Modelo Dinámico del Diseño del Software y Representación en UML

UNIDAD 9

Análisis y Diseño de Sistemas de Información

El Modelo Dinámico

- El objetivo del modelo Dinámico es presentar o describir el comportamiento del sistema a través del tiempo

Componentes del Modelo

- Los elementos principales del modelo Dinámico son:
- Vista de Interacción
  - Diagramas de Secuencia
  - Diagramas de Colaboración
- Modelo de Máquina de Estados
  - Diagrama de Estados
- Vista de Actividades
  - Diagrama de Actividades

Interacciones

- La Vista de Interacción presenta las interacciones del usuario con el sistema a través del intercambio de mensajes
- Conjunto de mensajes que intercambian entre sí los objetos que componen un sistema
- Estos mensajes se intercambian a través de enlaces

Mensaje

- Un mensaje se define como una comunicación unidireccional entre objetos, adicionalmente puede contener parámetros

Colaboración

- Una colaboración se define como una colección de objetos que interactúan entre ellos para representar un comportamiento en un determinado contexto.

Colaboración

- Una colaboración está formada por ranuras de tiempo que son ocupadas por objetos y ligas entre ellos en tiempo de ejecución
- Cada objeto o liga tienen un rol dentro de una colaboración
- Un objeto puede participar en más de una colaboración

Diagramas de Secuencia

Diagramas de Secuencia

- Se representa con un gráfico en dos dimensiones, el elemento fundamental es una línea vertical que representa el eje del tiempo
- En la dimensión horizontal se presentan los distintos roles o estereotipos presentes en una colaboración
- Cada estereotipo tiene una línea que representa su línea de vida representada por una línea punteada

Mensajes en Diagramas de Secuencia

- Un mensaje es una flecha que parte de la línea de vida de un objeto hacia la línea de vida de otro
- La secuencia se muestra en forma ordenada de manera descendente
- No existe relación entre el tiempo de vida y la distancia entre flechas o líneas de vida
- Una actividad iterativa se representa con un *

Activaciones

- Una activación representa la ejecución de operaciones por parte de un objeto
- Se representa con una línea de doble trazo colocada sobre la línea de vida
- Es posible que haya más de un objeto activo a la vez

Construcción de Diagramas de Secuencia

1. Escribir el nombre y objetivo o post condición del caso de uso a representar en el diagrama
2. Colocar una instancia de las clases entidad que se han identificado y que aparecen en el caso de uso
3. Agregar las interfaces o bordes que forman parte del caso de uso
4. Para identificar los métodos de las clases, se convierten los controles en métodos y mensajes asegurando que el intercambio de mensajes es correcto

Código y Diagramas de Secuencia

- Un Diagrama de Secuencia está un nivel de abstracción más arriba que el código
- No todo lo que es código se dibuja en el diagrama
- Un diagrama puede ser implementado en varios lenguajes
- No es necesario saber programar para realizar un diagrama

Diagramas de Colaboración

Diagramas de Colaboración

- Diagrama que permite representar la interacción entre objetos a través de mensajes que se envían vía enlaces
- Un caso de uso se implementa a través de una colaboración entre clases y otros objetos que se comunican para reflejar la funcionalidad del caso de uso

Elementos del Diagrama

- Objetos (un rectángulo que representa un objeto de una cierta clase)
- Los objetos pueden presentarse como Estereotipos
- Enlaces (representado por una línea que une a dos objetos)
- Flujo de Mensajes (una flecha cerca de un enlace)

Tipos de Mensajes

- Los mensajes pueden ser:
  - Llamadas (un objeto llama un método de otro objeto)
  - Retorno (un receptor regresa un valor si es necesario)
  - Envío (envía una señal a un objeto)
  - Creación y Destrucción

Secuencia en los Diagramas

- El flujo de mensajes forma una secuencia
- La secuencia tiene un número antes y una flecha dirigida
- Caminos alternativos se colocan con el mismo número y la sub secuencia
- Una actividad iterativa se representa con un *
- Una condición se representa con la condición entre [ ]

Construcción de los Diagramas

- Para construir un diagrama de colaboración, se necesita:
  - Casos de Uso
  - Diagramas de Secuencia
  - Diagrama de Clases

La Colaboración Genera

- Los diagramas de colaboración sirven para generar:
  - Diagramas de Estados
  - Diagramas de Componentes
  - Diagramas de Despliegue

Diagramas de Secuencia y Colaboración

- Ambos muestran las mismas relaciones entre objetos, sin embargo bajo diferente enfoque
- Un Diagrama de Secuencia da prioridad a la secuencia de tiempo durante el ciclo de vida de un objeto
- Un Diagrama de Colaboración da prioridad a las relaciones entre objetos sin importar el tiempo

Diagramas de Estado

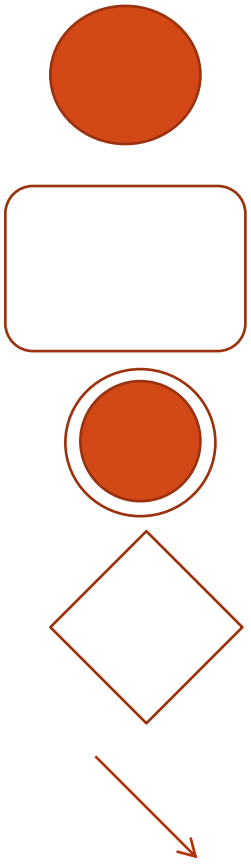
Diagramas de Estados

- Muestran el conjunto de estados por los que pasa un objeto durante su ciclo de vida en la aplicación cuando se presentan diversos eventos

Elementos del Diagrama de Estados

- Estado. Representa la condición de un determinado objeto durante la realización de una actividad
- Evento. Representa un acontecimiento significativo en el tiempo que puede o no generar un cambio de estado
- Transición. Relación entre dos estados que refleja las acciones que ocurrieron para que un objeto pase del Estado A al Estado B

Representación en el Diagrama



- Inicio del Flujo
- Tarea a Realizar
- Fin del Flujo
- Decisión
- Transición

Tipos de Diagramas

- Se pueden tener dos tipos de diagramas de estado en general:
  - Diagrama de Estado Simple
    - No contiene diagramas de estado dentro de otro
  - Diagrama de Estado Compuesto
    - Contiene diagramas de estado dentro de otros
    - Secuenciales (solo un estado activo a la vez)
    - Concurrentes (varios estados activos a la vez)
- Sub máquina. Es un tipo de Diagrama Compuesto que puede ser referenciado por otras máquinas de estado

Vista de Actividad

Diagramas de Actividad

- Es una variante de los Diagramas de Estado que muestra una secuencia de actividades
- Un diagrama de Actividad documenta la lógica de un caso de uso
- Representan el flujo de control entre objetos al momento de ejecutar un caso de uso

Elementos

- Inicio

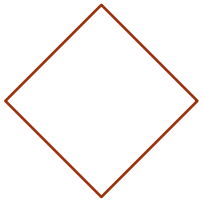
- Actividad

- Transición

- Finalizado

Elementos

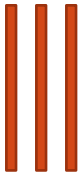
- Vías alternas



- Barras de Sincronización (*Fork* - *Join*)

- Objeto

- Carriles (*Swimlane*)



Consideraciones

- Algunas actividades pueden representarse mediante varios diagramas
- Si existen más de tres caminos alternativos, se recomienda realizar un diagrama por cada uno de ellos

Construcción

- Crear los carriles para los Actores y a los elementos del negocio
- Colocar la marca de inicio en el objeto u Actor que inicia la secuencia
- Comenzar a colocar las transiciones para las actividades de un objeto o entre distintos objetos

Actividad vs Estado

- Ambos modelan el comportamiento dinámico del sistema
- Un diagrama de actividad no muestra los estados que toma un objeto, muestra las actividades que realiza cada uno de ellos

Modelo de Implementación

Modelo de Implementación

- Representa la composición física de la implementación de u sistema
- Un componente físico es algo que se “puede ver” en el disco duro
- Pueden contener:
  - Archivos ejecutables
  - Código fuente
  - Directorios entre otros elementos

Contenido del Diagrama

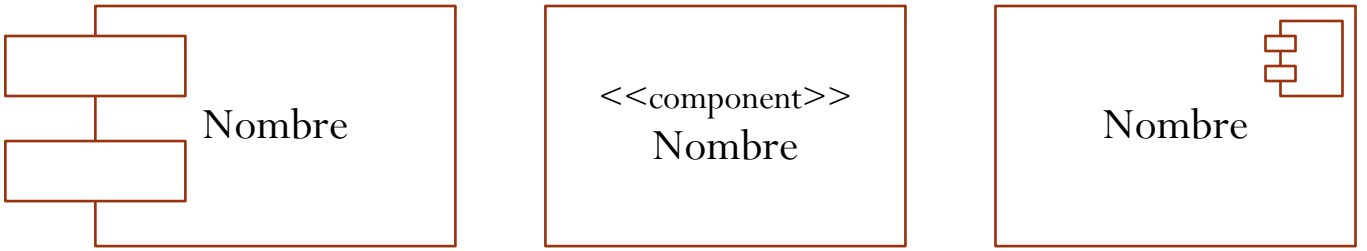
- Dependencias entre las partes que forman el código del sistema a través de un diagrama de componentes
- La estructura del sistema cuando está en ejecución a través de un diagrama de despliegue

Diagrama de Componentes

- Representa la parte física de un sistema
- Ayuda a comprender de mejor manera el camino de la implementación
- Los componentes pueden agruparse en paquetes y puede haber relaciones entre ellos

Representación

- Un componente se representa con un rectángulo que tiene un nombre y dos rectángulos en la parte izquierda

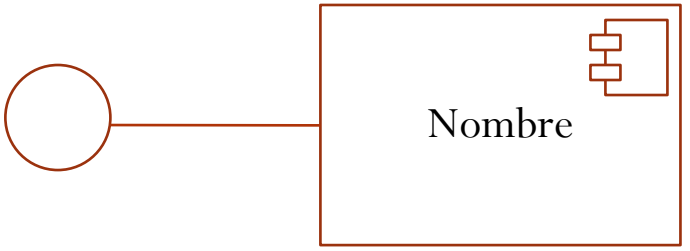


Estereotipos

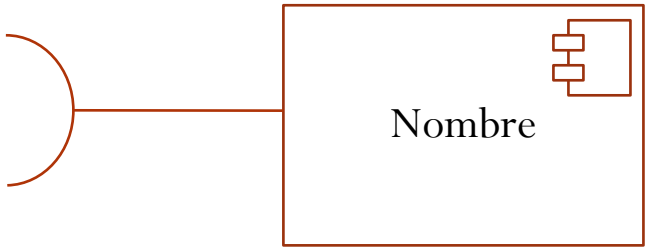
- Se tienen los siguientes estereotipos para los Diagramas de Componentes
  - Ejecutable. Componentes ejecutables
  - Bibliotecas. Estáticas o dinámicas
  - Tabla. Tabla de una base de datos
  - Archivos. Datos o código fuente
  - Documento. Documentos generales

Interfaces

- Una interfaz es la forma de unión entre varios componentes
  - Un componente proporciona una interfaz



  - Un componente usa una interfaz



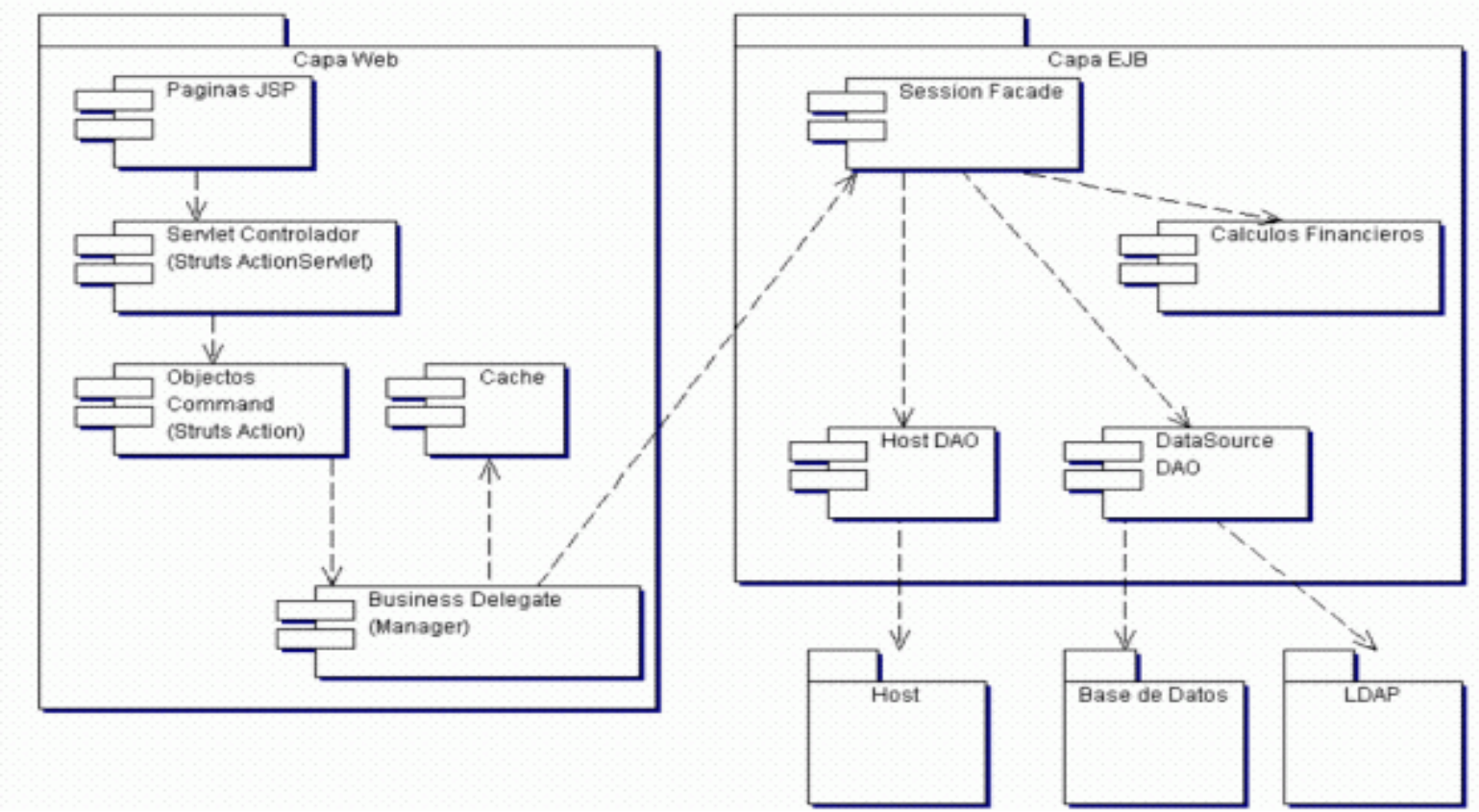
Diseño y Diagrama de Componentes

- Los elementos que surgieron a partir del diseño, pueden agregarse a los diagramas de componentes

Creación del Diagrama

- Es necesario contar con el diagrama de clases, tanto las del modelo del dominio, como las que ya tienen funcionalidades
- Los métodos de estas clases se consideran módulos independientes
- Estos módulos se convierten en componentes

Ejemplo (nivel superior)



Ejemplo (nivel implementación)

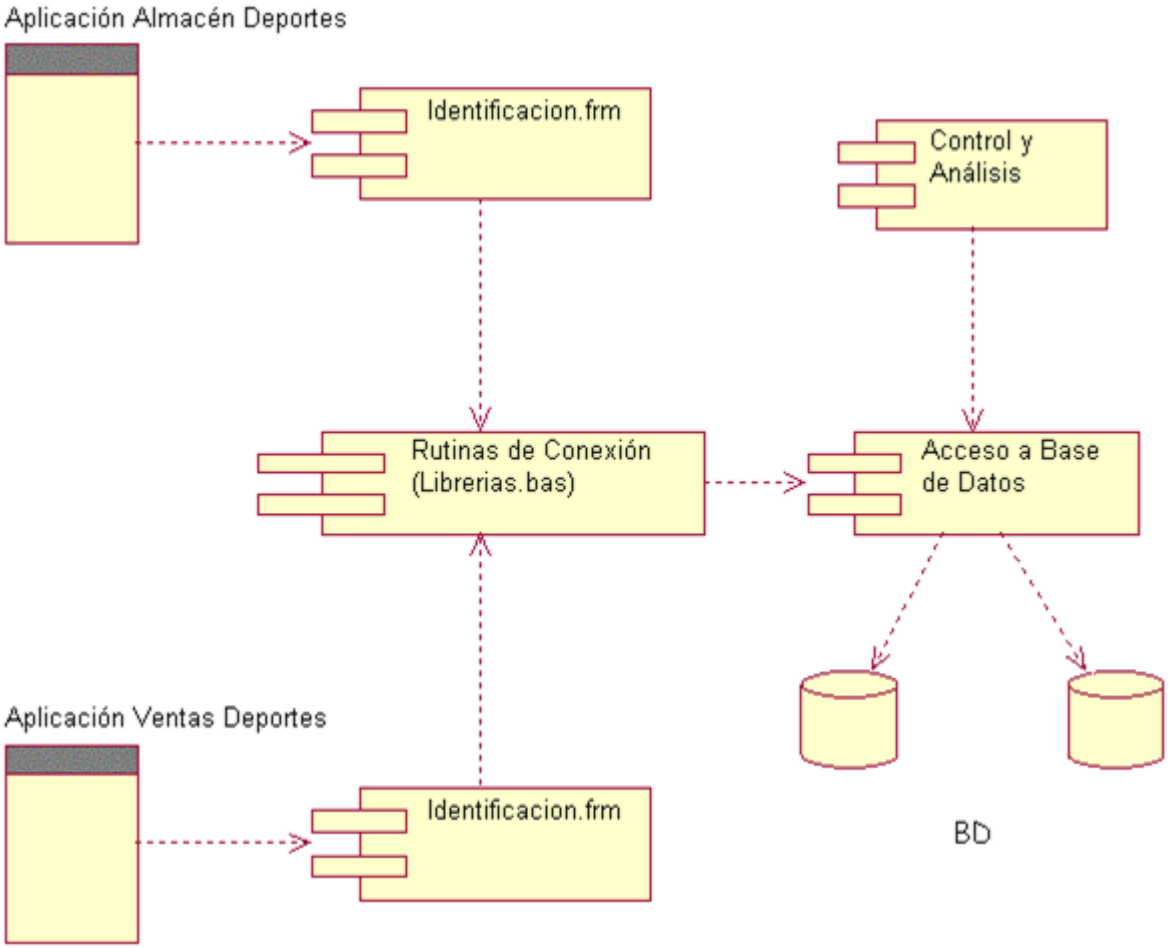


Diagrama de Despliegue

Diagrama de Despliegue

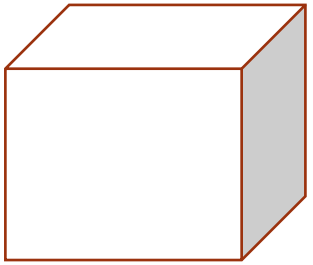
- Muestra la arquitectura del sistema considerando software y hardware durante el tiempo de ejecución
- Los elementos que lo componen son:
  - Nodos
  - Relaciones entre ellos
  - Componentes

Elementos del Diagrama

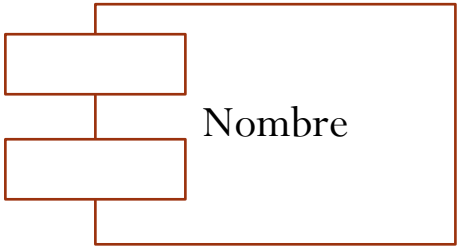
- Nodos. Representan objetos físicos durante el tiempo de ejecución
- Relaciones. Representan una comunicación entre nodos
- Componentes. Los componentes que se utilizan y que forman parte de un nodo

Representación

- Nodos



- Componentes



Artefactos

- Elementos del mundo físico que resultan del proceso de desarrollo:
  - Archivos
  - Bibliotecas
  - Bases de Datos entre otros

Ventajas

- Se presenta una idea clara del hardware en donde se estarán ejecutando los componentes del sistema