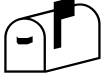


# Metodologías de Desarrollo de Software


1. Introducción.
2. Características principales.
3. Clasificación de las metodologías.
4. Principales metodologías de desarrollo.

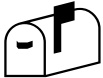
## CONCEPTOS GENERALES

 **Metodología:** Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

 **Tarea:** Actividades elementales en que se dividen los procesos.

 **Procedimiento:** Definición de la forma de ejecutar la tarea.

 **Técnica:** Herramienta utilizada para aplicar un procedimiento.  
Se pueden utilizar una o varias.

 **Herramienta:** Para realizar una técnica, podemos apoyarnos en las herramientas software que automatizan su aplicación.




 **Producto:** Resultado de cada etapa.

## METODOLOGÍA Vs CICLO DE VIDA

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo.

La metodología indica cómo hay que obtener los distintos productos parciales y finales

## GENERACIONES DE METODOLOGÍA

-  Desarrollo Convencional (Sin Metodología) .
-  Desarrollo Estructurado.
-  Desarrollo Orientado a Objetos.

## **DESARROLLO CONVENCIONAL**

- ☹ Los resultados finales son impredecibles
- ☹ No hay forma de controlar lo que está sucediendo en el Proyecto
- ☹ Los cambios organizativos afectan negativamente al proceso de desarrollo

## DESARROLLO ESTRUCTURADO

- ☑ *Programación estructurada*
- ☑ *Diseño estructurado*
- ☑ *Análisis estructurado*
- ☑ *Especificaciones funcionales:*
  - ☒ *Gráficas*
  - ☒ *Particionadas*
  - ☒ *Mínimamente redundantes*

## EJEMPLO DE PROGRAMACIÓN CONVENCIONAL

```
10    CLS
20    A=10
30    INPUT B
40    IF B=A THEN GOTO 50 ELSE GOTO 70
50    PRINT "A Y B SON IGUALES"
60    GOTO 100
70    IF A>B THEN GOTO 80 ELSE GOTO 90
80    B= B + 1; GOTO 40
90    B= B - 1; GOTO 40
100   END
```

## EJEMPLO DE PROGRAMACIÓN ESTRUCTURADA

*PROGRAM NUMEROSIGUALES*

*BEGIN*

*CLEARSCREEN;*

*A := 10 ;*

*INPUT B;*

*REPEAT*

*IF B=A THEN PRINT "A Y B SON IGUALES"*

*ELSE REDUCEDIFERENCIA(A,B);*

*UNTIL B=A;*

*END;*

*PROCEDURE REDUCEDIFENCIA(A,B);*

*BEGIN*

*IF A>B THEN B:= B+1*

*ELSE B:= B - 1*

*END*

# METODOLOGIAS DE DESARROLLO DE SOFTWARE

4.040

## RELACION HISTORICA DE LAS PRINCIPALES METODOLOGIAS

AÑO	METODOLOGÍA
1968	Conceptos sobre la programación estructurada de DIJKSTRA
1974	Técnicas de programación estructurada de WARNIER y JACKSON
1975	Primeros conceptos sobre diseño estructurado de MYERS y YOURDON
1977	Primeros conceptos sobre análisis estructurado GANE y SARSON
1978	Análisis estructurado: DEMARCO y WEINBERG Nace MERISE
1981	SSADM (versión inicial) Information Engineering (versión inicial)
1985	Análisis y Diseño estructurado para sistemas de tiempo real de WARD y MELLOR
1986	SSADM Versión 3
1987	Análisis y Diseño estructurado para sistemas de tiempo real de HATLEY y PIRHBAY
1989	METRICA (versión inicial)
1990	SSADM Versión 4
1993	METRICA Versión 2
1995	METRICA Versión 2.1

## **DESARROLLO ORIENTADO AL OBJETO**

La esencia del desarrollo orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación y no tanto de su representación final en un lenguaje de programación.

## **CONSIDERACIONES SOBRE METODOLOGIAS OO**

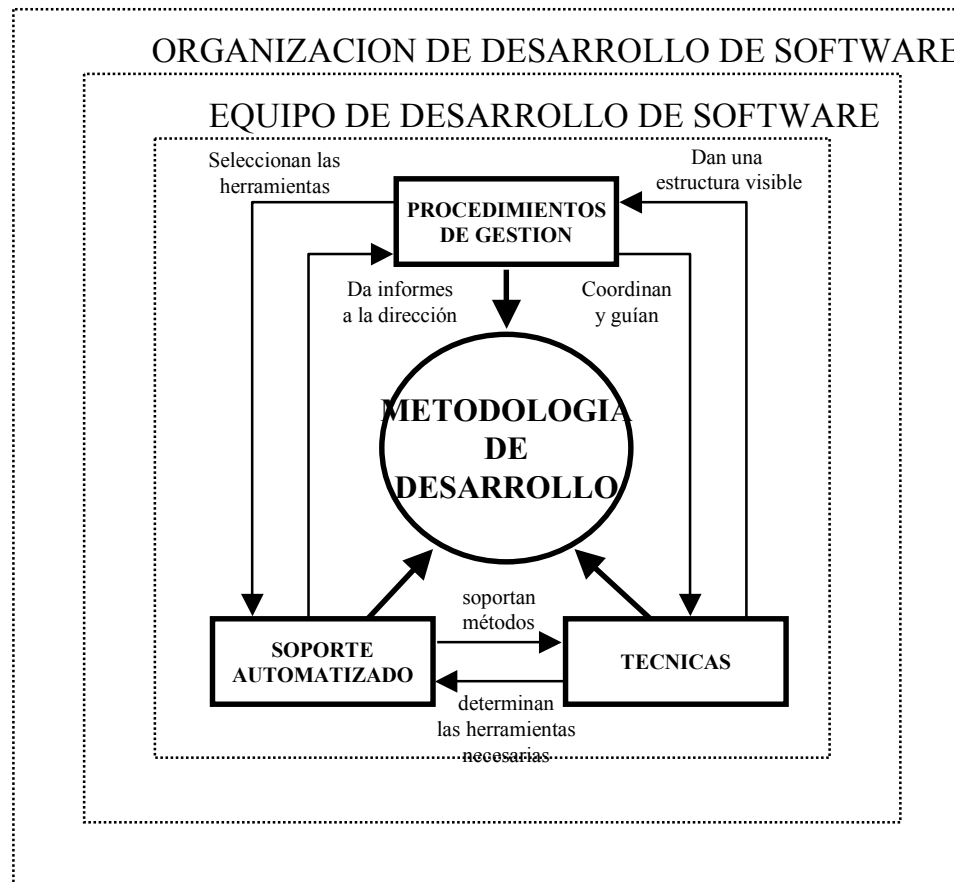
- ❖ Se eliminan fronteras entre fases debido a la naturaleza iterativa del desarrollo orientado al objeto.
- ❖ Aparece una nueva forma de concebir los lenguajes de programación y su uso al incorporarse bibliotecas de clases y otros componentes reutilizables.
- ❖ Hay un alto grado de iteración y solapamiento, lo que lleva a una forma de trabajo muy dinámica.

## **ASPECTOS POSITIVOS DE LAS METODOLOGÍAS OO**

- ❖ Son interactivas e incrementales.
- ❖ Fácil de dividir el sistema en varios subsistemas independientes.
- ❖ Se fomenta la reutilización de componentes.

## IMPACTO DE LA METODOLOGIA EN EL ENTORNO DE DESARROLO

ENTORNO DE DESARROLLO DE SOFTWARE



## **CARACTERISTICAS DESEABLES** **DE UNA METODOLOGIA**

- ☺ Existencia de reglas predefinidas
- ☺ Cobertura total del ciclo de desarrollo
- ☺ Verificaciones intermedias
- ☺ Planificación y control
- ☺ Comunicación efectiva
- ☺ Utilización sobre un abanico amplio de proyectos
- ☺ Fácil formación
- ☺ Herramientas CASE
- ☺ Actividades que mejoren el proceso de desarrollo
- ☺ Soporte al mantenimiento
- ☺ Soporte de la reutilización de software

## CLASIFICACION DE LAS METODOLOGIAS

### ⚡ *Estructuradas*

- ⊗ *Orientadas a Procesos*

- ⊗ *Orientadas a datos*

  - ★ *Jerárquicas*

  - ★ *No Jerárquicas*

- ⊗ *Mixtas*

### ⚡ *Orientadas a Objetos*

### ⚡ *Para Sistemas de Tiempo Real*

## **METODOLOGIAS ESTRUCTURADAS**

### *METODOLOGIAS ORIENTADAS A PROCESOS*

- ***Especificación estructurada:***

- ▣ *Diagramas de Flujo de Datos*

- ▣ *Diccionario de Datos*

- ▣ *Especificaciones de procesos*

## METODOLOGIAS ESTRUCTURADAS

### *METODOLOGIAS ORIENTADAS A PROCESOS*

#### **FASES DEL ANALISIS ESTRUCTURADO**

<b>Método de DeMarco</b>	<b>Método de Gane y Sarson</b>
<ol style="list-style-type: none"><li>1. Construir el modelo físico actual (DFD físico actual)</li><li>2. Construir el modelo lógico actual (DFD lógico actual)</li><li>3. Crear un conjunto de modelos físicos alternativos</li><li>4. Estimar los costes y tiempos de cada opción</li><li>5. Seleccionar un modelo</li><li>6. Empaquetar la especificación</li></ol>	<ol style="list-style-type: none"><li>1. Construir el modelo lógico actual (DFD lógico actual)</li><li>2. Construir el modelo del nuevo sistema: elaborar una especificación estructurada y construir un modelo lógico de datos en tercera forma normal que exprese el contenido de los almacenes de datos.</li><li>3. Seleccionar un modelo lógico</li><li>4. Crear el nuevo modelo físico del sistema</li><li>5. Empaquetar la especificación</li></ol>

## **METODOLOGIAS ESTRUCTURADAS**

### *METODOLOGIAS ORIENTADAS A PROCESOS*

- *Metodología de Yourdon/Constantine*

- ▣ *Realizar los DFD del sistema*

- ▣ *Realizar el diagrama de estructuras*

- ▣ *Evaluar el diseño*

- ▣ *Preparar el diseño para la implantación*

## METODOLOGIAS ESTRUCTURADAS

### *METODOLOGIAS ORIENTADAS A DATOS JERARQUICOS*

- *La estructura de control del programa debe ser jerárquica y se debe derivar de la estructura de datos del programa*
- *El proceso de diseño consiste en definir primero las estructuras de los datos de entrada y salida, mezclarlas todas en una estructura jerárquica de programa y después ordenar detalladamente la lógica procedimental para que se ajuste a esta estructura*
- *El diseño lógico debe preceder y estar separado del diseño físico*

## METODOLOGIAS ESTRUCTURADAS

### *METODOLOGIAS ORIENTADAS A DATOS NO JERARQUICOS*

- ***Metodología Ingeniería de la Información***

- ☐ ***Planificación:*** *construir una arquitectura de la Información y una estrategia que soporte los objetivos de la organización*
- ☐ ***Análisis:*** *comprender las áreas del negocio y determinar los requisitos del sistema*
- ☐ ***Diseño:*** *establecer el comportamiento del sistema deseado por el usuario y que sea alcanzable por la tecnología*
- ☐ ***Construcción:*** *construir sistemas que cumplan los tres niveles anteriores*

## METODOLOGIAS ORIENTADAS A OBJETOS

☐ *“Revolucionarios” o “puros”*

☐ *“Sintetistas” o “evolutivos”*

## **METODOLOGIAS PARA SISTEMAS DE TIEMPO REAL**

- ☐ *Manejo de interrupciones*
- ☐ *Comunicación y sincronización entre tareas*
- ☐ *Gestión de procesos concurrentes*
- ☐ *Respuesta oportuna ante eventos externos*
- ☐ *Datos continuos o discretos*

*Se está produciendo una evolución de las metodologías orientadas a objetos para desarrollos de sistemas de tiempo real*

## **PRINCIPALES METODOLOGIAS DE DESARROLLO**

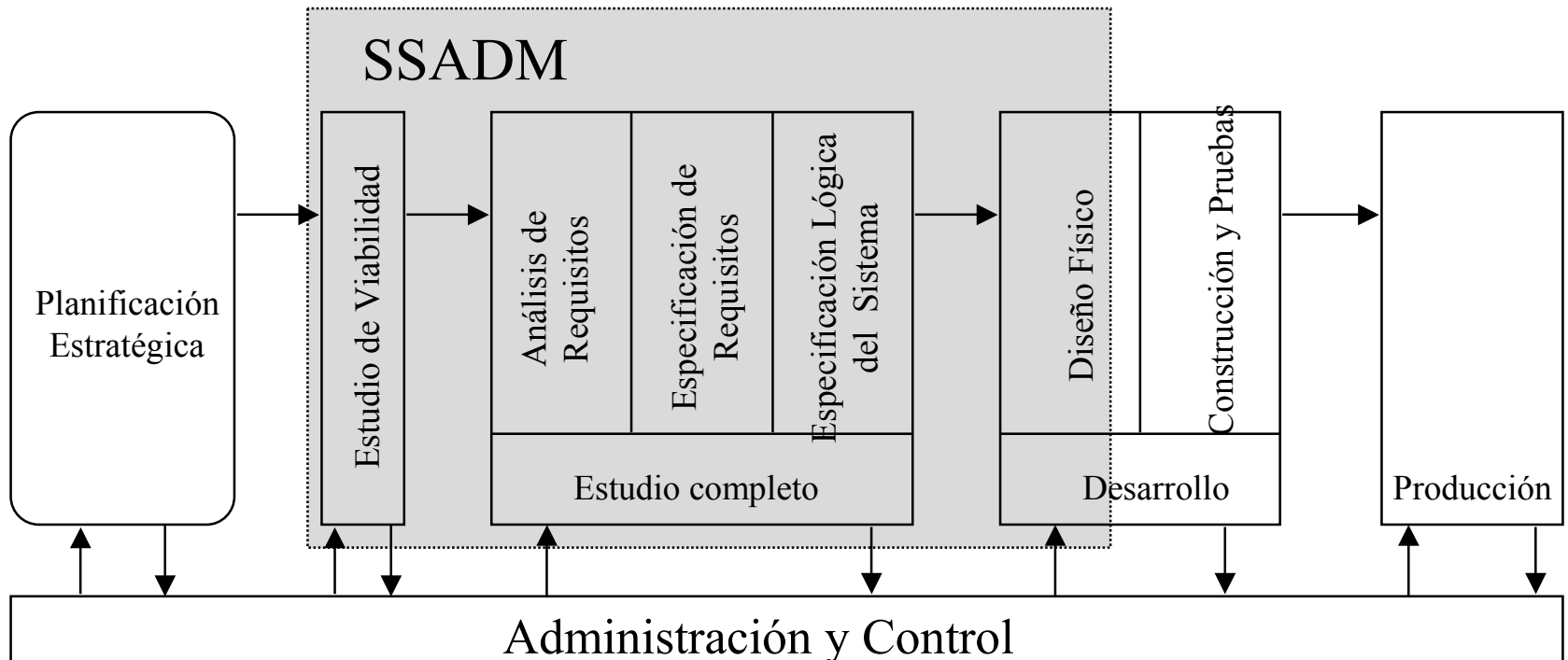
### *METODOLOGIA MERISE*

#### *Fases de la Metodología:*

- *Estudio Preliminar*
- *Estudio Detallado*
- *Implementación*
- *Realización y puesta en marcha*

## PRINCIPALES METODOLOGIAS DE DESARROLLO

### *METODOLOGIA SSADM*



## **PRINCIPALES METODOLOGIAS DE DESARROLLO**

### *METODOLOGIA METRICA*

- ☑ FASE 0: Plan de Sistemas de Información
- ☑ FASE 1: Análisis de Sistemas
- ☑ FASE 2: Diseño de Sistemas
- ☑ FASE 3: Construcción de Sistemas
- ☑ FASE 4: Implantación de Sistemas

## Ejercicio 1

Señalar las respuestas correctas.

El análisis estructurado se diferencia del clásico en:

- a) Emplear un método de partición efectivo.
- b) Construir un modelo lógico del sistema.
- c) Definir los procesos.
- d) Definir las líneas del diseño.

En el análisis estructurado:

- a) El texto se introduce en todos los detalles inmediatamente.
- b) Se va de lo abstracto al detalle, es gráfico y unidimensional.
- c) Se usa un método para particionar exclusivamente problemas complejos.
- d) Ninguna de las anteriores.

## **Ejercicio 2**

Proponer y discutir las principales actividades y productos resultantes de una metodología de análisis estructurado.

## **Ejercicio 3**

Indíquese qué metodología utilizaría para realizar una aplicación que controle el funcionamiento de un conjunto de ascensores de un gran edificio.

## **Ejercicio 4**

Indíquese qué metodología utilizaría en el caso de realizar una aplicación que consista en la gestión y mantenimiento de una gran base de datos.

## **Ejercicio 5**

Intenta realizar una clasificación de los tipos de metodología de desarrollo del Software existentes indicando brevemente sus principales características.