

**Prácticas Ingeniería del Software 3º**



# **Una Herramienta CASE para ADOO: Visual Paradigm**

**Análisis y Diseño Orientado  
a Objetos**



UNIVERSIDAD DE  
CASTILLA-LA MANCHA

ES de Informática de Ciudad Real García /  
Ismael Caballero

# Modelado Visual

- **Modelado Visual**

- **Modelado Visual** es el proceso que permite representar gráficamente el sistema software, permitiendo resaltar los detalles más importantes.
- Un buen modelo
  - Identifica requisitos y comunica información
  - Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos.
  - Permite ver las relaciones entre los componentes del diseño
  - Mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

# Visual Paradigm

- **Visual Paradigm**

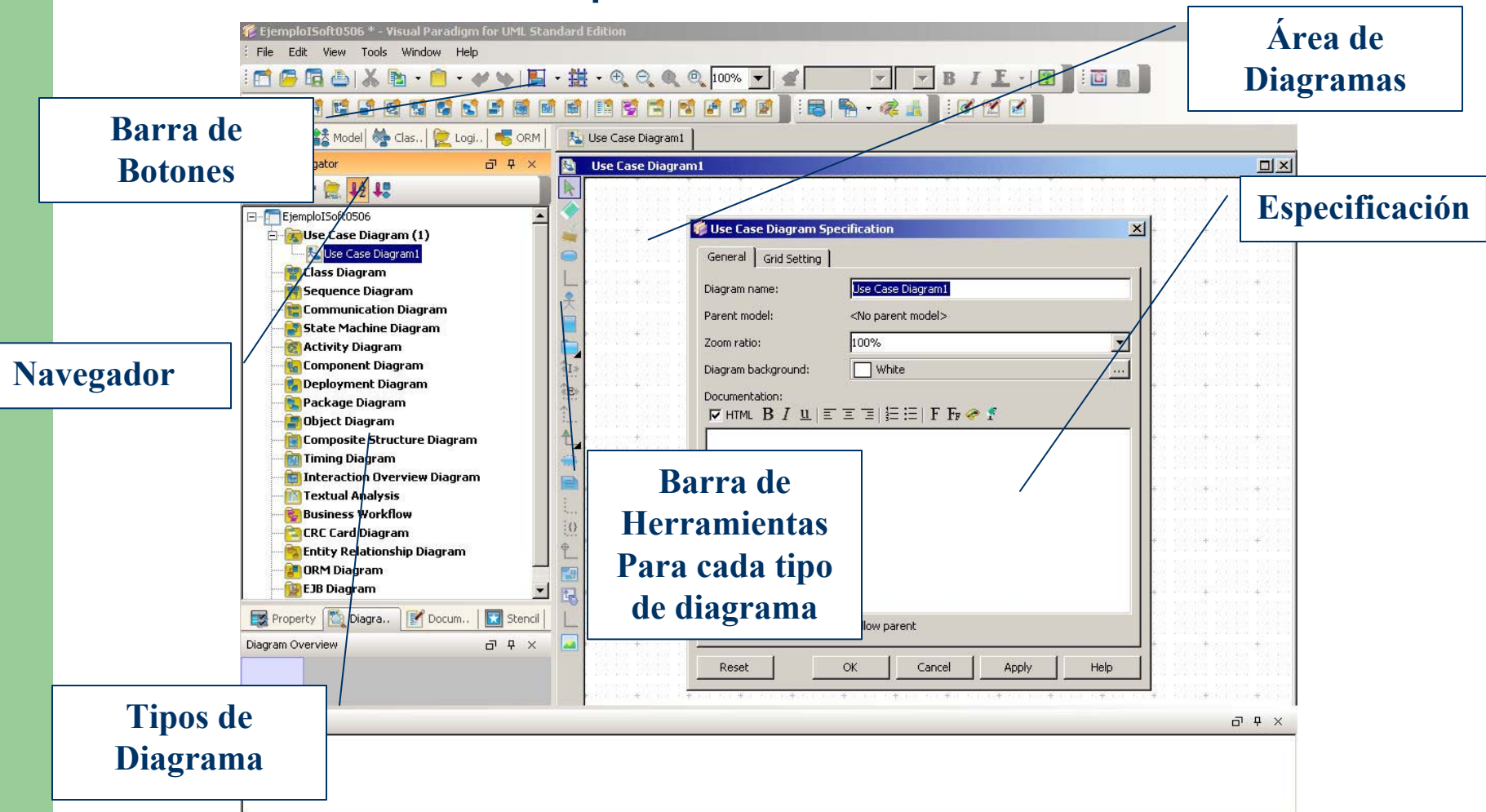
- Herramienta CASE que da soporte al **modelado visual** con UML 2.0 ofreciendo distintas
- Se puede descargar una versión gratuita (licencia para Community Edition) de: <http://www.visual-paradigm.com/product/vpuml/productinfovpumlce.jsp>
- Se pueden descargar manuales en PDF de :
  - <http://www.visual-paradigm.com/product/vpuml/vpumluserguide.jsp?format=pdf>
  - Página de la asignatura.

## Visual Paradigm (II)

- Visual Paradigm ofrece:
  - Entorno de creación de diagramas para UML 2.0
  - Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
  - Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
  - Capacidades de ingeniería directa (version profesional) e inversa.
  - Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
  - Disponibilidad de múltiples versiones, para cada necesidad.
  - Disponibilidad de integrarse en los principales IDEs.
  - Disponibilidad en múltiples plataformas
- Más información sobre VP en: <http://www.visual-paradigm.com/>

# IGU de Visual Paradigm

- Los elementos que ofrecen:



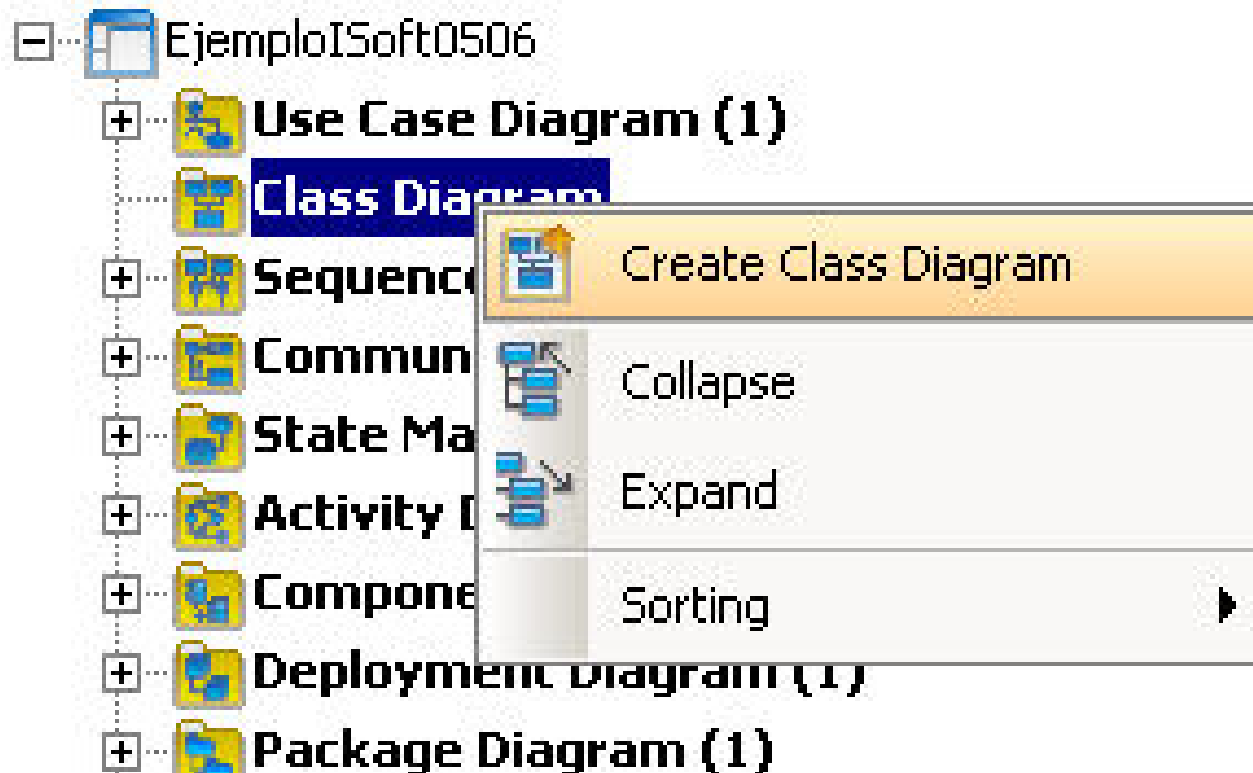
# Diagramas en Visual Paradigm

- **Diagramas**

- Representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos.
- Diferentes tipos de diagrama permiten ver el sistema desde diferentes perspectivas
- Los tipos de diagramas que se pueden incluir son:
  - Casos de Uso
  - Clase
  - Actividad
  - Estado
  - Componentes
  - Despliegue
  - Secuencia

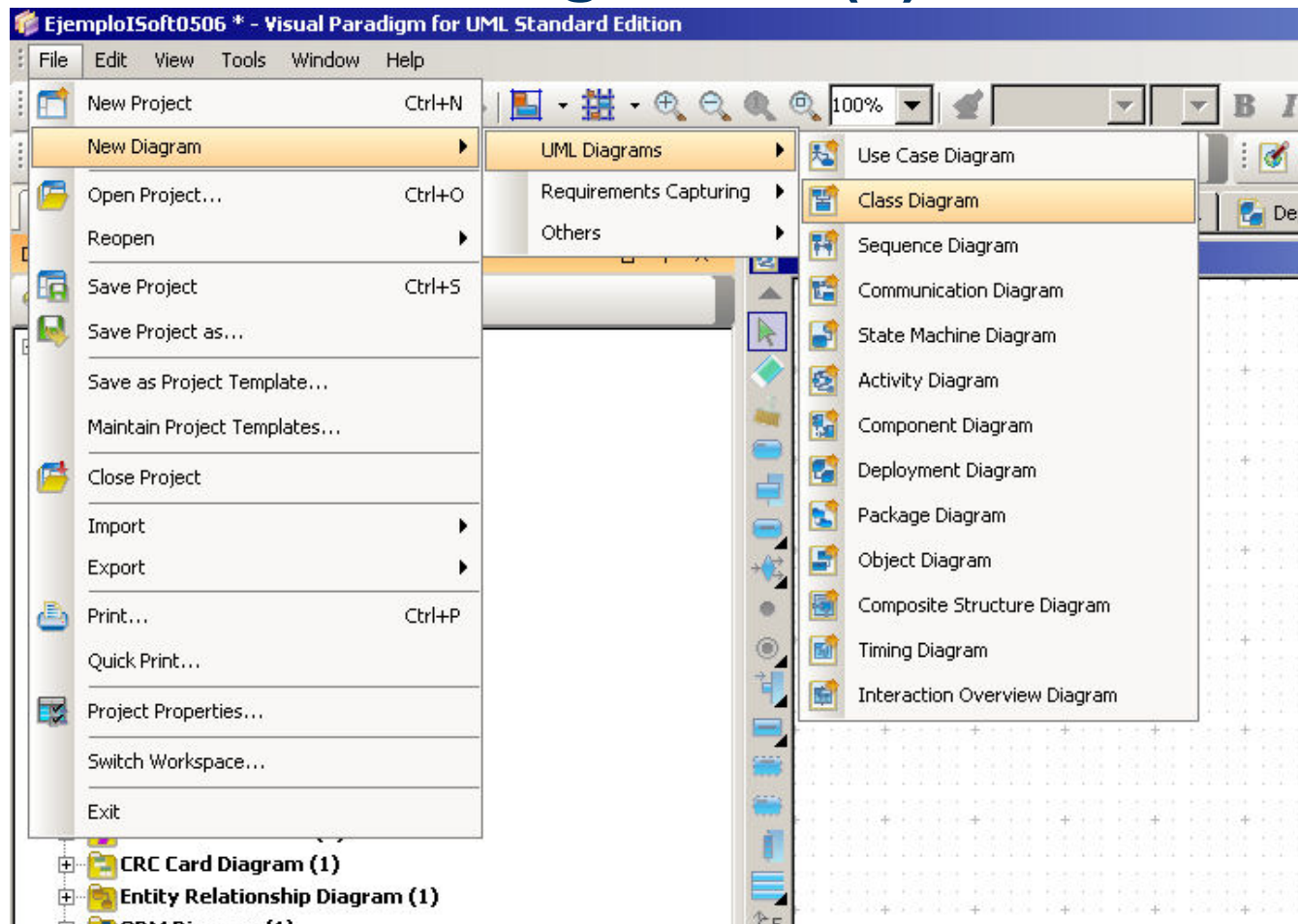
## Diagramas en Visual Paradigm (II)

- Creación de Diagramas



# Diagramas en Visual Paradigm (III)

- Creación de Diagramas (II)





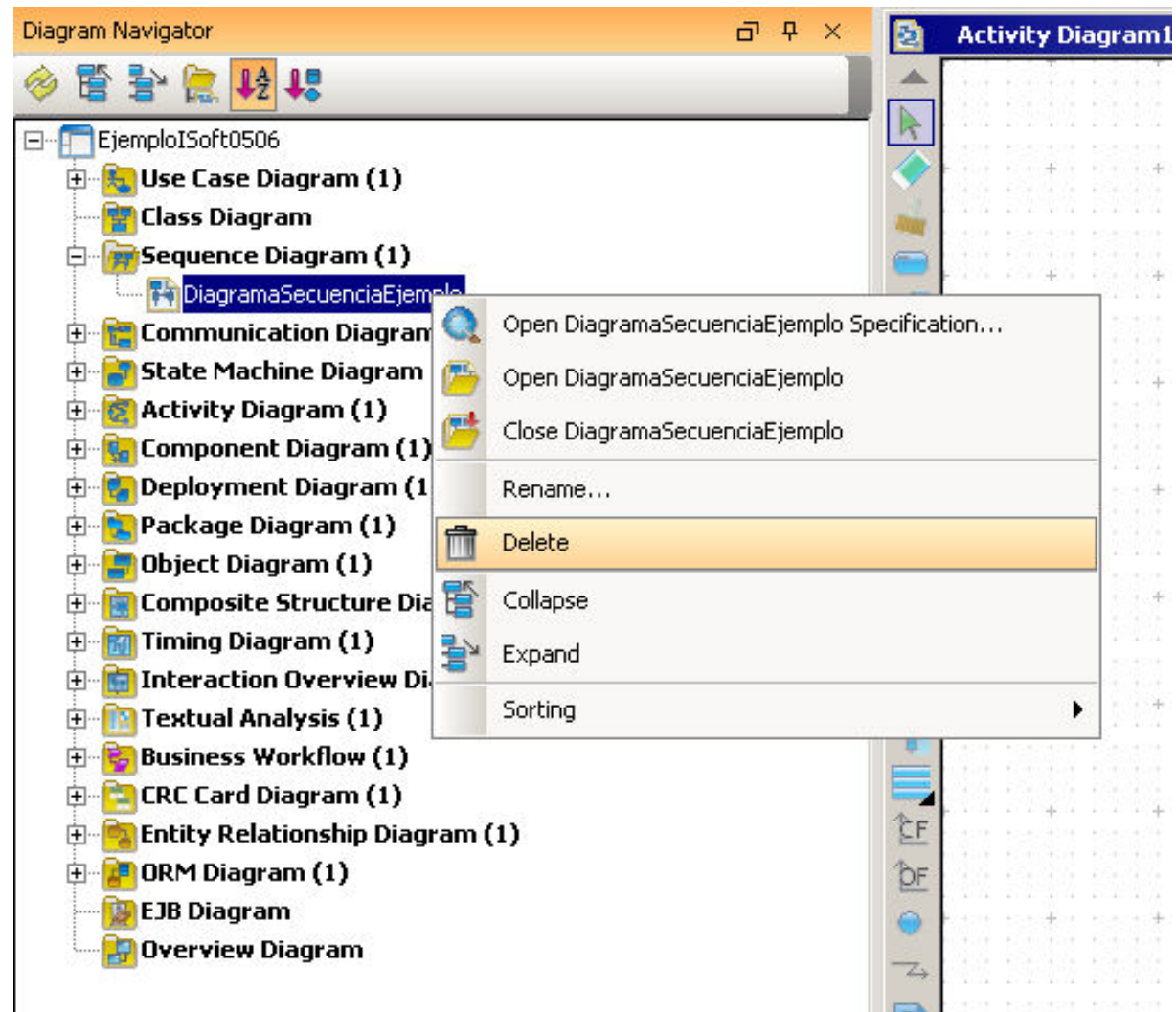
## Diagramas en Visual Paradigm (III)

- **Enlazado de diagramas**

- Se pueden enlazar diagramas mediante notas.
  - En el diagrama origen del enlace, insertando una nota
  - Arrastrando el diagrama destino desde el browser hasta esa nota
  - Haciendo clic en la nota se irá al diagrama destino

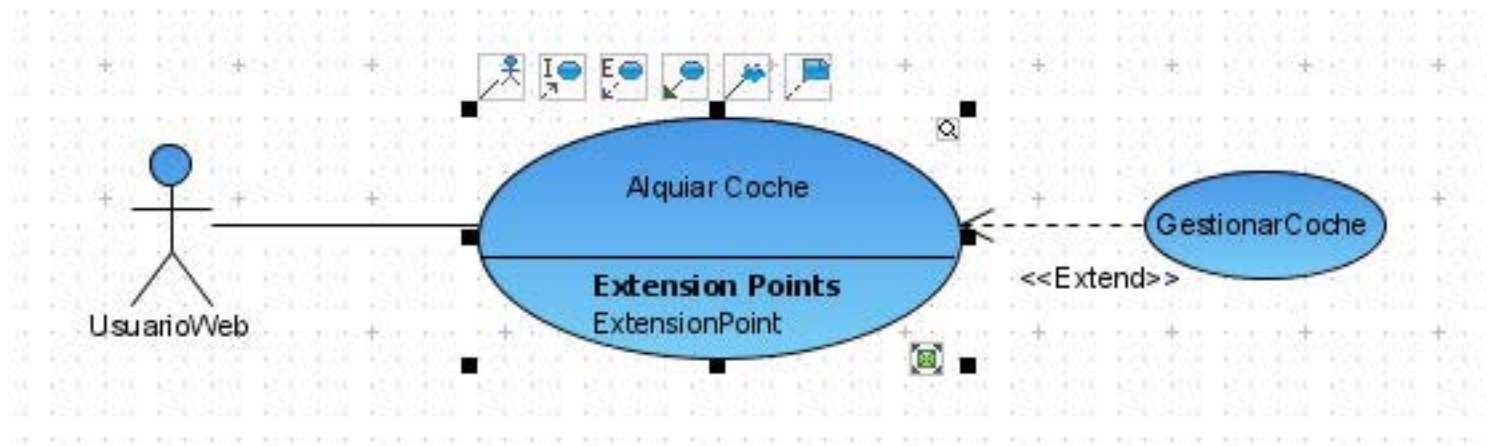
## Diagramas en Visual Paradigm (IV)

- Borrado de Diagramas



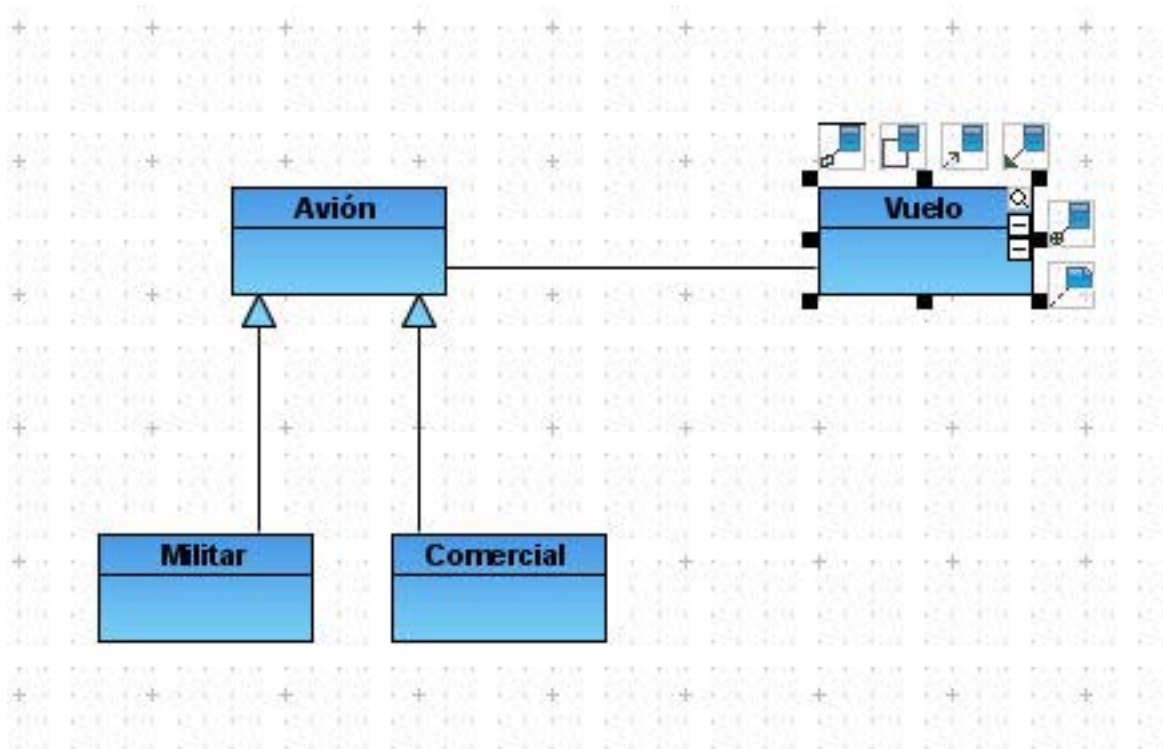
# Tipos de Diagramas

- **Diagramas de Caso de Uso**
  - Representan la funcionalidad del sistema



## Tipos de Diagramas (II)

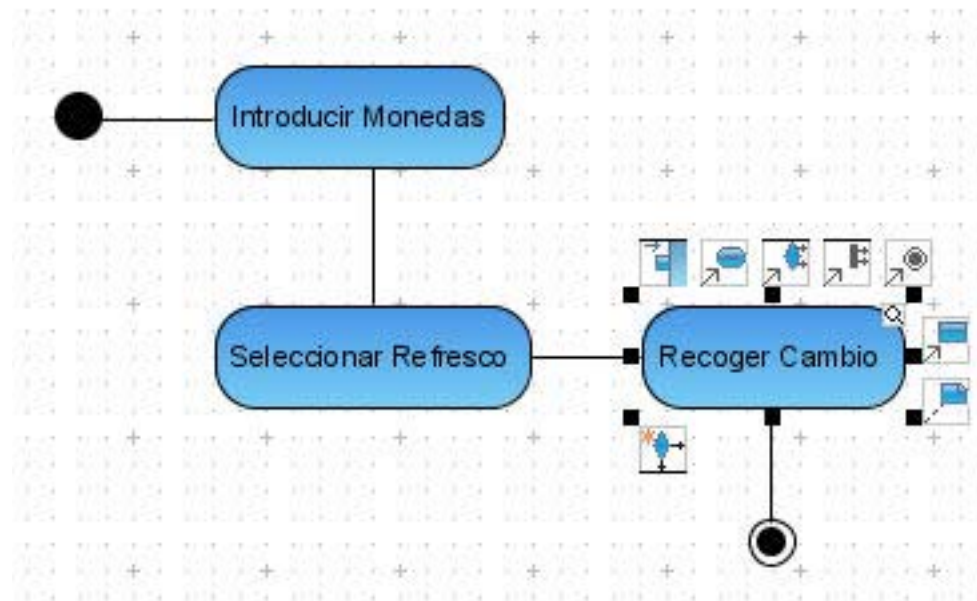
- **Diagramas de Clase**
  - Describen la estructura (estática) del sistema



## Tipos de Diagramas (III)

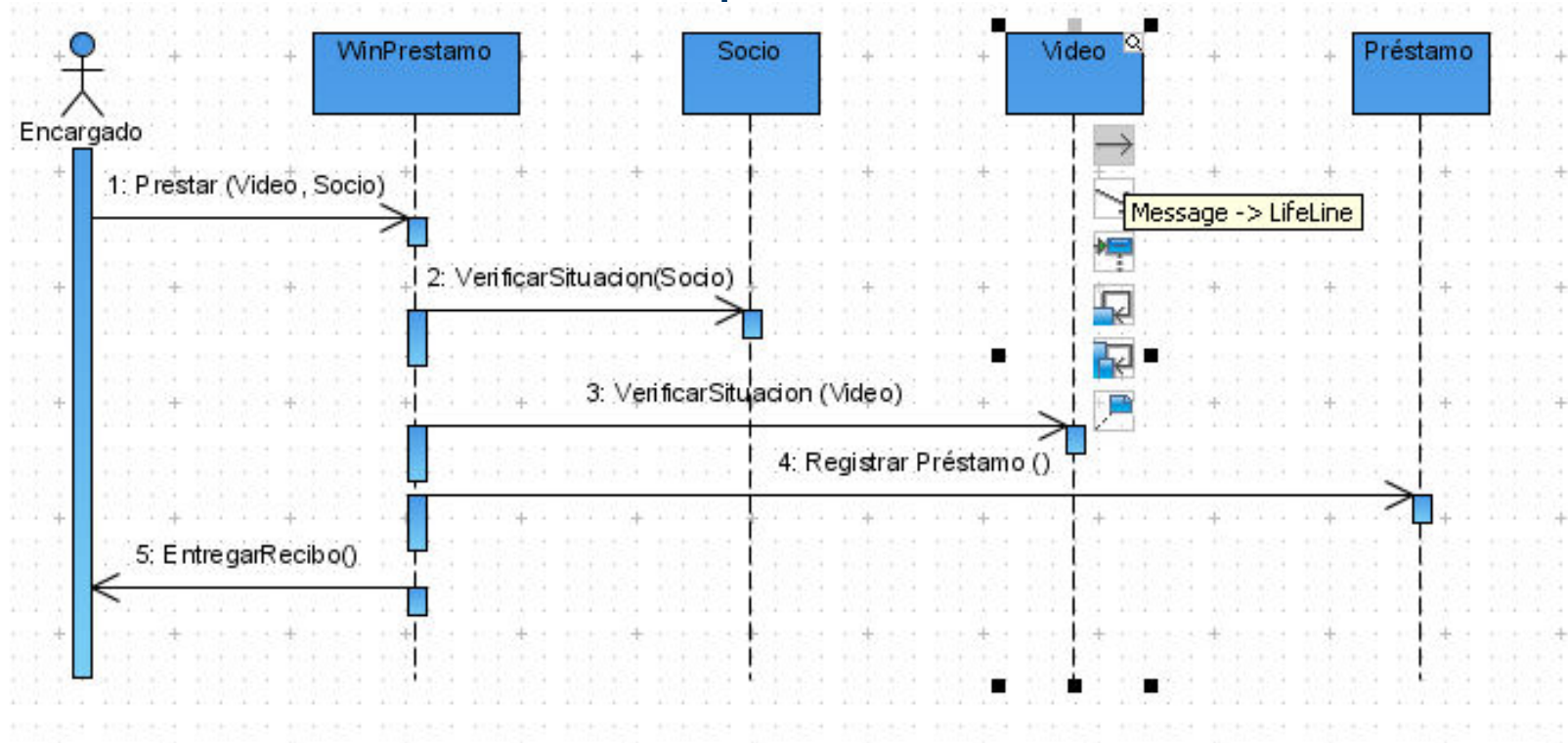
- **Diagrama de Actividad**

- Describen como se desarrolla un flujo de actividades entre elementos del sistema o del dominio.



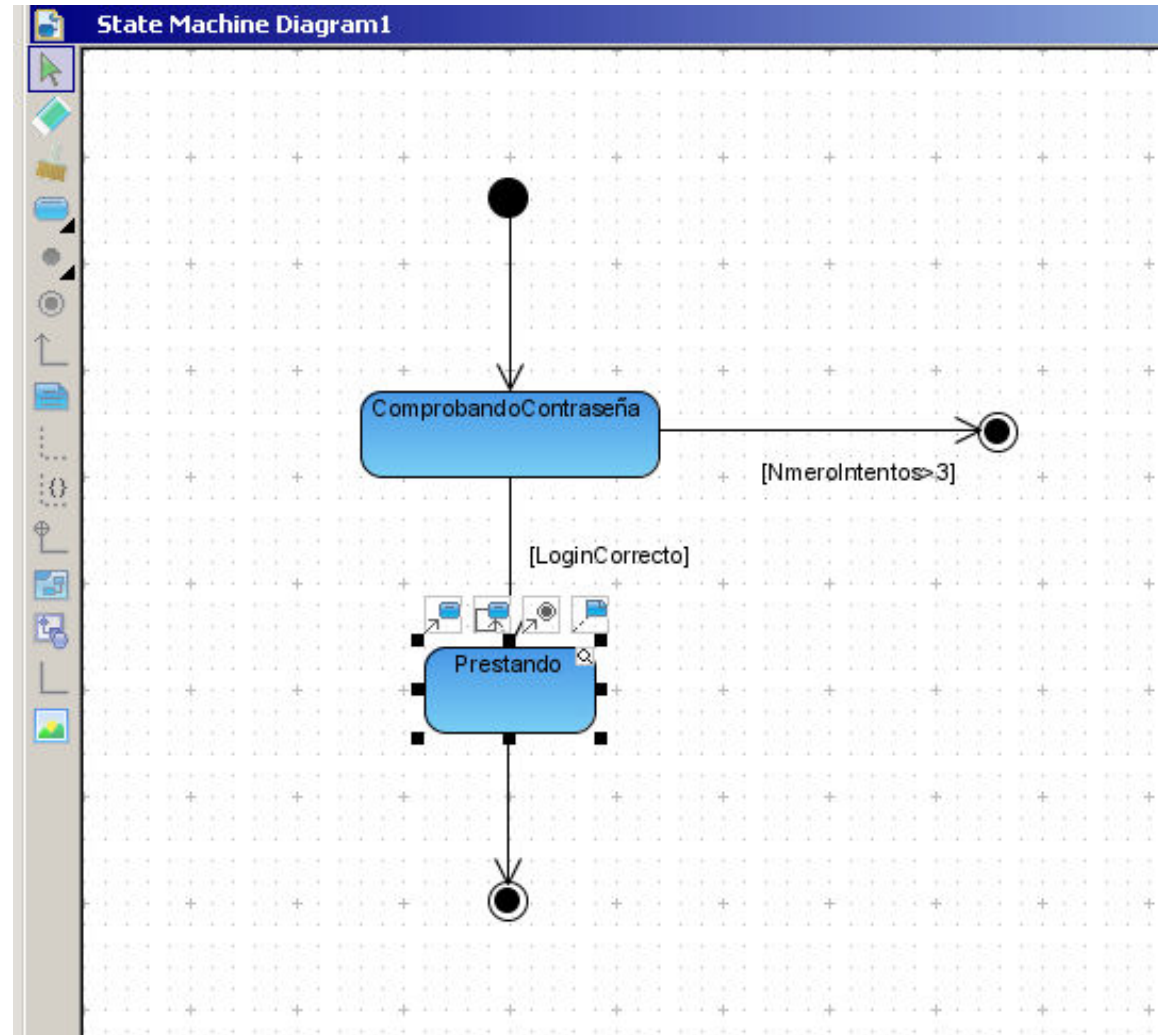
## Tipos de Diagramas (IV)

- **Diagramas de Secuencia**
  - Describen la interacción entre elementos del sistema en el tiempo



## Tipos de Diagramas (VI)

- **Diagramas de Estado**
  - Describen el estado, condiciones y respuesta de los elementos del sistema

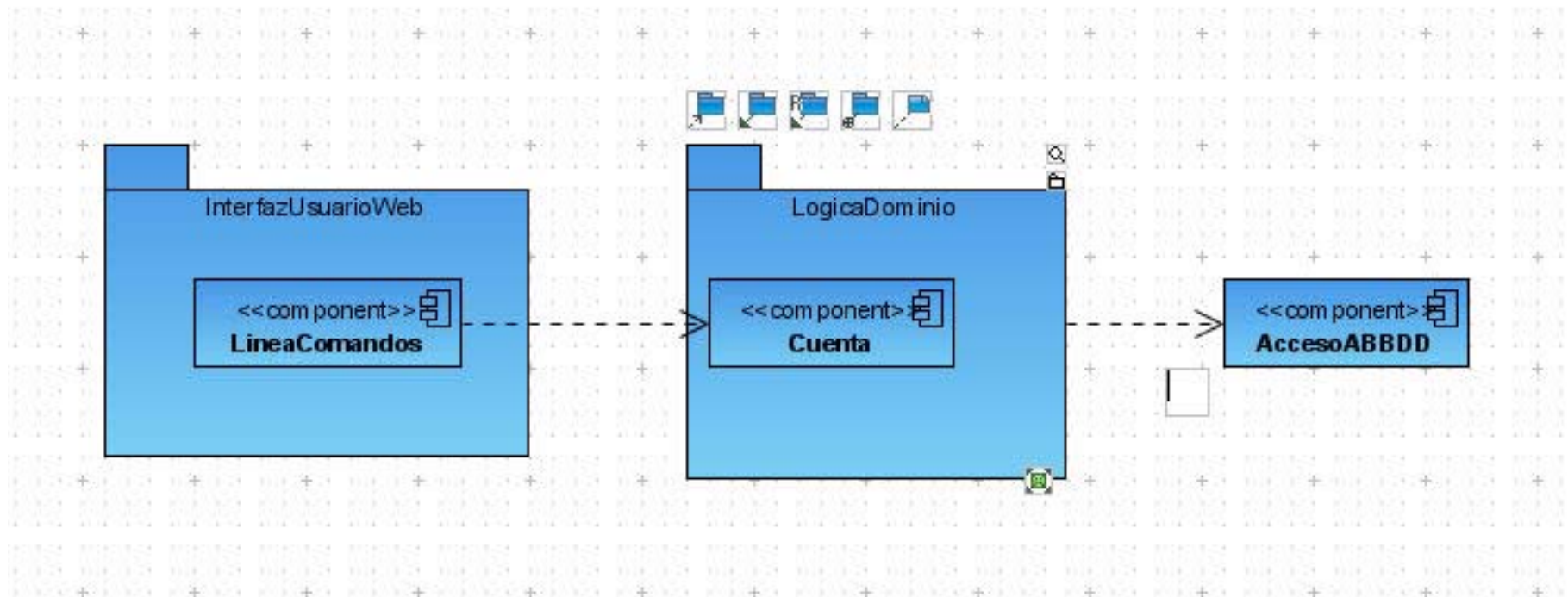




## Tipos de Diagramas (VII)

- **Diagrama de Componentes**

- Describen la organización de los elementos físicos que implementan sistema

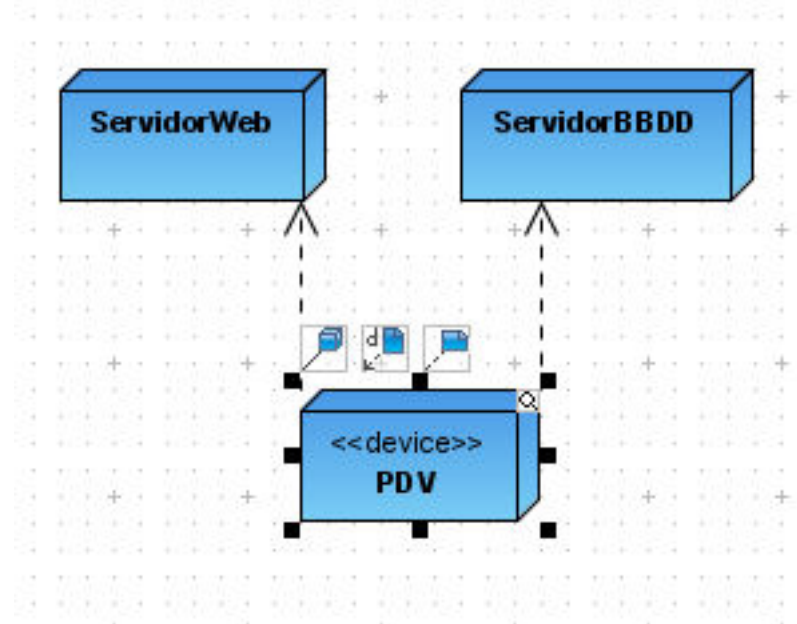




## Tipos de Diagramas (VIII)

- **Diagrama de Despliegue**

- Describen la configuración del entorno de máquinas y redes sobre el que se distribuyen componentes y procesos del sistema



# Paquetes de Trabajo

- **Características**

- Ofrecen un mecanismo general para la organización de los modelos/subsistemas/capas agrupando elementos de modelado
  - Cada paquete se corresponde a un submodelo (subsistema) del modelo (sistema)
  - Se pueden anidar paquetes
  - Una clase de un paquete puede aparecer en otro paquete por importación a través de una relación de dependencia entre paquetes.

## Problemas

1. Investigar sobre otras herramientas CASE existentes en el mercado para modelado visual.
2. Investigar las versiones de Visual Paradigm.
3. Identificar las distintas características del IGU de Visual Paradigm.
4. Reproducir distintos diagramas que se han propuesto como ejemplo en clase de teoría, identificando los elementos que participan en ellos.