

UML

Elaboración de Diagramas

La importancia de la comunicación

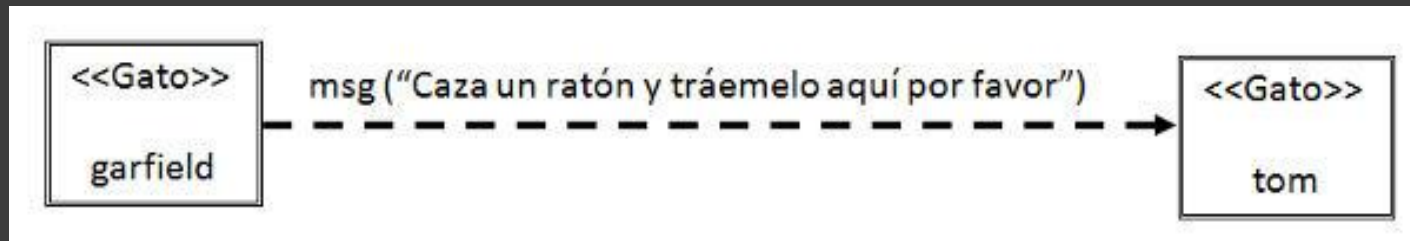


**UNIFIED
MODELING
LANGUAGE™**



Lenguaje Unificado de Modelado

Representar la información



Definición de UML

- ❑ Estándar adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

Definición de UML

- ❑ NO es un lenguaje de programación.
- ❑ Serie de normas y estándares para modelar, construir y documentar un sistema software.

Definición de UML

- ❑ Esquema teórico de un sistema y que ayuda a facilitar su comprensión.
- ❑ Representación gráfica o esquemática de una realidad desde diferentes perspectivas.

Utilidad UML

- ❑ Facilitar la comunicación entre todos los miembros del equipo de desarrollo.
- ❑ Documentar:
 - ✓ Requisitos
 - ✓ Objetivos
 - ✓ Arquitectura
 - ✓ Pruebas
 - ✓ Versiones
 - ✓ Librerías
 - ✓ Frameworks

Utilidad UML

- ❑ Representar la arquitectura del sistema.
 - ✓ Estructura del software
 - ✓ Organización del software
 - ✓ Componentes del sistema
 - ✓ Interfaces físicos y lógicos entre componentes
 - ✓ Comportamiento del sistema
- ❑ Especificar y definir de forma precisa los modelos de decisiones tomadas en todas las fases:
 - ✓ Análisis
 - ✓ Diseño
 - ✓ Implementación
- ❑ Programas que a partir del modelo generan código.

Programas para diseñar con UML

- ❑ Visio
- ❑ Lucidchart
- ❑ Gliffy
- ❑ Draw.io
- ❑ ArgoUML
- ❑ MagicDraw

Elementos

- 1) Estructuras: Representa un nodo en el diagrama
- 2) Relaciones: Representa la asociación de un nodo con otro.
- 3) Notas: Comentarios para entender mejor el diagrama.
- 4) Agrupaciones: Agrupa elementos y facilita modelar los diagramas grandes.

Tipos de diagramas

❑ Estructurales: Visión estática del sistema.

- ✓ Diagrama de clases.
- ✓ Diagrama de objetos.
- ✓ Diagrama de componentes.
- ✓ Diagrama de despliegue.
- ✓ Diagrama de paquetes.
- ✓ Diagrama de estructura compuesta.

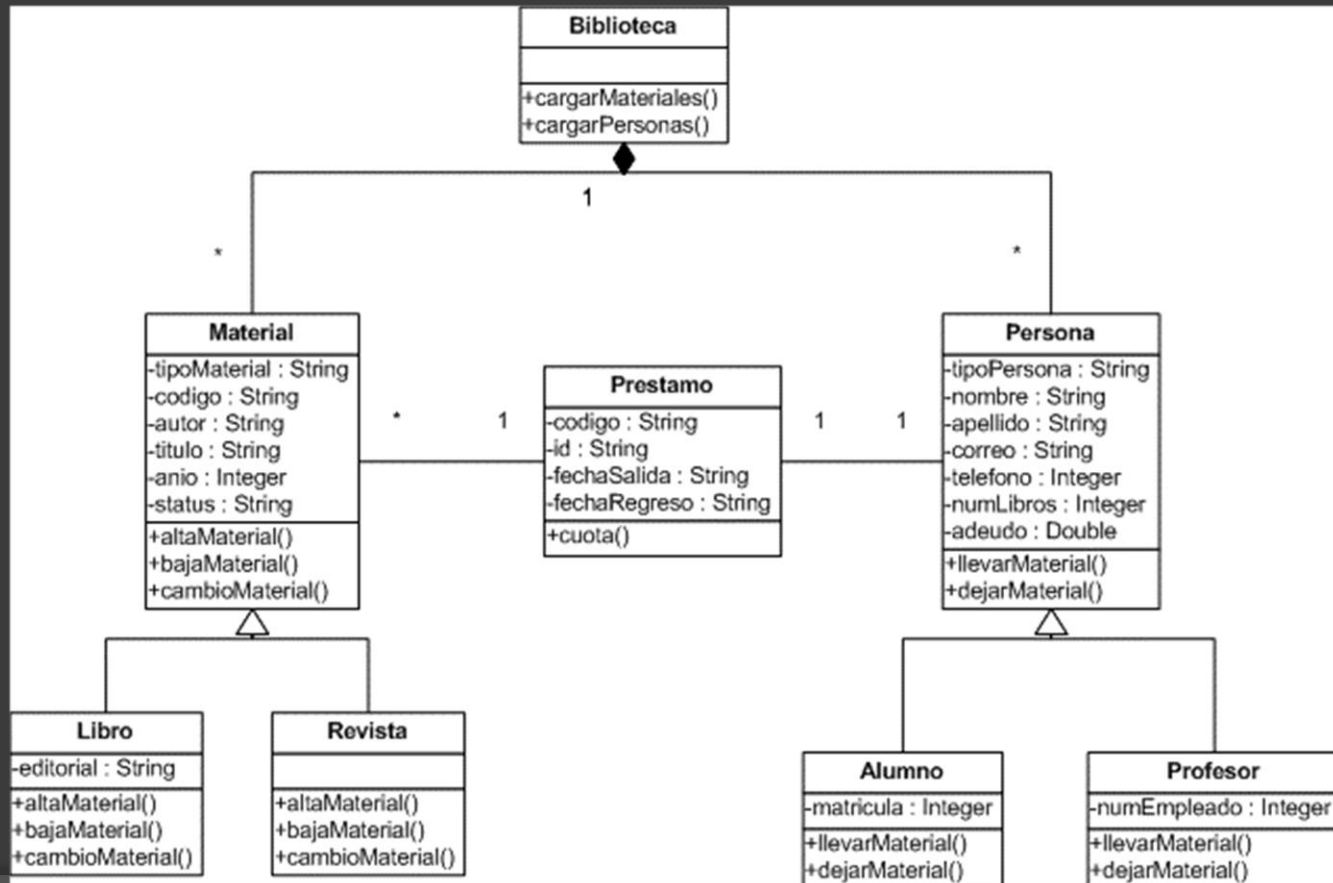
❑ Comportamiento: Visión dinámica del sistema.

- ✓ Diagrama de casos de uso.
- ✓ Diagrama de secuencia.
- ✓ Diagrama de estados.
- ✓ Diagrama de actividades.

Diagramas estructurales

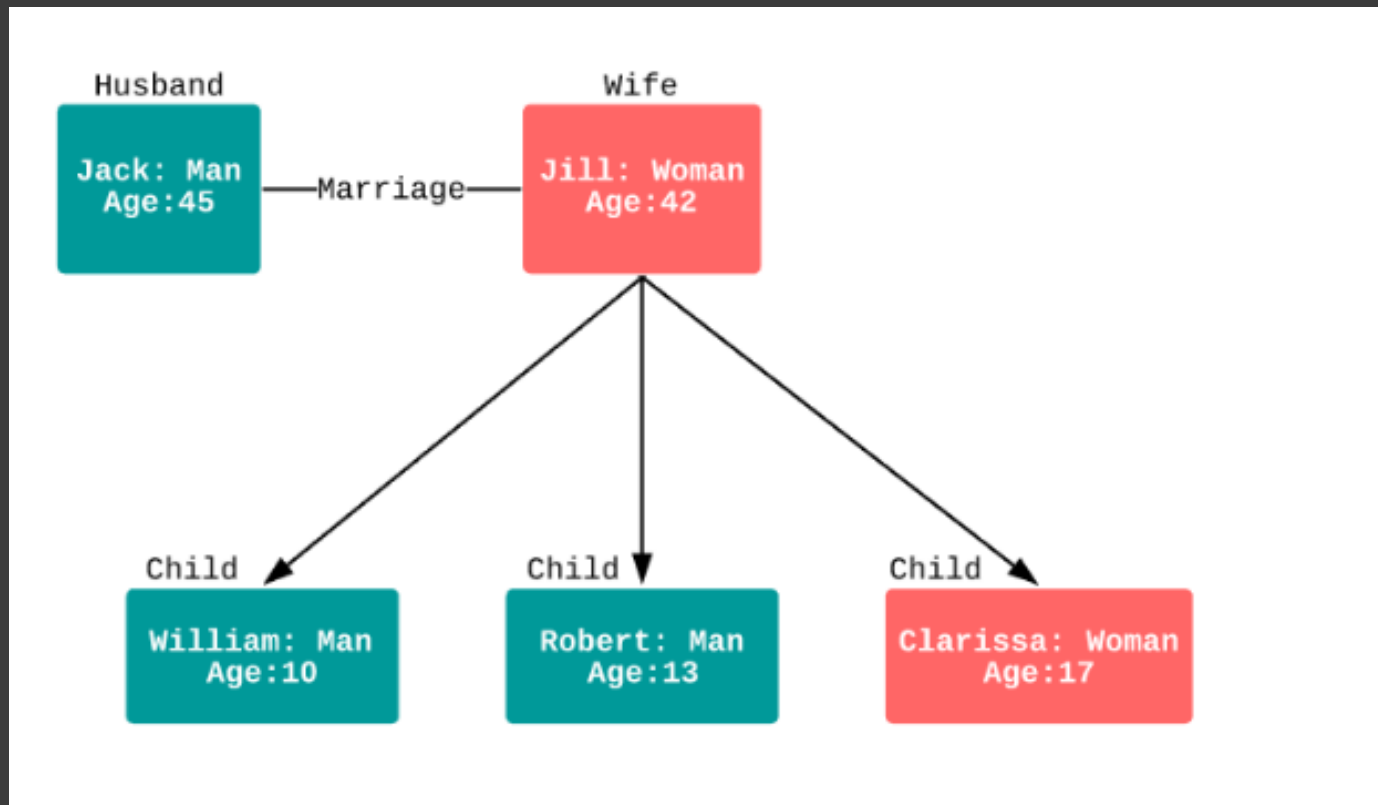
Diagramas de clases

Muestra las clases que componen el sistema y como se relacionan entre ellas.



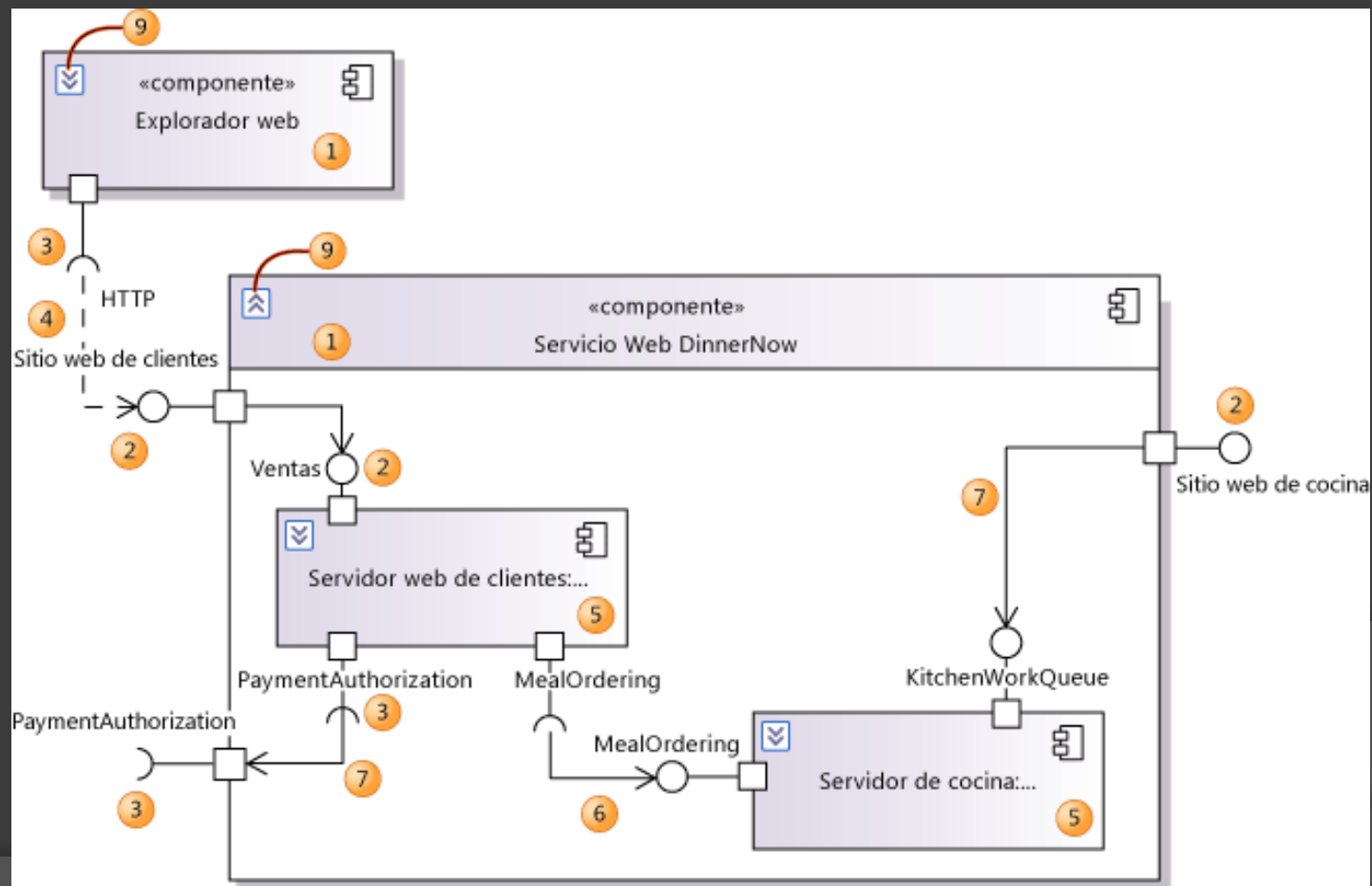
Diagramas de objetos

Muestra relación entre objetos del mundo real en un momento determinado.



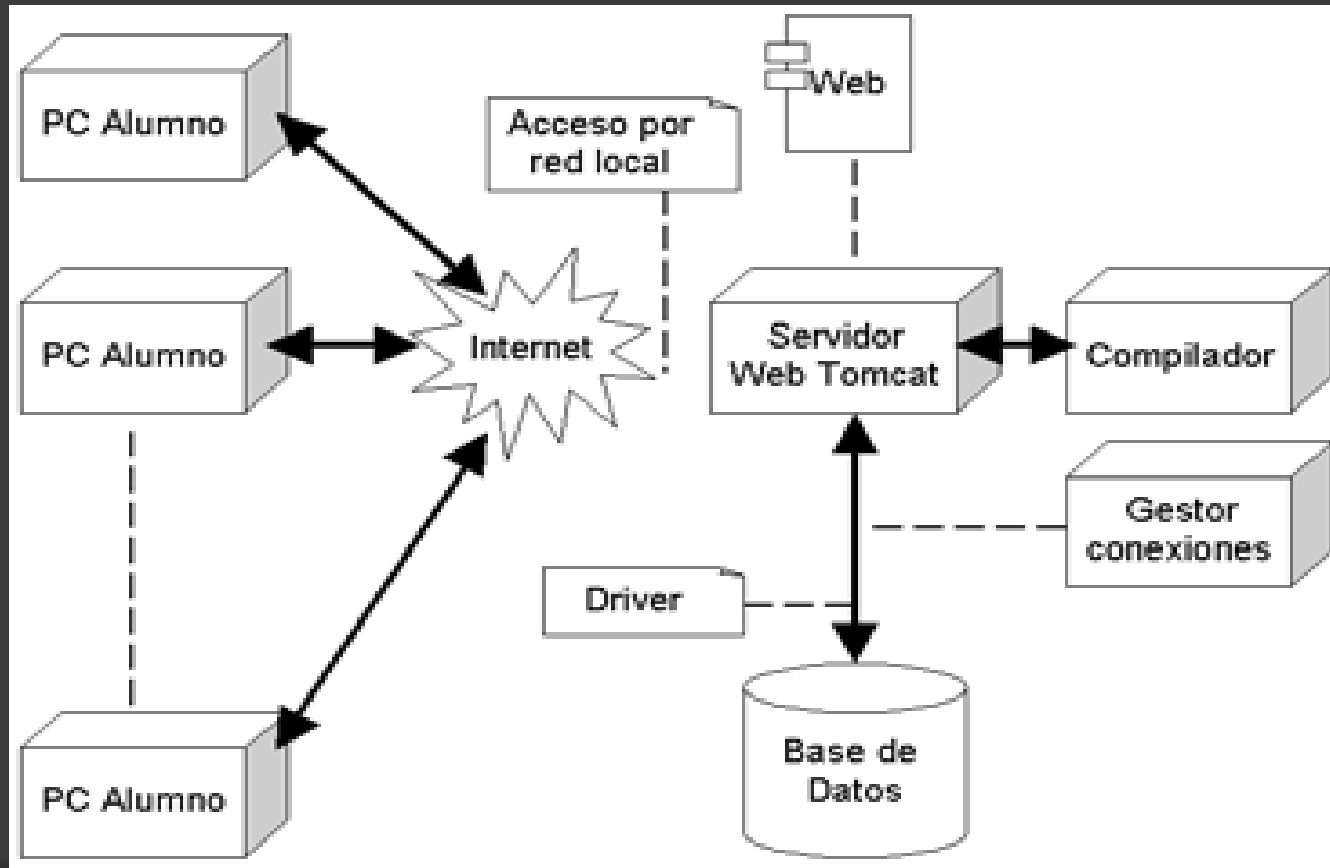
Diagramas de componentes

Describe el software, controladores, framework, librerías, ... que conforman el sistema.



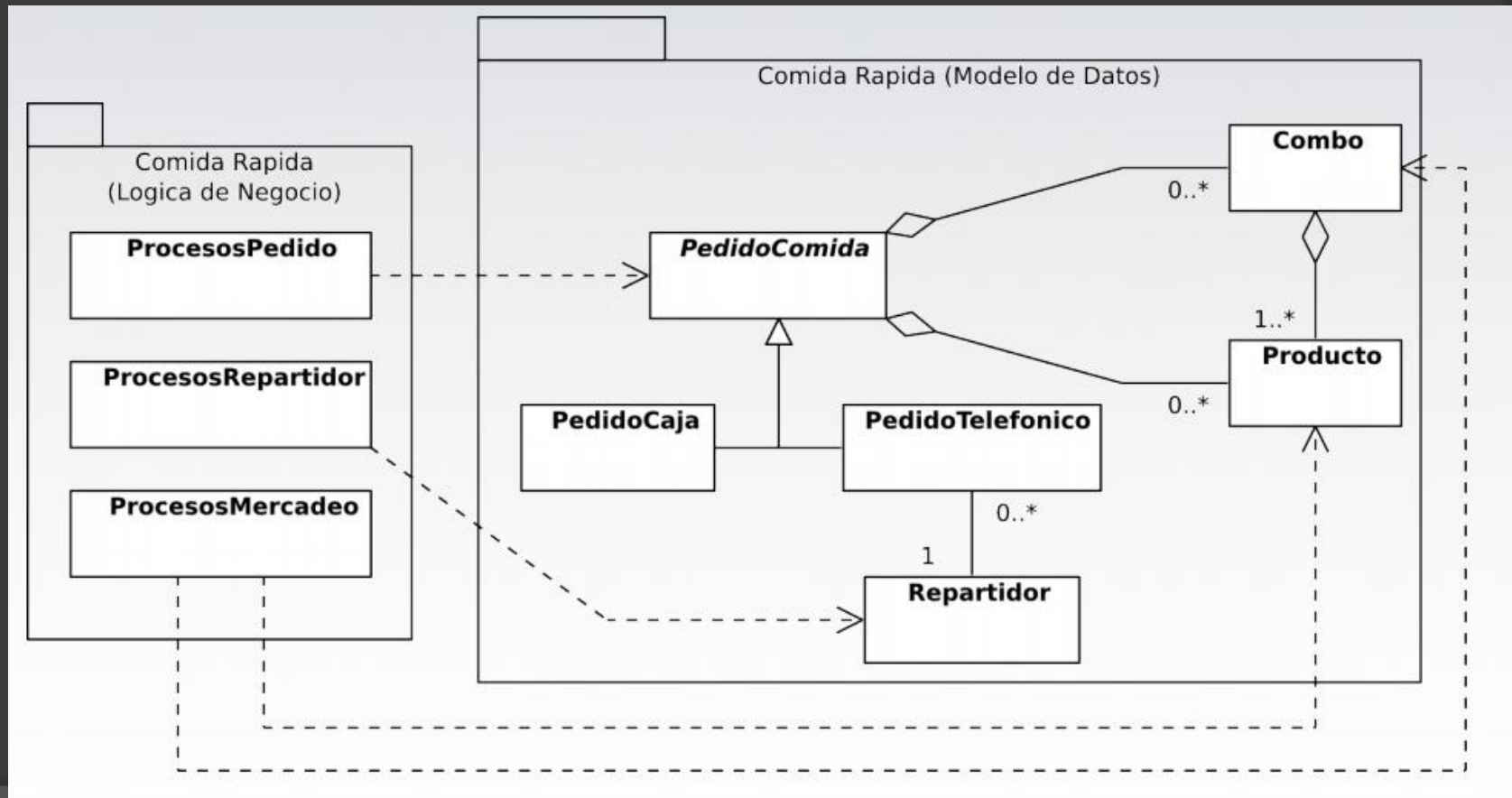
Diagramas de despliegue

Muestra cómo y dónde se despliega el sistema: máquinas, servidores, interfaces...



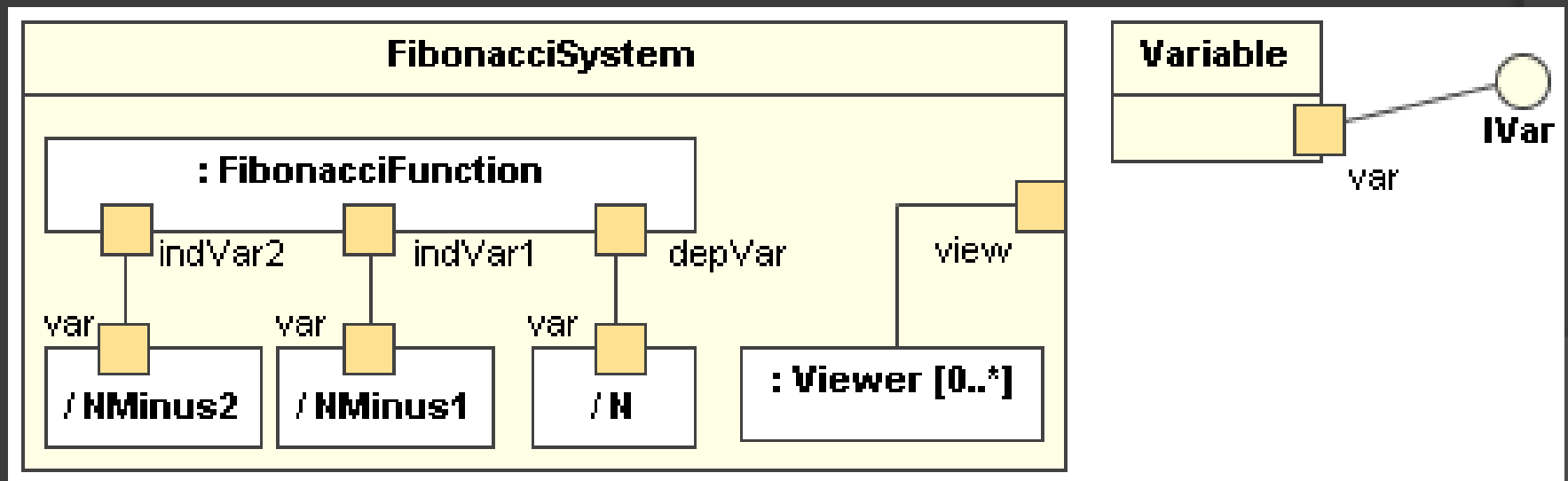
Diagramas de paquetes

Se utiliza para organizar los paquetes y sus elementos.



Diagramas de estructura compuesta

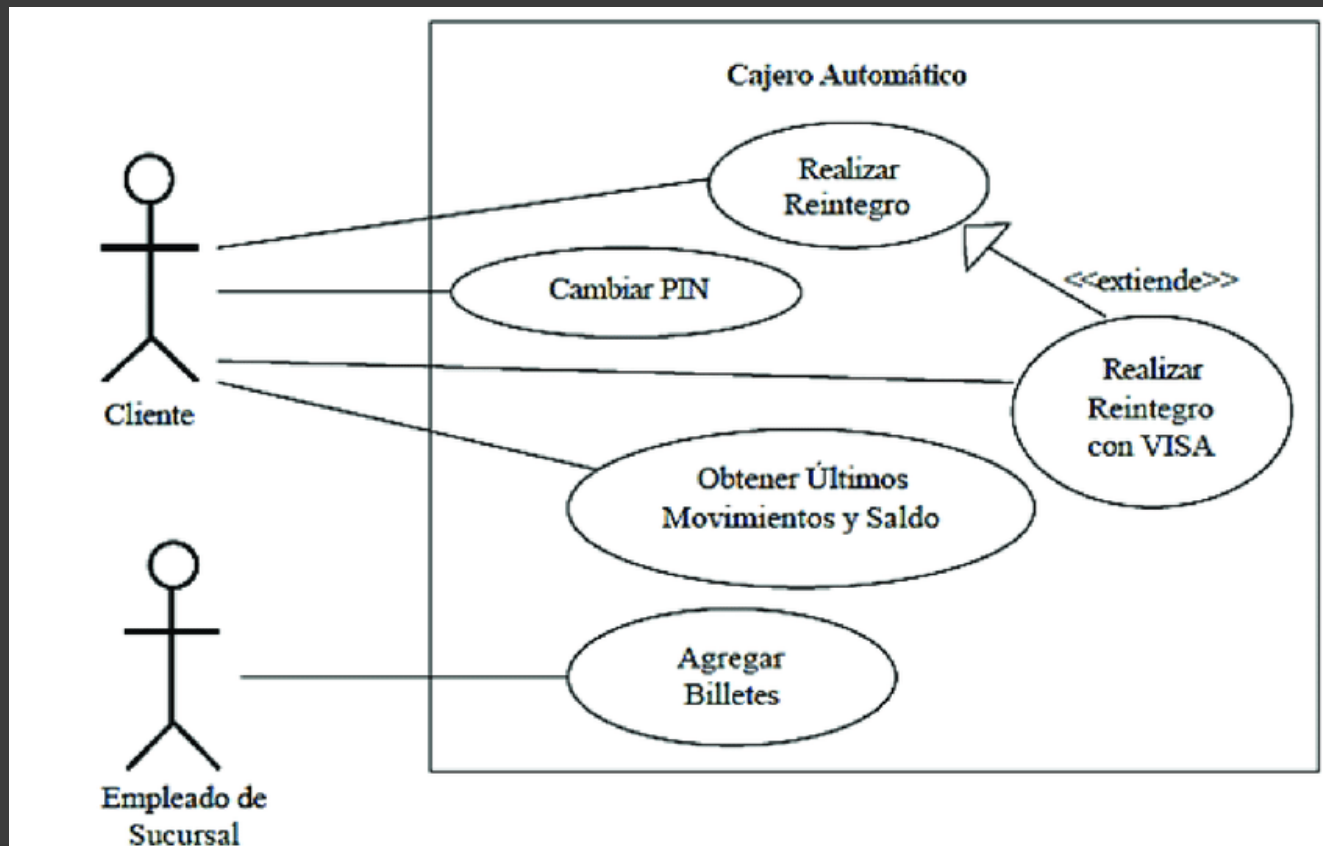
Muestra colaboración entre clases, componentes o interfaces para definir una funcionalidad del sistema.



Diagramas de comportamiento

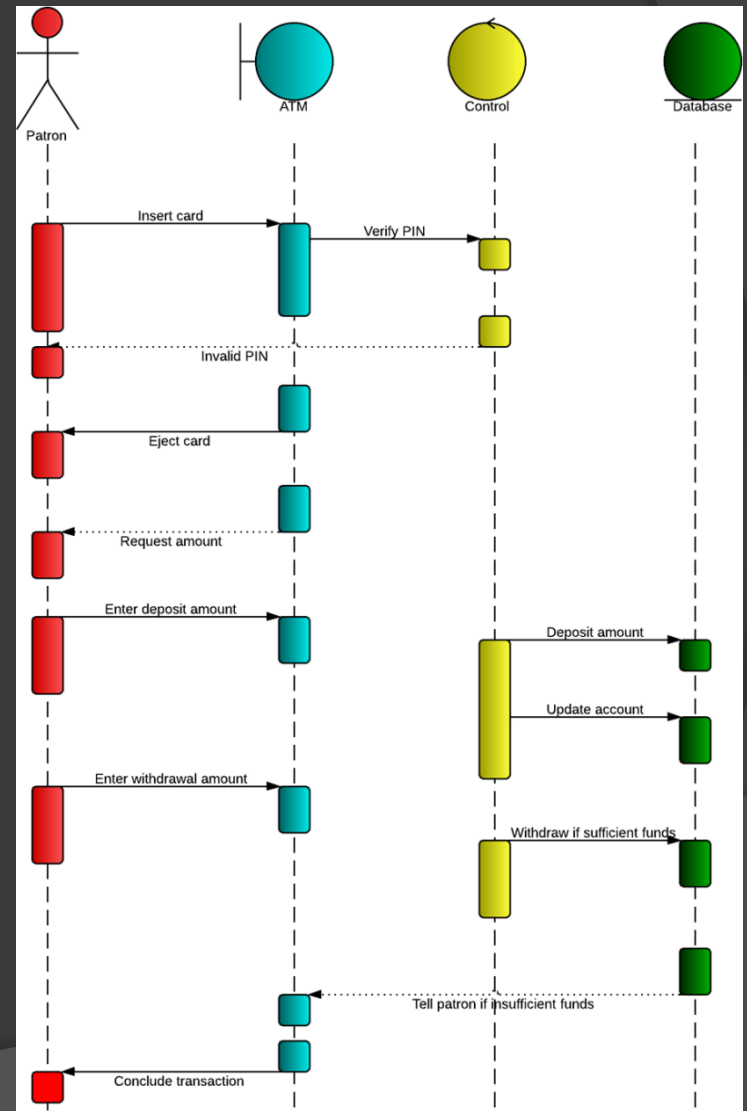
Diagramas de casos de uso

Muestra cómo se relacionan las funcionalidades con los controladores internos o externos.



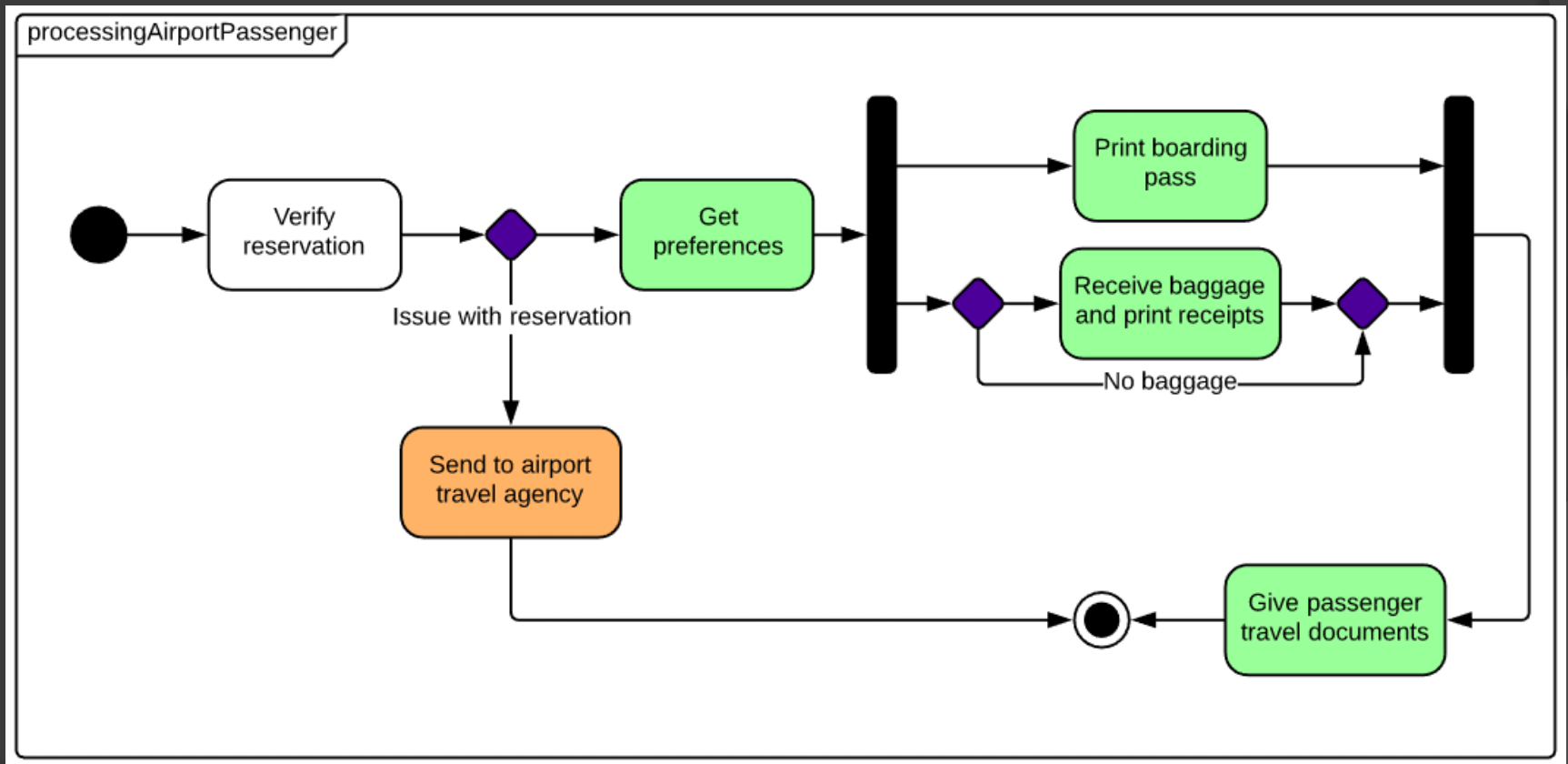
Diagramas de secuencia

Muestra cómo los objetos interactúan entre sí a lo largo del tiempo.



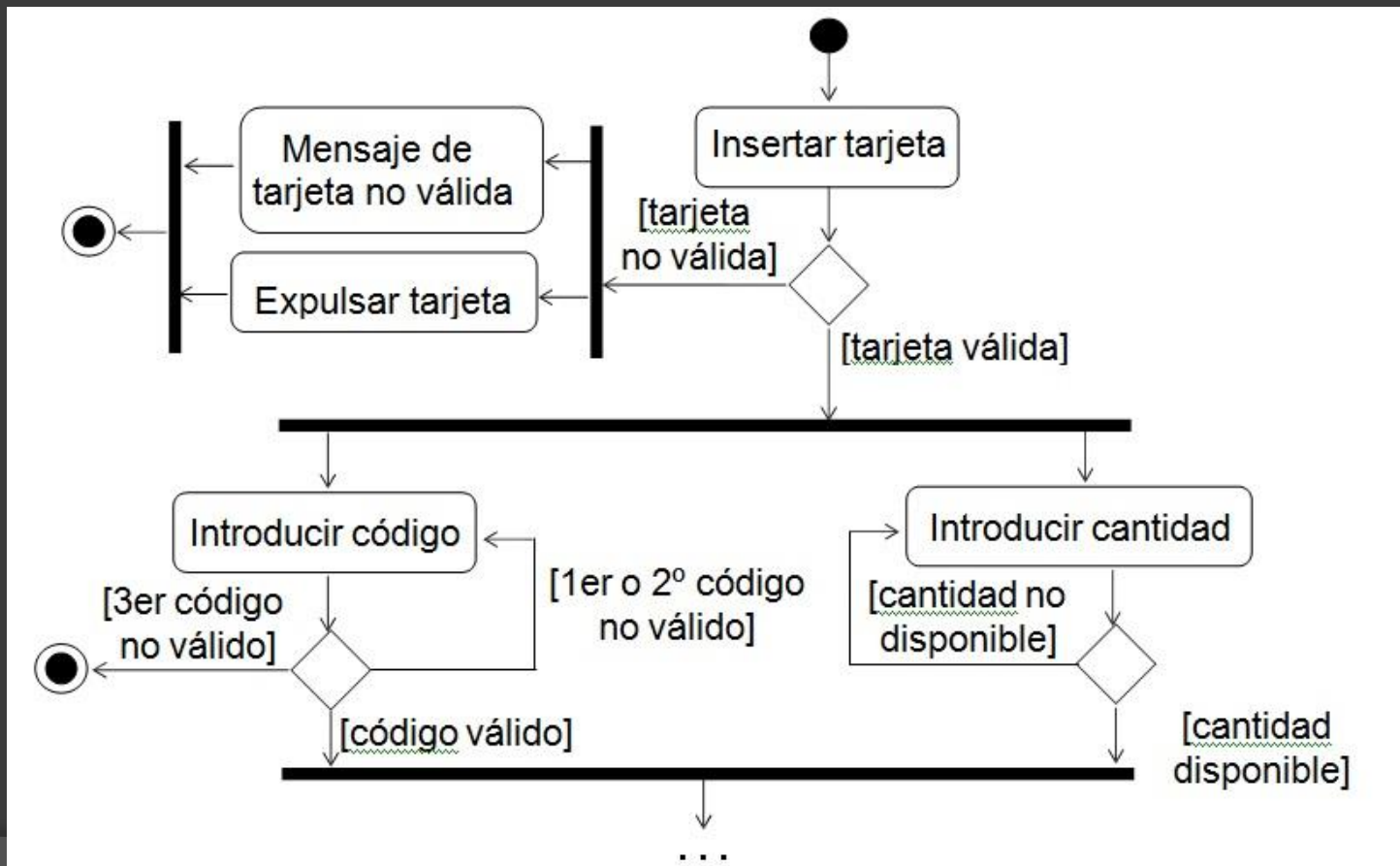
Diagramas de estados

Muestra el conjunto de estados por los que pasa un objeto durante la vida de la aplicación.



Diagramas de actividades

Muestra las interacciones internas entre objetos, como los pasos de un algoritmo.



**MUCHAS GRACIAS POR
VUESTRA ATENCIÓN**