

ESQUEMA DE CICLOS DE VIDA , ANALISIS Y DISEÑO DE SOFTWARE

1 . CICLOS DE VIDA

1 . 1 . Ciclo de Vida en Cascada

- **Ventajas**
 1. Planificación sencilla
 2. Calidad alta
 3. Personal poco cualificado
- **Desventajas**
 1. Requisitos al principio
 2. Dificil rectificar errores
 3. El producto al final
 4. No se sabe el progreso
 5. Lento y costoso
- **Adecuado para**
 1. Se tienen requisitos al principio
 2. Proyectos novedosos
 3. Proyectos que se entienden al principio
- **Subtipos**
 1. **En V** : Se valida y verifica
 2. **Sashimi**
 - Solapamiento de fases
 - Poca documentación
 - Dificil control del proceso y comunicación
 3. **Con subproyectos** : A partir del diseño se subdivide y se integra al final
 4. **Incremental**
 - Se van añadiendo funcionalidades
 - No son necesarios todos los requisitos al principio
 - Análisis común y diseño e implementación se añaden funcionalidades y se crean prototipos
 5. **Reducción de riesgos** : Análisis y diseño , luego prototipo supervisado por el cliente y vuelta atrás

1 . 2 . Ciclo de Vida en Espiral

- **Ciclos que se repiten**
- **Todos los ciclos tienen las mismas fases**
- **Ventajas**
 1. No son necesarios todos los requisitos inicialmente
 2. Se sacan proyectos rápidamente

3. Menor riesgo al perder sólo una fase

- **Inconvenientes**

1. Dificil evaluar riesgos
2. Se necesita continuamente al cliente

- **Adecuado para**

1. Sistemas de gran tamaño
2. Cuando no se conocen al principio todos los requisitos

1 . 3 . Ciclo de vida Orientado a Objetos

- **Se divide el trabajo en miniproyectos**
- **Ciclos de vida iterativo e incremental**
- **Fases**

1. Planificación del negocio
2. Construcción
3. Entrega

- **Ventajas**

1. Solapamiento
2. Iteratividad

2 . FASE DE ANALISIS (REQUISITOS)

2 . 1 . Fases de Requisitos

- **Obtención**
- **Análisis**
- **Especificación**
- **Validación**
- **Documentación**

2 . 2 . Métodos de obtención de requisitos

- **Entrevistas**
- **Desarrollo conjunto de aplicaciones (JAD)**

1. Basado en dinámica de grupos
2. Reuniones de 2 a 4 días
3. Uso de técnicas multimedia
4. Trabajo sistemático
5. **Ventajas**

- Contraste de información obtenida in situ
- Clientes involucrados

6. **Inconvenientes**

- Dificil reunir a todos

- Técnica difícil

7. Participantes

- Jefe
- Analista
- Patrocinador
- Representantes de usuarios
- Desarrolladores
- Especialistas

8. Fases

- Adaptación
- Sesión
- Organización de la documentación

• Planificación conjunta de requisitos (JAR)

- 1. Subconjunto de JAD**
- 2. Para alta dirección**
- 3. Requisitos de alto nivel o estratégicos**
- 4. Pasos**

- Iniciación
- Búsqueda
- Lugar
- Selección de participantes
- Redactar agenda
- Realización

• Brainstorming

- 1. Tormenta de ideas**
- 2. Entrevista de grupo**
- 3. No estructurada**
- 4. Adecuado al principio del proyecto**
- 5. Muchos puntos de vista**
- 6. Ventajas**

- Fácil
- Poco formalizado

7. Inconvenientes

- Resultados con poco detalle
- Difícil reunir a todo el mundo

8. Fases

- Preparación
- Desarrollo
- Consolidación
- Documentación

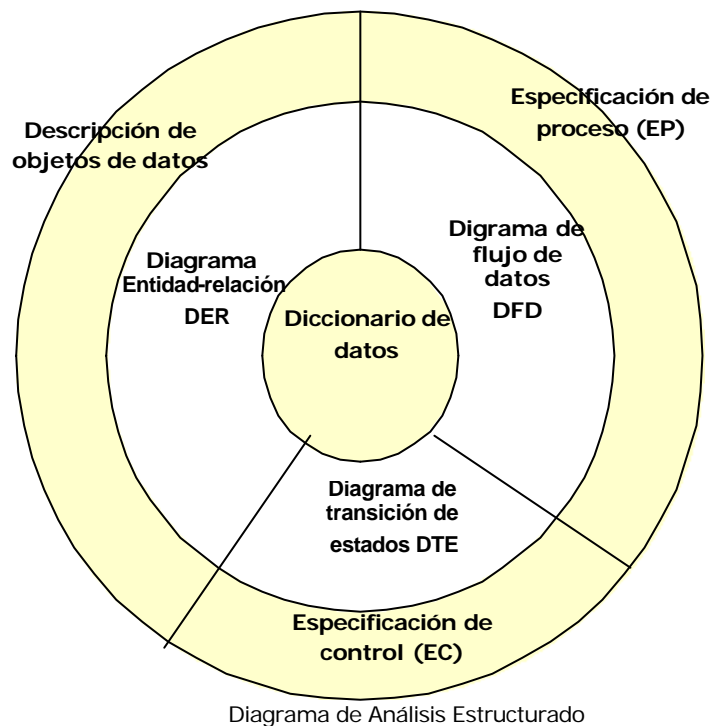
• Prototipos

1. Versión reducida de la aplicación final
2. Prototipos de desarrollo rápido o de desarrollo incremental
3. No utilizable como sistema final
4. Técnicas

- Lenguajes de alto nivel
- Programación de bases de datos
- Componentes reutilizables

- Casos de uso

1. Muestran la interacción entre el sistema y el exterior
2. Lenguaje natural
3. Requisitos funcionales desde el punto de vista del usuario
4. Lo inicia un actor



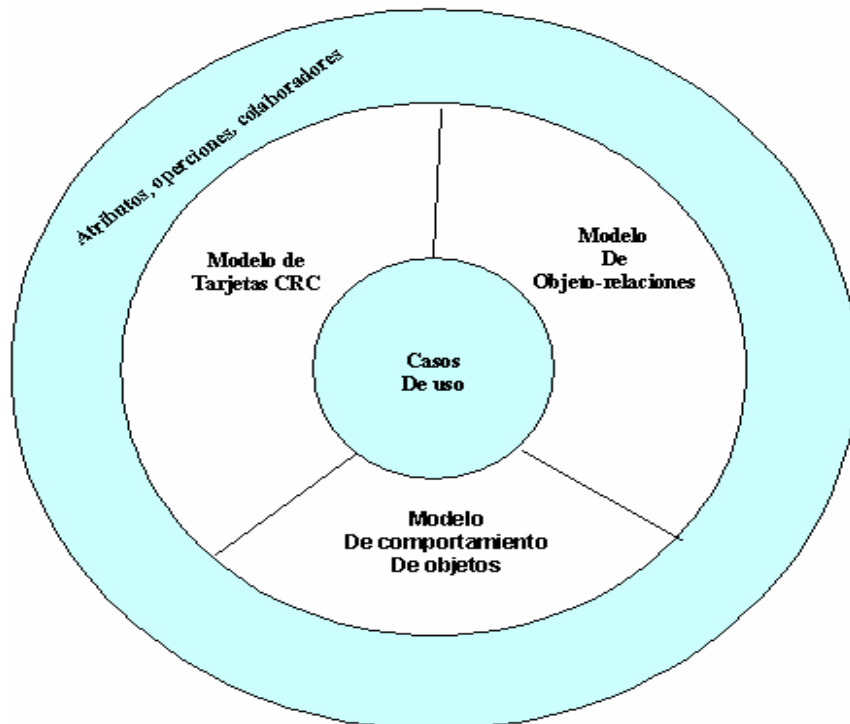
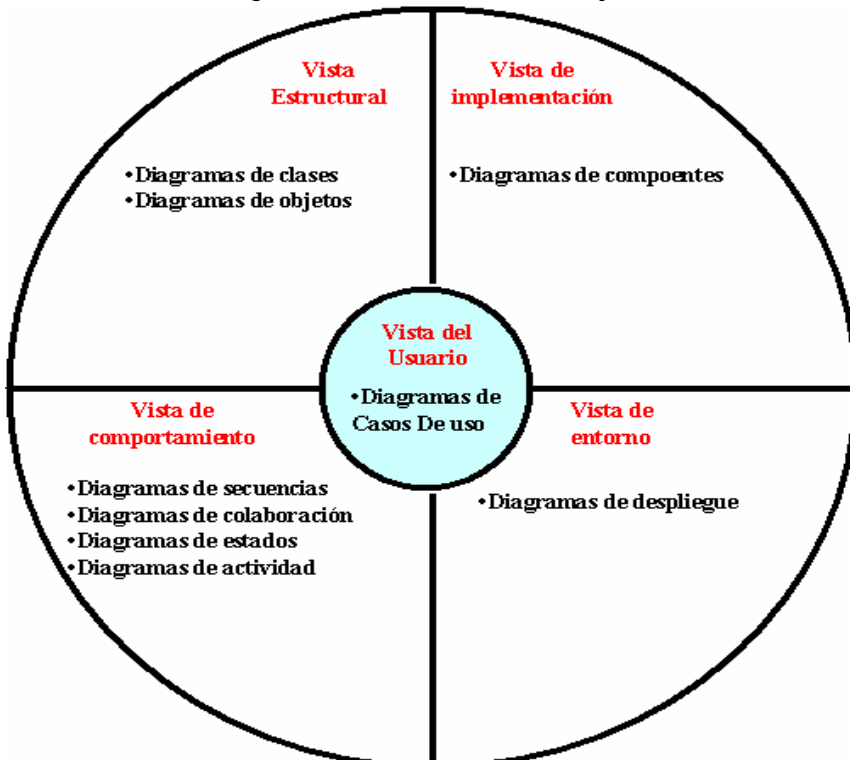


Diagrama de Análisis Orientado a Objetos



Vistas de Análisis Orientado a Objetos

3 . DISEÑO

3 . 1 . Conceptos y elementos del diseño

- **Finalidades :**
 1. Diseño de datos
 2. Diseño de arquitectura
 3. Diseño de la interfaz
 4. Diseño a nivel de componentes
- **Conceptos :**
 1. **Abstracción**
 2. **Modularidad**
 3. **Ocultación**
 4. **Refinamiento**
- **Diseño modular efectivo :**
 1. **Independencia funcional** = Modularidad+Abstracción+Ocultamiento
 2. **Cohesión (Debe ser ALTA) :**
 - Coincidental (Tareas relacionadas)
 - Lógica
 - Temporal
 - Procedimental (Tareas en secuencia)
 - De Comunicación (Centradas en un área)
 3. **Acoplamiento (Debe ser BAJO) :**
 - De Datos (Transferencia de datos)
 - De Marca (Transferencia de referencias)
 - De Control (Control entre módulos)
 - Externo (Almacén externo de datos)
 - Común (Almacén común a varios módulos)
 - De Contenido (Solapamiento de código entre módulos)

3 . 2 . Diseño arquitectónico

1. **Común al diseño estructurado y al Orientado a Objetos**
2. **Fases :**
 - Descomposición estructural (descomponer en subsistemas)
 - Intercambio de información entre subsistemas (mediante mensajes o con un almacén de datos común)
 - Control (con eventos o centralizado)
3. **Flujos de información**
 - o De transformación (los datos de entrada se transforman)
 - o De transacción (los datos de entrada provocan acciones)

3 . 3 . Diseño estructurado

- **Refinamientos sucesivos**

- **Fase previa de diseño arquitectónico**
- **Utilización de la secuencia , la iteración y el condicional**
- **Utilización del pseudocódigo**

3 . 4 . Diseño orientado a objetos

- **Principios :**
 1. Abstracción
 2. Modularidad
 3. Ocultación
 4. Independencia funcional
- **Capas :**
 1. Diseño de responsabilidades (Estructuras de datos y algoritmos)
 2. Diseño de mensajes (Mensajes a métodos propios o de otras clases)
 3. Diseño de Clases y Objetos (Jerarquía de clases y objetos)
 4. Diseño de subsistemas (Arquitectura del sistema)
- **Utilización de UML**
- **Fases :**
 1. Diseño del sistema (Diseño arquitectónico)
 2. Diseño de objetos (Diseño detallado)
- **Paso de la fase de Análisis a la de Diseño :**
 1. Casos de Uso + Modelo de comportamiento de objetos = Diseño de subsistemas
 2. Atributos + Operaciones + Colaboraciones + Modelo de tarjetas CRC = Diseño de clases y objetos
 3. Modelo de objetos-relaciones = Diseño de mensajes
 4. Atributos + Operaciones + Colaboraciones + Modelo de tarjetas CRC = Diseño de responsabilidades