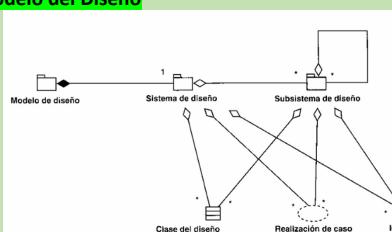
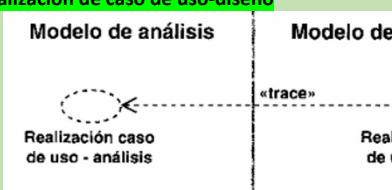
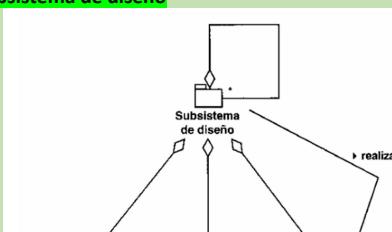
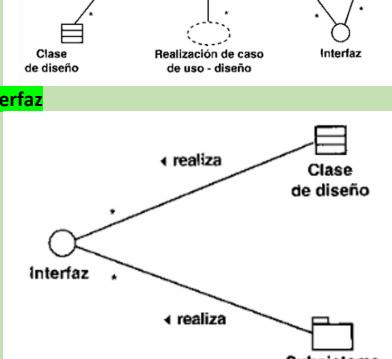
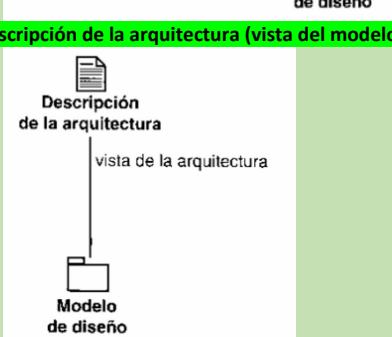
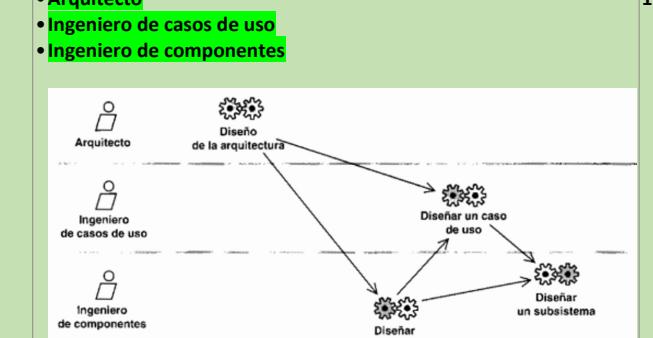
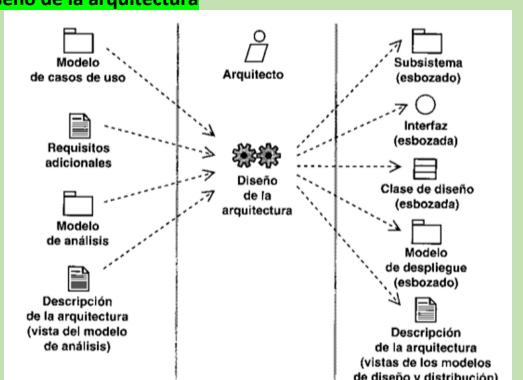
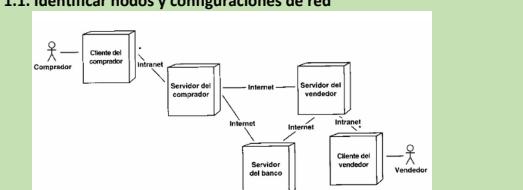
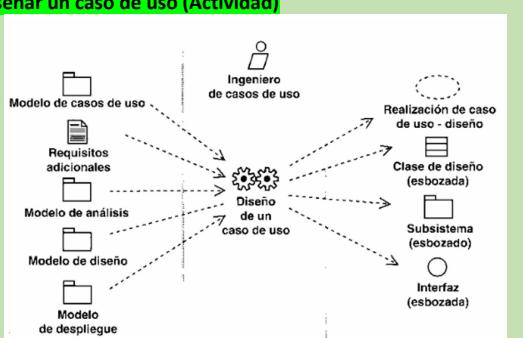
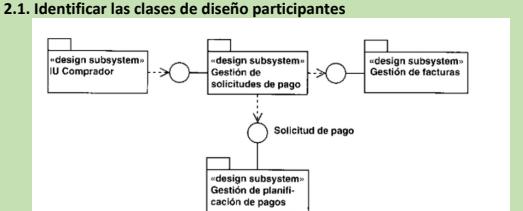
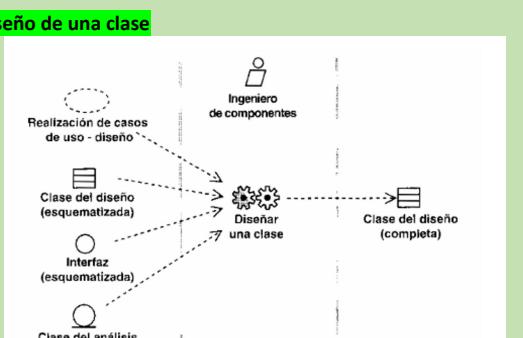
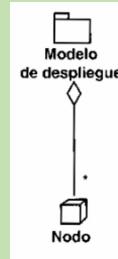


Disciplina	Propósitos/Características	Artefactos	Trabajadores	Flujo de Trabajo (actividades)	Visión general / Comparativa
Captura de Requisitos	<ul style="list-style-type: none"> • Enumerar los requisitos candidatos (Lista de características) • comprender el contexto del sistema (Modelo del Dominio/Negocio) • capturar requisitos funcionales (Modelo de Casos de Uso) • Capturar requisitos no funcionales (Lista especificaciones adicionales) <p>Pasos:</p> <ol style="list-style-type: none"> 1. Lista de Características 2. Entender el negocio: Modelo del Dominio 3. Modelado del Negocio 4. Modelo del Negocio 5. Búsqueda de Casos de Uso a partir de un modelo del negocio 6. Captura de Requisitos como casos de Uso <p>6. Captura de Requisitos como casos de Uso</p>	<ul style="list-style-type: none"> • Modelo de Casos de uso • Actor • Caso de Uso • Descripción de la arquitectura (vista del modelo de CU) • Glosario • Prototipo de interfaz de usuario 	<ul style="list-style-type: none"> • Analista de Sistema • Especificador de casos de Uso • Diseñador de interfaz de Usuario • Arquitecto 	<ul style="list-style-type: none"> • Buscar Actores y casos de Uso • Priorizar casos de Uso • Detallar un caso de Uso • Prototipar la interfaz de usuario • Estructurar el modelo de casos de uso 	<p>La totalidad de los casos de uso constituyen el Modelo de Casos de Uso, trabajadores y artefactos</p>

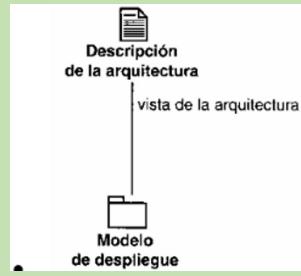
Disciplina	Propósitos/Características	Artefactos	Trabajadores	Flujo de Trabajo (actividades)	Visión general / Comparativa
Análisis	<ul style="list-style-type: none"> Se analizan los requisitos refinandolos y estructurandolos Se utiliza un lenguaje basado en modelos de objetos conceptual (modelo de análisis) Características del Modelo de Análisis <ul style="list-style-type: none"> Especificación más precisa Lenguaje de los desarrolladores Facilita su comprensión, preparación, modificación, mantenimiento Primera aproximación al modelo de diseño 	<p>• Modelo del análisis</p> <p>• Clase del análisis (encajan en alguno de los estereotipos básicos)</p> <p>• Realización caso de uso</p> <p>• Paquete del análisis</p> <p>• Descripción de la arquitectura</p>	<p>• Arquitecto • Ingeniero de Casos de uso • Ingeniero de Componentes</p>	<p>1. Análisis de la arquitectura</p> <p>1.1. Identificación del paquete del análisis</p> <p>1.2. Clases evidentes del análisis</p> <p>1.3. Requisitos especiales comunes</p> <p>2. Analizar un caso de uso</p> <p>3. Analizar una clase</p> <p>4. Analizar un paquete</p>	<p>Modelo de Casos de Uso</p> <p>Describo en el lenguaje del cliente</p> <p>Vista externa del sistema</p> <p>Estructurado por casos de uso, proporciona la estructura a la vista externa</p> <p>Utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre que debería y que no debería hacer el sistema</p> <p>Puede contener redundancias e inconsistencias entre requisitos</p> <p>Captura la funcionalidad del sistema, incluida la funcionalidad significativa para la arquitectura</p> <p>Define casos de uso que se analizaron con más profundidad en el modelo de análisis.</p> <p>Modelo de Análisis</p> <p>Describo en el lenguaje del desarrollador</p> <p>Vista interna del sistema</p> <p>Estructurado por clases y paquetes estereotipados; proporciona la estructura a la vista interna</p> <p>Utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado</p> <p>No debería contener redundancias ni inconsistencias entre requisitos</p> <p>Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño</p> <p>Define realizaciones de caso de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.</p>

Disciplina	Propósitos/Características	Artefactos	Trabajadores	Flujo de Trabajo (actividades)	Visión general / Comparativa																				
Diseño	<ul style="list-style-type: none"> • Lograr una comprensión profunda de los problemas relacionados con los requisitos no funcionales • Crear un punto de partida para las actividades de implementación • Descomponer el trabajo de implementación en piezas más manejables • Capturar las principales interfaces entre subsistemas • Razonamiento sobre la arquitectura • Uso de interfaces para sincronizar diferentes equipos • Visualizar y razonar sobre el diseño mediante el uso de una notación común • Crear una abstracción sin costuras de la implementación del sistema <ul style="list-style-type: none"> • La implementación es un refinamiento directo del diseño al rellenar la "carne" pero no cambia la estructura • Permite el uso de técnicas como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación 	<ul style="list-style-type: none"> • Modelo del Diseño  • Clase del diseño  • Realización de caso de uso-diseño  • Subsistema de diseño  • Interfaz  • Descripción de la arquitectura (vista del modelo de diseño)  	<ul style="list-style-type: none"> • Arquitecto • Ingeniero de casos de uso • Ingeniero de componentes 	<h3>1. Diseño de la arquitectura</h3>  <p>Figura 9.17. Las entradas y los resultados del diseño de la arquitectura.</p> <h4>1.1. Identificar nodos y configuraciones de red</h4>  <h4>1.2. Identificar subsistemas y sus interfaces</h4> <h4>1.3. Identificar clases de diseño arquitecturalmente significativas</h4> <h4>1.4. Identificar mecanismos de diseño genericos</h4> <h3>2. Diseñar un caso de uso (Actividad)</h3>  <h4>2.1. Identificar las clases de diseño participantes</h4>  <h4>2.2. Identificar las interacciones de objeto de diseño</h4> <h4>2.3. Identificar los subsistemas e interfaces participantes</h4> <h3>3. Diseño de una clase</h3> 	<table border="1"> <thead> <tr> <th>Modelo de Análisis</th> <th>Modelo de Diseño</th> </tr> </thead> <tbody> <tr> <td>Modelo conceptual.</td> <td>Modelo físico (implementación)</td> </tr> <tr> <td>Genérico (aplicable a varios diseños)</td> <td>Específico para una implementación</td> </tr> <tr> <td>Tres estereotipos: entidad, control, frontera</td> <td>Cualquier nro. de estereotipos físicos</td> </tr> <tr> <td>Menos formal.</td> <td>Más formal.</td> </tr> <tr> <td>Menos caro de desarrollar</td> <td>Más caro.</td> </tr> <tr> <td>Menos capas.</td> <td>Más capas</td> </tr> <tr> <td>Dinámico (muy centrado en la secuencia)</td> <td>Dinámico (no muy centrado en la secuencia)</td> </tr> <tr> <td>Creado principalmente como trabajo manual</td> <td>Uso de "programación visual" en ing. de ida y vuelta.</td> </tr> <tr> <td>Puede no mantenerse todo el ciclo de vida.</td> <td>Debe ser mantenido todo el ciclo de vida.</td> </tr> </tbody> </table>	Modelo de Análisis	Modelo de Diseño	Modelo conceptual.	Modelo físico (implementación)	Genérico (aplicable a varios diseños)	Específico para una implementación	Tres estereotipos: entidad, control, frontera	Cualquier nro. de estereotipos físicos	Menos formal.	Más formal.	Menos caro de desarrollar	Más caro.	Menos capas.	Más capas	Dinámico (muy centrado en la secuencia)	Dinámico (no muy centrado en la secuencia)	Creado principalmente como trabajo manual	Uso de "programación visual" en ing. de ida y vuelta.	Puede no mantenerse todo el ciclo de vida.	Debe ser mantenido todo el ciclo de vida.
Modelo de Análisis	Modelo de Diseño																								
Modelo conceptual.	Modelo físico (implementación)																								
Genérico (aplicable a varios diseños)	Específico para una implementación																								
Tres estereotipos: entidad, control, frontera	Cualquier nro. de estereotipos físicos																								
Menos formal.	Más formal.																								
Menos caro de desarrollar	Más caro.																								
Menos capas.	Más capas																								
Dinámico (muy centrado en la secuencia)	Dinámico (no muy centrado en la secuencia)																								
Creado principalmente como trabajo manual	Uso de "programación visual" en ing. de ida y vuelta.																								
Puede no mantenerse todo el ciclo de vida.	Debe ser mantenido todo el ciclo de vida.																								

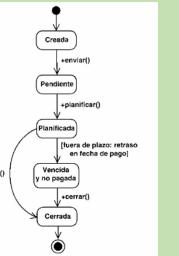
• **Modelo de despliegue**



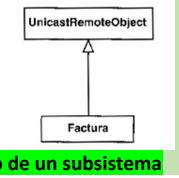
• **Descripción de la arquitectura (vista del modelo de despliegue)**



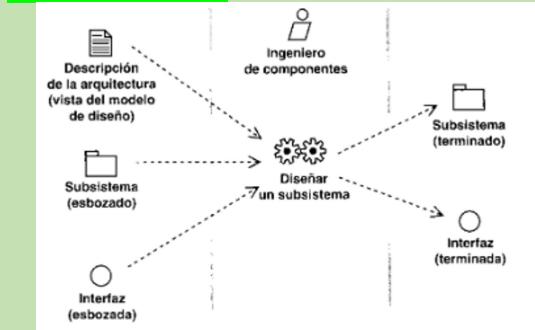
- 3.1. Identificar operaciones
- 3.2. Identificar atributos
- 3.3 Identificar asociaciones y agregaciones
- 3.4 Identificar generalizaciones
- 3.5 Describir métodos
- 3.6 Describir estados



3.7 Manejar requerimientos especiales

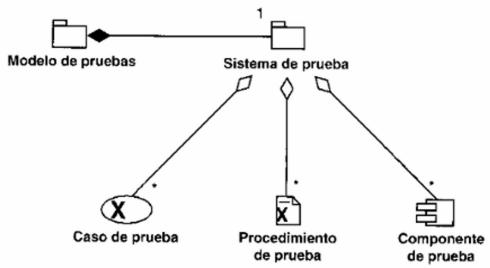
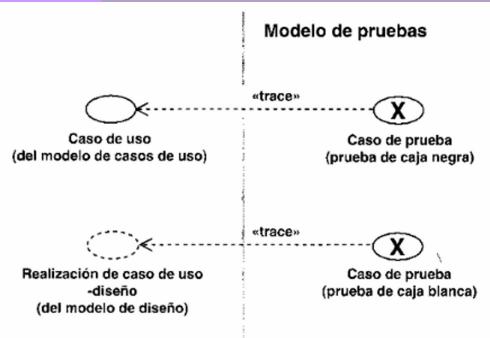
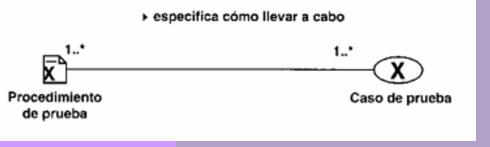
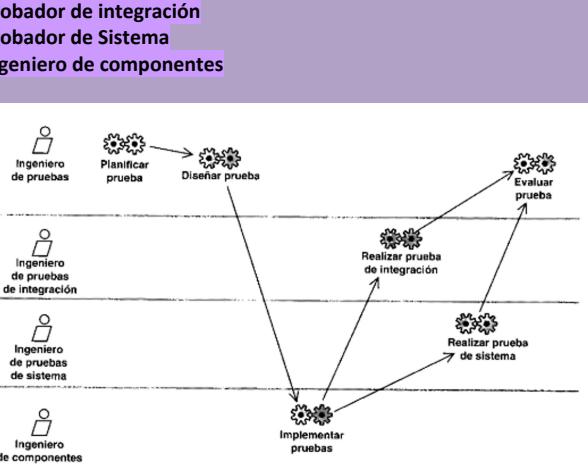
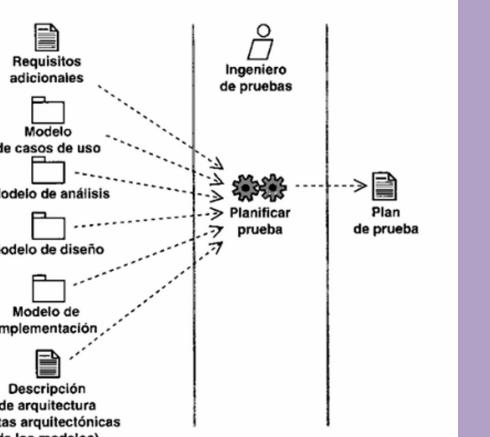
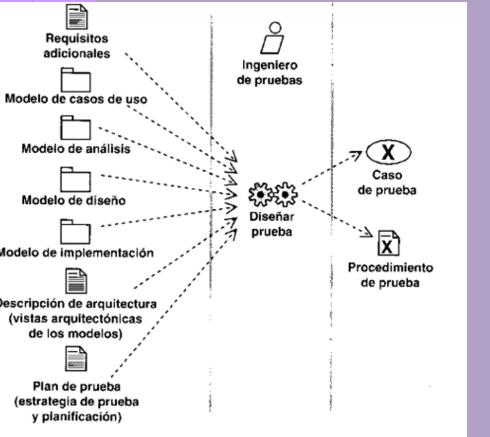
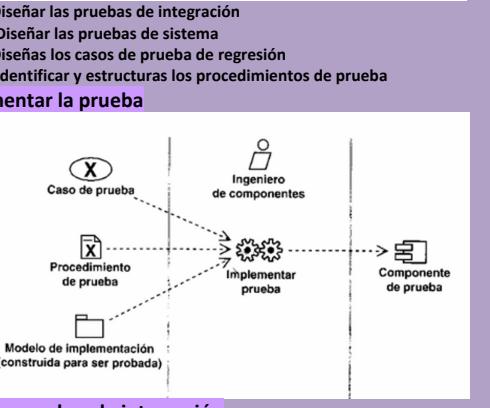
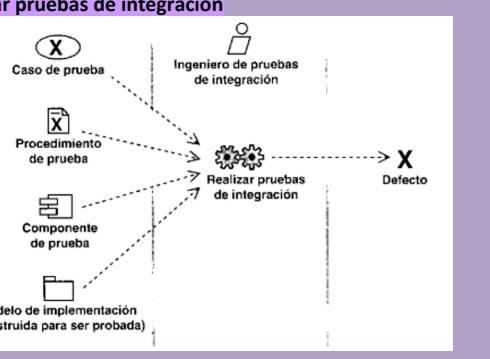
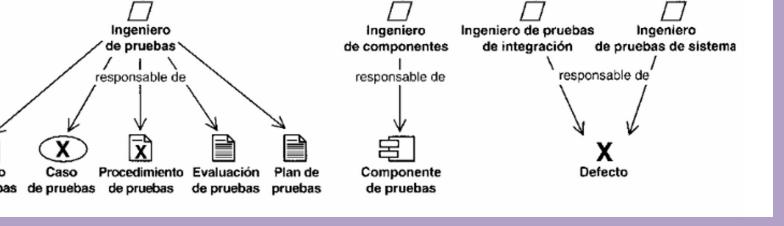


4. Diseño de un subsistema

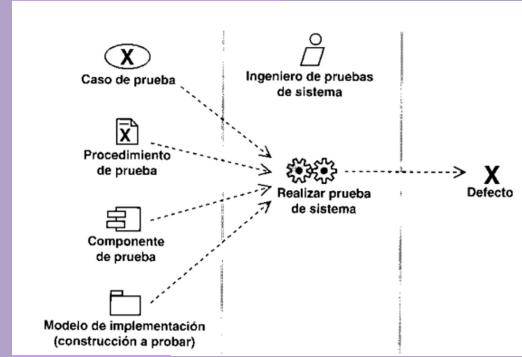


- 4.1 Mantener las dependencias de subsistema
- 4.2 Mantener las interfaces provistas por el subsistema
- 4.3 Mantener el contenido del subsistema

Disciplina	Propósitos/Características	Artefactos	Trabajadores	Flujo de Trabajo (actividades)	Visión general / Comparativa
I Implementación	<ul style="list-style-type: none"> Planear las integraciones requeridas en cada iteración Distribuir el sistema mapeando componentes ejecutables a nodos Implementar las clases y subsistemas encontrados en el diseño Realizar pruebas unitarias e integrar 	<p>• Modelo de Implementación</p> <p>• Componente</p> <p>Figura 10.4. Dependencias de traza entre componentes y clases de diseño.</p> <p>• Subsistema de implementación</p> <p>• Interfaz • Descripción de la arquitectura • Plan de integración de Construcciones</p>	<p>• Arquitecto • Ingeniero de componentes • Integrador de sistemas</p>	<p>1. Implementación de la arquitectura</p> <p>2. Integrar el sistema</p> <p>2.1. Planear la construcción subsecuente 2.2. Integrar la construcción</p> <p>3. Implementar un subsistema</p> <p>3.1 Mantener el contenido de los subsistemas</p> <p>4. Implementar una clase</p> <p>5. Realizar prueba de unidad</p>	<p>Figura 10.1. Los trabajadores y artefactos involucrados en la implementación.</p>

Disciplina	Propósitos/Características	Artefactos	Trabajadores	Flujo de Trabajo (actividades)	Visión general / Comparativa
Prueba	<ul style="list-style-type: none"> • Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema. • Diseñar e implementar pruebas creando: <ul style="list-style-type: none"> • los casos de prueba (especifican qué probar) • procedimientos de prueba (especifican cómo realizar las pruebas) • componentes de prueba para automatizar las pruebas. • Realizar las pruebas. <ul style="list-style-type: none"> • Gestionar resultados • Volver a probar y reenviar a corregir 	<ul style="list-style-type: none"> • Modelo de Pruebas  <p>Figura 11.3. El modelo de pruebas.</p> <ul style="list-style-type: none"> • Caso de Prueba  <ul style="list-style-type: none"> • Procedimiento de prueba  <ul style="list-style-type: none"> • Componente de Prueba  <ul style="list-style-type: none"> • Plan de Prueba • Defecto • Evaluación de prueba 	<ul style="list-style-type: none"> • Ingeniero/Diseñador de pruebas • Probador de integración • Probador de Sistema • Ingeniero de componentes 	<p>1. Planear la prueba</p>  <p>2. Diseñar la prueba</p>  <p>2.1 Diseñar las pruebas de integración 2.2. Diseñar las pruebas de sistema 2.3 Diseñar los casos de prueba de regresión 2.4. Identificar y estructurar los procedimientos de prueba</p> <p>3. Implementar la prueba</p>  <p>4. Efectuar pruebas de integración</p> 	

5. Realizar la prueba de Sistema



6. Evaluar la prueba

