación de
        + Requisitos
        + Diseño
        + Pruebas
    - Todas deberían tener
      * Manual de usuarios
      * Manual de técnicos
  + Explotación
* Roles y fases del ciclo de vida
  + Roles
    - Analista
    - Programador
    - Diseñador
    - Tester
  + Fases
    - Suele haber todas las fases

Diagrams

* De clases de uso
  + Habla de como funcionara el sistema
* De secuencia
  + Habla de las relaciones entre los objetos
* De actividad
  + Las actividades que se hacen en el programa
* De comunicación
* De tiempo