

# DESARROLLO DEL SOFTWARE

Proceso que ocurre desde que se concibe una idea hasta que un programa está implementado en el ordenador funcionando

## SOFTWARE

Conjunto de programas inf. que actúan sobre el hardware para ejecutar lo que el usuario desee.

DE PROGRAMACIÓN  
conjunto de herramientas para desarrollar programas.  
Ej: compiladores, intérpretes, IDE...

S.O  
Para que las apps puedan ejecutarse

APLICACIONES  
Finalidad concreta.  
procesador txt, hoja cálculo...

PROGRAMA  
Conjunto de instrucciones escritas en un lenguaje de programación. Los lenguajes: están escritos en un idioma fácil para el ser humano. El hardware (0, 1)

## MODELOS CLÁSICOS

CASCADA / V  
Requiere saber de antemano todos los requisitos del sistema, las etapas pasan de una a otra sin retorno posible (se supone que no habrá errores ni variaciones del software)

## PROTOTIPOS

Se define el problema. Prototipo inicial que el cliente evalúa y se amplía hasta obtener el definitivo (útil cuando el cliente interactúa mucho con el sistema).

## MODELOS EVOLUTIVOS

Más modernos. Tienen en cuenta la naturaleza cambiante y evolutiva del software.

## EN ESPIRAL

iterativo + cascada  
Se construye en forma de versiones repetidamente, incrementando la funcionalidad









## ITERATIVO INCREMENTAL

Basado en el modelo cascada, fases se repiten y se refinan, y van mejorando.





# DESARROLLO DEL SOFTWARE

| POR HACER  | EN PROCESO  | HECHO  |
|--|---|--|
| <br> | <br><br> | <br><br> |

## METODOLOGÍAS ÁGILES

Basados en el iterativo-incremental. Los requisitos evolucionan según la necesidad del proyecto. El trabajo es realizado mediante la colaboración de equipos auto-organizados (toma de decisiones a corto plazo)

### VENTAJAS

MAYOR CONTROL Y CAPACIDAD DE PREDICCIÓN

MEJORA CALIDAD DEL PRODUCTO

MAYOR SATISFACCIÓN DEL CLIENTE

USO MÉTRICAS MÁS RELEVANTES

MAYOR MOTIVACIÓN TRABAJADORES

TRABAJO COLAVORATIVO

REDUCCIÓN COSTES

## KANBAN

"Sistema de tarjetas"

Diseñado: Toyota

Controla por demanda la fabricación de los productos necesarios en la cantidad y tiempo necesarios. Enfocado a entregar el máximo valor para los clientes, utilizando los recursos justos.

## SCRUM

Modelo de desarrollo incremental. Iteraciones (sprint) regulares cada 2 a 4 semanas. Al principio de cada iteración se establecen sus objetivos priorizados (sprint backlog). Al finalizar cada iteración se obtiene una entrega parcial utilizable por el cliente. Existen reuniones diarias para tratar la marcha del sprint

## XP (PROGRAMACIÓN EXTREMA)

El objetivo es producir un software de mayor calidad para mejorar la eficiencia del equipo de desarrollo. La más exitosa. Características: diseño sencillo, pequeñas mejoras continuas, pruebas y refactorización, integración continua...





# FASES EN EL DESARROLLO Y EJECUCIÓN DEL SOFTWARE

Independientemente del modelo elegido hay que seguir las etapas.

## 1. ANALISIS DE REQUISITOS

Fase de mayor importancia (+ detalle = mejor), buena comunicación (analista - cliente)

### \* FUNCIONALES

Funciones de la app. Situaciones inesperadas, entradas...

### NO FUNCIONALES

Tiempo de respuesta, legislación, tratamiento ante sobrecarga...

### DOCUMENTO ERS

Especificación Requisitos de Software (Culminación fase)

PLANIFICACIÓN REUNIONES

R. FUNC Y NO FUNC.

OBJETIVOS CLIENTE-SISTEMA

OBJETIVOS PRIORITARIOS Y DE TEMPORIZACIÓN

## 2. DISEÑO

Modelo Funcional-Estructural: como hacerlo, dividir el sistema en partes y establecer sus relaciones

ENTIDADES Y RELACIONES BBDD

LENGUAJE

SISTEMA GESTOR BBDD

## 3. CODIFICACIÓN

Proceso de programación. Codificar toda la info. anterior y llevarlo a código fuente (Programador). Cumplir todos los datos impuestos en el análisis de diseño

MODULARIDAD

CORRECCIÓN

FÁCIL LEER

EFICIENCIA

PORTABILIDAD

## FASES

### 1 - C. FUENTE

instrucciones necesarias en algún editor de texto escrito por programadores

### 2 - C. OBJETO

c. binario resultado de compilar el código fuente (c. intermedio)

Programas interpretados no producen código

### 3 - C. EJECUTABLE

c. binario resultante de enlazar los archivos de c. objeto con ciertas rutinas y bibliotecas necesarias. El S.O. es el encargado de cargar el código en memoria RAM y proceder a ejecutarlo







# FASES EN EL DESARROLLO Y EJECUCIÓN DEL SOFTWARE



Servicio de máquina virtual que sirve como base software para la ejecución de programas.



## 3. CODIFICACIÓN

Eclipse  
IntelliJ

## ENTORNOS DE DESARROLLO

IDE  
(ENTORNO DE  
DESARROLLO  
INTEGRADO)

Es el software en el cual escribes tu código y que te ayudará revisando la sintaxis, auto-completando, corriendo tu código, y lo más importante hacer debug de tu aplicación de una manera sencilla.

FRAMEWORK

.NET y Spring

Plataforma software donde están definidos programas de soporte, librerías que te permiten reusarlas y reducir código que tienes que desarrollar. En Java -> Spring. Para testing -> Junit.

VTJAS

Desarr.  
rápido

Reusar  
código

Diseño  
uniforme

Portabilidad

DVTJAS

Dependencia cod.  
respecto al  
framework

Consumo muchos recursos la  
instalación del framework

JAVA ROUTINE  
ENVIROMENT

JVM  
(machine).  
Programa  
interpreta  
cod.

API  
bibliotecas  
que  
implementan  
API de java

## ENTORNOS DE EJECUCIÓN

RUNTIME

Tiempo que tarda  
en ejecutarse un  
programa

## FUNCIONAMIENTO

Consta de una máquina virtual y API's, que son bibliotecas estándar necesarias para ejecutar aplicaciones escritas en un lenguaje de programación. Estos componentes suelen distribuirse juntos para garantizar la compatibilidad. El entorno actúa como intermediario entre el código fuente y el sistema operativo, permitiendo la ejecución de aplicaciones. Sin embargo, para desarrollar nuevas aplicaciones, se requiere un "entorno de desarrollo" adicional.

se compone de utilidades que permitirá la ejecución de programas java sobre cualquier tipo de plataforma





# DESARROLLO DEL SOFTWARE

## 6. EXPLOTACIÓN

Fase en que los usuarios finales conocen la aplicación y comienzan a utilizarla. (equipos del cliente)

### 1 - INSTALACIÓN

### 2 - BETA TEST

### 3 - CONFIGURACIÓN

La pueden hacer ellos con guía. Si es sencilla le damos autoconfiguración.

## 4 - APP USUARIO FINAL Y EXPLOTACIÓN DE SOFTWARE

### PERFECTIVOS

### EVOLUTIVOS

### CORRECTIVOS

### ADAPTATIVOS

## 5. DOCUMENTACIÓN

Fases perfectamente documentadas. Permitirá la reutilización de parte de los programas en otras apps.  
-Guías (Técnica, Uso, Instalación) (a quien va dirigido y cuál es el objetivo)

## 7. MANTENIMIENTO

Fase más larga. El software es cambiante = actualizar según hardware.

Proceso de control, mejora y optimización.

Servicio mantenimiento pactado con el cliente.

Solventar errores

## 4. PRUEBAS

Conjunto seleccionado y predefinido de datos límite a los que la aplicación es sometida.

### UNITARIAS

probar, una a una, las diferentes partes de software y comprobar su funcionamiento

### JUnit

framework de pruebas Java.

### INTEGRACIÓN

Después de unitarias.

comprobar funcionamiento sistema completo: con todas sus partes interrelacionadas.

### Beta Test

prueba final, se realiza en el entorno donde el software va a ser utilizado (cliente, sus equipos...) Periodo de prueba pactado con el cliente

