



# Ejemplos UML

## Tema 4

Grupo 46

TACC II  
Curso 2008/09

# Indice



- Cajeros Automáticos
- Sistema de Gestión de Tráfico Ferroviario

*“Object-Oriented Analysis and Design with Applications, Third Edition”* Grady Booch; Robert A. Maksimchuk; Michael W. Engle; Bobbi J. Young Ph.D.; Jim Conallen; Kelli A. Houston. Addison Wesley Professional, 2007.

# Ejemplo de Análisis Orientado a Objetos

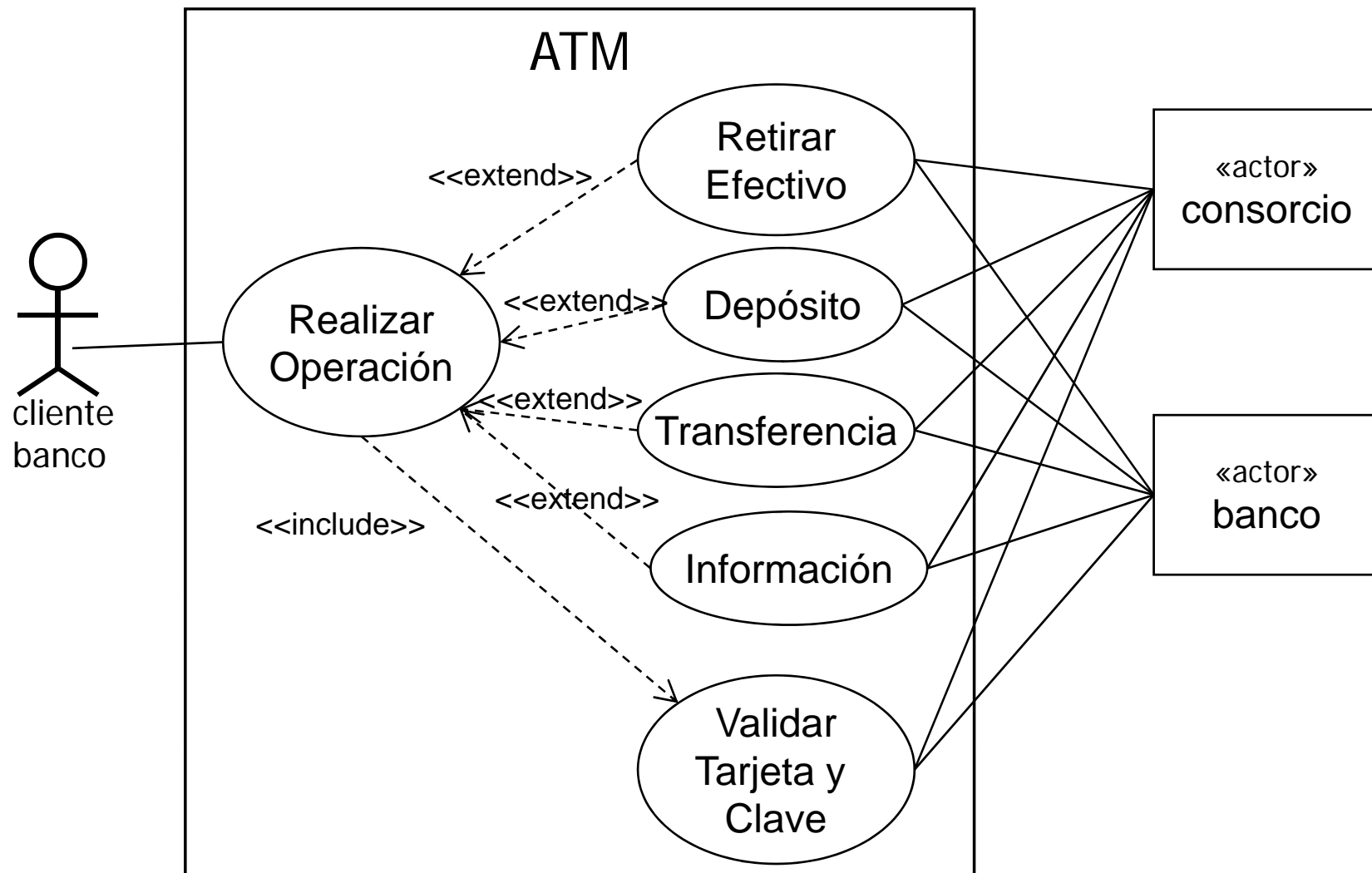
## ATMs

Se desea diseñar el software necesario para una red bancaria provista de cajeros automáticos (ATMs), que serán compartidos por un consorcio de bancos. Cada banco dispone de una serie de servidores, provistos de software propio, que llevan la información sobre sus cuentas y procesa las transacciones que actúan sobre dichas cuentas. A estos servidores están conectados las estaciones de cajero, que son propiedad del banco y en las que operan cajeros humanos, que pueden crear cuentas e introducir transacciones sobre ellas.

Los cajeros automáticos aceptan tarjetas de crédito, interaccionan con el usuario, se comunican con un ordenador central para llevar a cabo las transacciones, entregan dinero en efectivo al usuario e imprimen recibos. El sistema llevará el registro de las transacciones efectuadas, cumplirá características aceptables de seguridad y manejará accesos concurrentes a la misma cuenta.

El coste de desarrollo de la parte compartida del sistema se dividirá entre los bancos que forman parte del consorcio en función del número de clientes provistos de tarjetas de crédito.

# Diagrama de Casos de Uso



# Caso de Uso

## *Validar Tarjeta y Clave (Refinado)*

### **Actores primarios:**

Cliente del Banco, Consorcio, Banco

### **Interesados y Objetivos:**

**Cliente del Banco**: quiere realizar una operación con el ATM de manera rápida, para lo que debe validar su tarjeta y contraseña.

**Consorcio**: Quiere identificar correctamente el banco del cliente y mediar en la validación de manera eficaz.

**Banco**: Quiere identificar correctamente la identidad de la tarjeta.

### **Precondiciones:**

El cliente tiene una cuenta en uno de los bancos del consorcio, así como una tarjeta emitida por el mismo.

### **Garantía de éxito (post-condiciones):**

La tarjeta se valida correctamente.

# Caso de Uso

*Validar Tarjeta y Clave (Refinado)*

***Escenario Principal de Éxito:***

1. El ATM pide al cliente que inserte la tarjeta de crédito.
2. El cliente inserta la tarjeta de crédito.
3. El ATM acepta la tarjeta de crédito y lee el número de tarjeta y el código del banco.
4. El ATM pide la contraseña al cliente.
5. El cliente teclea la contraseña.
6. El ATM envía el número de tarjeta, el código del banco y la contraseña al consorcio.
7. El consorcio envía el número de tarjeta y la contraseña al banco correspondiente.
8. El banco notifica la aceptación al consorcio.
9. El consorcio notifica la aceptación al cajero automático.

# Caso de Uso

## Validar Tarjeta y Clave (Refinado)

### **Escenario Alternativo:**

#### 3a. La tarjeta es ilegible

1. El ATM notifica al cliente de que la tarjeta no se puede leer
2. El ATM expulsa la tarjeta.
3. El ATM vuelve a la situación inicial.

#### 8a. El banco notifica el rechazo al consorcio.

1. El consorcio notifica el rechazo al cajero automático.
2. El cajero automático notifica el rechazo al cliente y pide que teclee de nuevo la contraseña.
3. Se ha repetido este escenario alternativo menos de 3 veces y el flujo continua en 5 (en el escenario principal).

#### 3a. Se ha repetido este escenario alternativo más de 3 veces:

1. El ATM retene la tarjeta.
2. El ATM notifica al cliente que la tarjeta queda retenida.
3. El ATM notifica al consorcio que la tarjeta queda retenida.
4. El consorcio notifica al banco que la tarjeta queda retenida.
5. El ATM vuelve a la situación inicial.

... (timeouts de teclado, de comunicaciones, rotura de elementos mecánicos del cajero, etc.)

# Caso de Uso

## *Validar Tarjeta y Clave (Refinado)*

### **Requisitos especiales:**

- Pantalla táctil en panel grande y plano. El texto debe ser visible desde un 50cms.
- Respuesta del ATM en menos de 5 secs, el 90% de las veces.
- Recuperación robusta cuando el acceso mediante comunicaciones falla.
- Posibilidades de internacionalización de texto.
- Comunicaciones cifradas.
- ...

### **Lista de variaciones de tecnología y datos:**

- 3a. Distintos tipos de tarjeta de crédito, dependiendo de los bancos emisores.
- 5a. Se introduce la contraseña mediante un teclado o en la pantalla táctil.
- 5b. En el futuro, creemos que se utilizarán otras técnicas de identificación basadas en biometría.

### **Frecuencia de ocurrencia:**

- Puede ser casi continua.

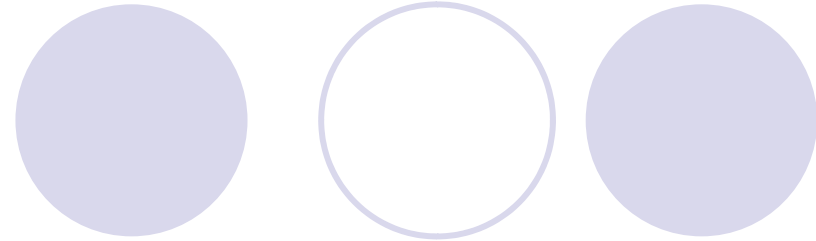
### **Temas abiertos:**

- Explorar el tema de recuperación en caso de fallo de sistemas externos.
- ¿Qué modificaciones se necesitan para idiomas y países distintos?
- ...



# Caso de Uso

## *Retirar Efectivo(Refinado)*



### **Actores primarios:**

Cliente del Banco, Consorcio, Banco

### **Interesados y Objetivos:**

**Cliente del Banco**: quiere retirar dinero de manera rápida, quiere que se anote la transacción en su cuenta de manera correcta, quiere la devolución de su tarjeta y quizá un recibo de la transacción.

**Consorcio**: Quiere identificar correctamente el banco del cliente y mediar en la transacción de manera eficaz.

**Banco**: Quiere identificar correctamente la cuenta del cliente, y anotar la transacción.

### **Precondiciones:**

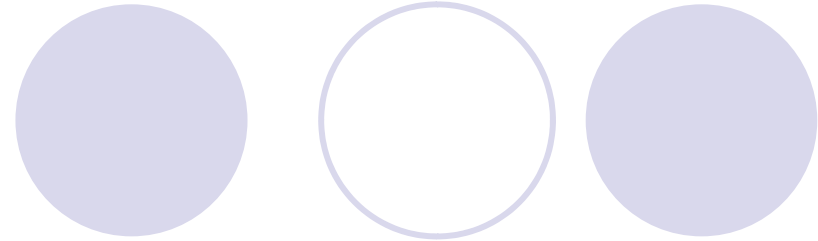
El cliente tiene una cuenta en uno de los bancos del consorcio, ha introducido su tarjeta, y contraseña, y ésta se ha validado correctamente por el banco correspondiente. El cliente selecciona retirar efectivo.

### **Garantía de éxito (post-condiciones):**

El cliente obtiene su dinero, la transacción se anota.

# Caso de Uso

## *Retirar Efectivo(Refinado)*

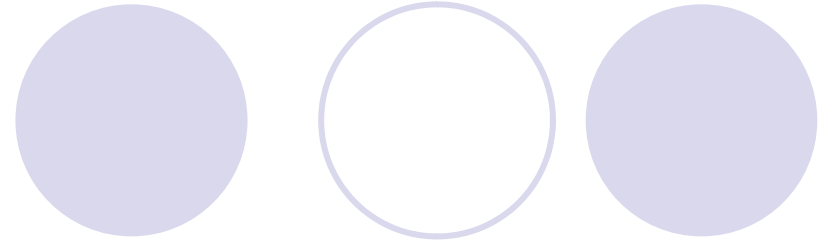


### ***Escenario Principal de Éxito:***

1. El ATM pide al cliente que teclee la cantidad.
2. El cliente teclea una cantidad.
3. El ATM comprueba que la cantidad está dentro de los límites.
4. El ATM genera una transacción y la envía al consorcio.
5. El consorcio pasa la transacción al banco.
6. El banco aprueba la transacción.
7. El banco actualiza la cuenta.
8. El banco envía al consorcio la notificación de aceptación y el nuevo saldo de la cuenta.
9. El consorcio envía al ATM la notificación de aceptación y el saldo.
10. El ATM entrega el dinero al cliente.

# Caso de Uso

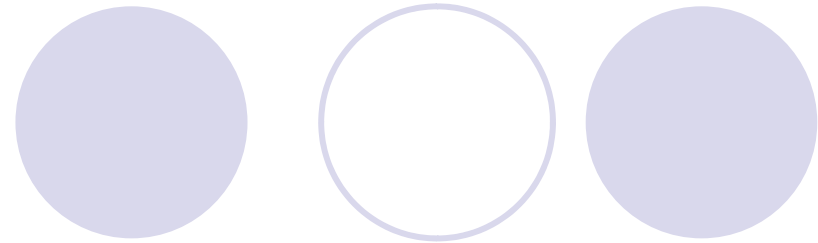
## *Retirar Efectivo(Refinado)*



11. El cliente toma el dinero.
12. El ATM pregunta al cliente si quiere un recibo.
13. El cliente contesta SI.
14. El ATM imprime un recibo y pide al cliente que lo tome.
15. El cliente toma el recibo.
16. El ATM pregunta al cliente si quiere hacer otra operación.
17. El cliente contesta NO.
18. El ATM expulsa la tarjeta de crédito e indica al cliente que la tome.
19. El cliente toma la tarjeta de crédito.
20. El ATM vuelve a la situación inicial.

# Caso de Uso

## *Retirar Efectivo(Refinado)*

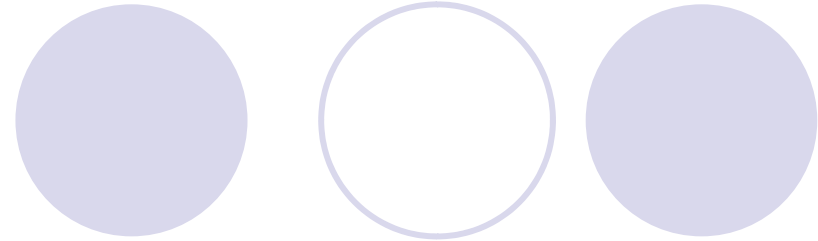


### ***Flujos Alternativos:***

- 2a. El cliente pulsa la tecla CANCELAR.
  - 1. El ATM expulsa la tarjeta de crédito e indica al cliente que la tome.
  - 2. El cliente toma la tarjeta de crédito.
  - 3. El ATM vuelve a la situación inicial.
  
- 3a. La cantidad excede el límite superior o inferior, se vuelve a 1.
  
- 6a. El banco no aprueba la transacción.
  - 1. El banco envía al consorcio la indicación de rechazo.
  - 2. El consorcio envía al ATM la notificación de rechazo.
  - 3. El ATM muestra un mensaje.
  - 4. Se vuelve al caso de uso “Realizar Operación” para que el usuario seleccione un tipo de transacción.

# Caso de Uso

## *Retirar Efectivo(Refinado)*



### ***Flujos Alternativos:***

- 11a. El usuario no toma el dinero después de 30secs.
  - 1. El ATM indica al cliente que tome el dinero y emite una señal sonora.
  - 2. El cliente toma el dinero y el flujo sigue en 11.
- 2a. El cliente no toma el dinero después de 30 secs.
  - 1. El ATM retiene el dinero y la tarjeta.
  - 2. El ATM muestra un mensaje al cliente.
  - 3. El ATM notifica al consorcio de la retención.
  - 4. El consorcio notifica al banco de la retención.
  - 5. El ATM vuelve a la situación inicial.
- 13a. El cliente contesta NO y el flujo continua en 16.
- 16a. El cliente contesta SI y el flujo continua en el paso 1 del caso de uso  
“Realizar Operación”
- ... (timeouts de comunicaciones, rotura de elementos mecánicos del cajero, etc.)

# Caso de Uso

## *Validar Tarjeta y Clave (Refinado)*

### **Requisitos especiales:**

- Pantalla táctil en panel grande y plano. El texto debe ser visible desde un 50cms.
- Respuesta del ATM en menos de 5 secs, el 90% de las veces.
- Recuperación robusta cuando el acceso mediante comunicaciones falla.
- Posibilidades de internacionalización de texto.
- Comunicaciones cifradas.
- ...

### **Lista de variaciones de tecnología y datos:**

2a. Se teclea la cantidad mediante un teclado o en la pantalla táctil.

12a. En lugar de imprimir un recibo se podría mandar un SMS o un e-mail.

### **Frecuencia de ocurrencia:**

- Puede ser casi continua.

### **Temas abiertos:**

- Explorar el tema de recuperación en caso de fallo de sistemas externos.
- ¿Qué modificaciones se necesitan para idiomas y países distintos?
- ...

# Modelo de Objetos

- Identificar objetos y clases
- Identificar y depurar relaciones
- Identificar atributos de objetos y relaciones
- Añadir herencia
- Comprobar los casos de uso (iterar)
- Modularizar
- Añadir y simplificar métodos

# Modelo de Objetos

## *Identificar Objetos y Clases*

- Seleccionar nombres en los requisitos.
- Añadir clases adicionales procedentes de nuestro conocimiento del dominio.
- Eliminar redundancias.
- Eliminar clases irrelevantes.
- Eliminar clases vagas.
- Separar atributos.
- Separar métodos.
- Eliminar objetos de diseño.
- Resultado: Preparar diccionario de clases.



# Modelo de Objetos

## *Seleccionar Nombres en los Requisitos*

Se desea diseñar el software necesario para una red bancaria provista de cajeros automáticos (ATMs), que serán compartidos por un consorcio de bancos. Cada banco dispone de una serie de servidores, provistos de software propio, que llevan la información sobre sus cuentas y procesa las transacciones que actúan sobre dichas cuentas. A estos servidores están conectados las estaciones de cajero, que son propiedad del banco y en las que operan cajeros humanos, que pueden crear cuentas e introducir transacciones sobre ellas.

Los cajeros automáticos aceptan tarjetas de crédito, interaccionan con el usuario, se comunican con un ordenador central para llevar a cabo las transacciones, entregan dinero en efectivo al usuario e imprimen recibos. El sistema llevará el registro de las transacciones efectuadas, cumplirá características aceptables de seguridad y manejará accesos concurrentes a la misma cuenta.

El coste de desarrollo de la parte compartida del sistema se dividirá entre los bancos que forman parte del consorcio en función del número de clientes provistos de tarjetas de crédito.

# Modelo de Objetos

## *Seleccionar Nombres en los Requisitos*

- Software,
- Red bancaria,
- Cajero automático (ATM),
- Consorcio de bancos,
- Banco,
- Servidores,
- Cuenta bancaria,
- Información sobre la cuenta,
- Transacción de cajero,
- Estaciones de cajero,
- Cajero humano,
- Tarjeta de crédito,
- Usuario,
- Ordenador central,
- Transacción Remota,
- Dinero en efectivo,
- Recibo,
- Sistema,
- Registro de transacciones,
- Características de seguridad,
- Acceso a la cuenta,
- Coste de desarrollo,
- Parte compartida,
- Cliente.

# Modelo de Objetos

## Identificar Objetos y Clases

- Añadir clases adicionales procedentes de nuestro conocimiento del dominio.
  - Podemos añadir la clase *Línea de comunicaciones*.
- Eliminar redundancias.
  - *Cliente* y *Usuario* son la misma clase. Nos quedamos con *Cliente* por adaptarse mejor al concepto.
- Eliminar clases irrelevantes.
  - *Coste de desarrollo* no tiene nada que ver con el problema, queda fuera del sistema.
- Eliminar clases vagas.
  - *Sistema*, *Características de seguridad*, *Red bancaria* y *Parte compartida* pueden considerarse vagas.

# Modelo de Objetos

## Identificar Objetos y Clases

- Separar atributos
  - Los atributos definen datos asociados a un objeto, en lugar de objetos (un atributo objeto se representa mediante una relación).
  - En el ejemplo, pueden considerarse atributos *Información sobre la cuenta*, (atributo de *Cuenta bancaria*), *Dinero en efectivo* y *Recibo* (atributos de *Cajero automático*), que pasan a ser clases eliminadas.
- Separar métodos
  - Observación: algunos nombres (por ejemplo, *Llamada telefónica*) definen realmente operaciones o eventos.
- Eliminar objetos de diseño
  - Todas las clases que corresponden más a la solución del problema que a la situación real, deben considerarse objetos de diseño y eliminarse en la fase del análisis.
  - En el ejemplo, eliminaremos *Registro de transacciones*, *Línea de comunicaciones*, *Acceso a la cuenta* y *Software*.

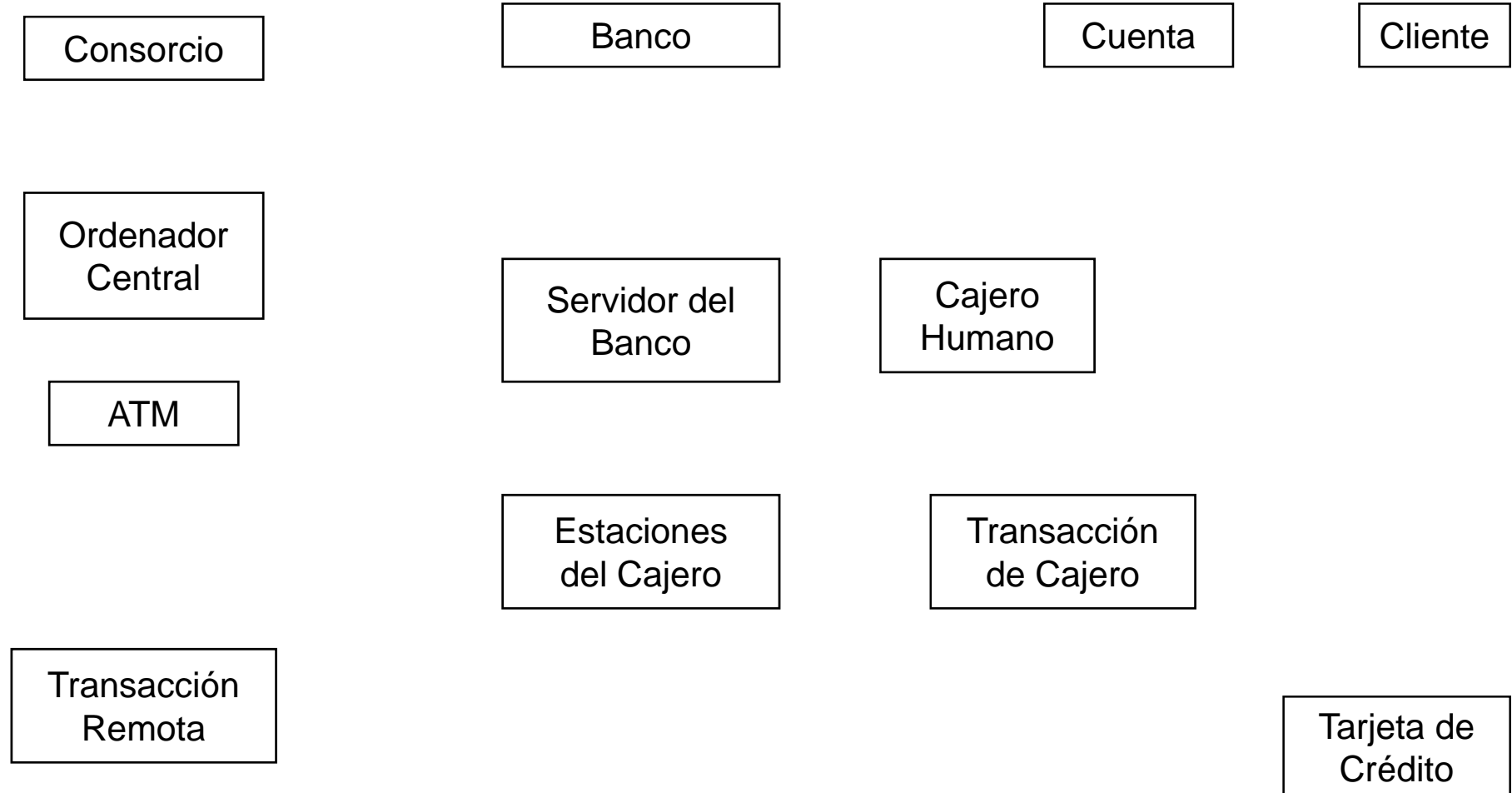
# Modelo de Objetos

## *Identificar Objetos y Clases*

- Cajero automático (ATM),
- Consorcio de bancos,
- Banco,
- Servidores,
- Cuenta bancaria,
- Transacción,
- Estaciones de cajero,
- Cajero humano,
- Tarjeta de crédito,
- Ordenador central,
- Cliente.

# Modelo de Objetos

*Identificar Objetos y Clases*



# Modelo de Objetos

## *Diccionario de Clases*

- El diccionario de clases contiene la definición detallada de todas las clases en lenguaje natural. Ejemplo:
  - *Cajero automático (ATM)*: Terminal remoto que permite a los clientes realizar transacciones utilizando tarjetas de crédito para identificarse. El ATM interacciona con el cliente para identificar la transacción deseada y sus datos asociados, envía esta información al ordenador central para su validación y proceso, y entrega al usuario dinero en efectivo y un recibo. Suponemos que el ATM no opera cuando está desconectado de la red.
  - *Consortio de bancos*: Conjunto organizado de bancos que lleva la gestión de los cajeros automáticos. Suponemos que sólo se gestionan transacciones para los bancos que pertenecen al consorcio.
  - *Banco*: Institución financiera que maneja las cuentas bancarias de sus clientes y emite tarjetas de crédito que facilitan el acceso a dichas cuentas a través de la red de cajeros automáticos.

# Modelo de Objetos

*Identificar y depurar relaciones*

- Seleccionar verbos relacionales en los requisitos.
- Añadir relaciones adicionales procedentes de nuestro conocimiento del dominio.
- Eliminar relaciones de diseño o entre clases eliminadas.
- Eliminar eventos transitorios.
- Reducir relaciones ternarias.
- Eliminar relaciones redundantes o derivadas.
- Añadir relaciones olvidadas.
- Definir la multiplicidad de cada relación.



# Identificar y Depurar Relaciones

## Seleccionar verbos relacionales en los requisitos

1. Una *Red bancaria* está provista de *Cajeros automáticos*.
2. El *Consorcio de bancos* comparte los *Cajeros automáticos*.
3. Cada *Banco* dispone de un *Servidor*.
4. El *Servidor* dispone de *Software*.
5. Cada *Servidor* lleva la información sobre las *Cuentas bancarias*.
6. Cada *Servidor* procesa *Transacciones*.
7. Una *Transacción* actúa sobre una *Cuenta bancaria*.
8. Las *Estaciones de cajero* están conectadas al *Servidor*.
9. Las *Estaciones de cajero* son propiedad del *Banco*.
10. El *Cajero humano* opera en la *Estación de cajero*.
11. El *Cajero humano* crea *Cuentas bancarias*.
12. El *Cajero humano* introduce *Transacciones* sobre las *Cuentas bancarias*.
13. Los *Cajeros automáticos* aceptan *Tarjetas de crédito*.
14. Los *Cajeros automáticos* interaccionan con el *Usuario*.

# Identificar y Depurar Relaciones

## *Seleccionar verbos relacionales en los requisitos*

15. Los *Cajeros automáticos* comunican con el *Ordenador central*.
16. El *Ordenador central* lleva a cabo las *Transacciones*.
17. Los *Cajeros automáticos* entregan *Dinero en efectivo* al *Usuario*.
18. Los *Cajeros automáticos* imprimen *Recibos*.
19. El *Sistema* lleva el *Registro de las transacciones*.
20. El *Sistema* cumple *Características de seguridad*.
21. El *Sistema* maneja *Accesos concurrentes* a la *Cuenta bancaria*.
22. El *Coste de desarrollo* se divide entre los *Bancos*.
23. Los *Bancos* forman parte del *Consortio*.
24. Los *Clientes* están provistos de *Tarjetas de crédito*.

Relaciones adicionales implícitas en el texto

25. Las *Cuentas bancarias* están en los *Bancos*.
26. El *Ordenador central* pertenece al *Consortio*.
27. Los *Bancos* tienen *Clientes*.

# Identificar y Depurar Relaciones

- Añadir relaciones adicionales procedentes de nuestro conocimiento del tema:
  - 28. Las *Tarjetas de crédito* están asociadas a las *Cuentas bancarias*.
  - 29. Los *Cajeros humanos* son empleados de los *Bancos*.
- Eliminar relaciones de diseño o entre clases eliminadas:
  - Las de diseño se dejan para la fase de diseño. Eliminamos las relaciones números 1, 4, 17, 18, 19, 20, 21, 22.
- Eliminar eventos transitorios:
  - Son sucesos que pertenecen al modelo dinámico y no constituyen relaciones estructurales (estáticas) entre los objetos. Tras ejecutarse estas operaciones no se modifica la estructura de los objetos involucrados.
  - Eliminamos las relaciones números 13 y 14.
  - Otras veces conviene reformularlas, como en el caso de la número 16, el *Ordenador central* lleva a cabo las *Transacciones*, que debería sustituirse por: 16a. El *Ordenador central* se comunica con el *Banco*.

# Identificar y Depurar Relaciones

## *Reducir Relaciones Ternarias*

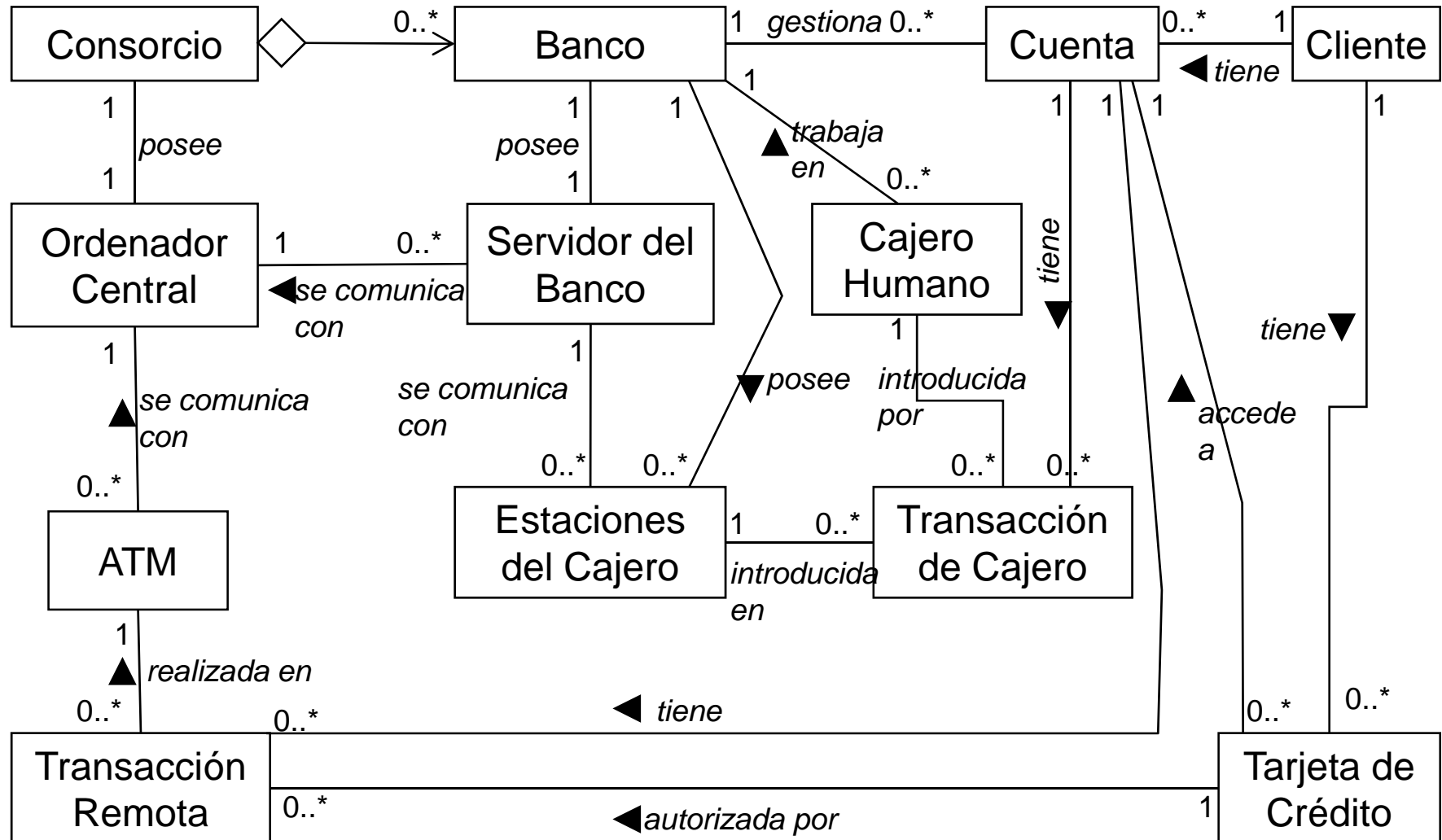
- Son relaciones entre tres o más clases.
- Muchas veces es posible descomponerlas en varias relaciones binarias (entre dos clases); si no es posible, sí que se pueden utilizar atributos de relación.
- Por ejemplo, la relación número 12 (El *Cajero humano* introduce *Transacciones* sobre las *Cuentas bancarias*) puede descomponerse en:
  - 12a. El *Cajero humano* introduce *Transacciones*
  - 12b. Las *Transacciones* actúan sobre las *Cuentas bancarias*.
- De igual modo, la número 17 puede descomponerse así:
  - 17a. Los *Cajeros automáticos* entregan *Dinero en efectivo*.
  - 17b. El *Usuario* recoge el *Dinero en efectivo*.

# Identificar y Depurar Relaciones

- Eliminar relaciones redundantes o derivadas
  - Por ejemplo, la relación número 2 es una combinación de las relaciones número 15 y 26. Hay que tener cuidado, sin embargo, de no eliminar relaciones aparentemente redundantes, pero que en realidad son necesarias. Las redundantes por ejemplo son las que se derivan de la propiedad transitiva para relaciones.
- Añadir relaciones olvidadas. Por ejemplo:
  - 30. Los *Clientes* tienen *Cuentas*.
  - 31. Las *Transacciones* son autorizadas por la *Tarjeta de crédito*.
  - 32. Las *Transacciones* pueden introducirse en una *Estación de cajero*.
- Definir la multiplicidad de cada asociación
  - Un *Banco* puede contener muchas *Cuentas*.
  - Un *Cliente* puede tener muchas *Cuentas*.
  - Un *Cliente* puede tener muchas *Tarjetas de crédito*.
  - Un *Banco* emplea muchos *Cajeros*.
  - Un *Banco* tiene un solo *Ordenador del banco*.
  - El *Ordenador central* se comunica con muchos *Ordenadores del banco*.
  - ....

# Modelo de Objetos

## Diagrama de Clases inicial





# Identificar Atributos de Objetos y Relaciones

- Distinguir los objetos de los atributos
- Distinguir entre los atributos de objetos y de relaciones
- Eliminar atributos privados (de diseño)
- Eliminar atributos de detalle fino
- Localizar atributos discordantes (muy diferentes de los demás; puede convenir dividir la clase en dos)

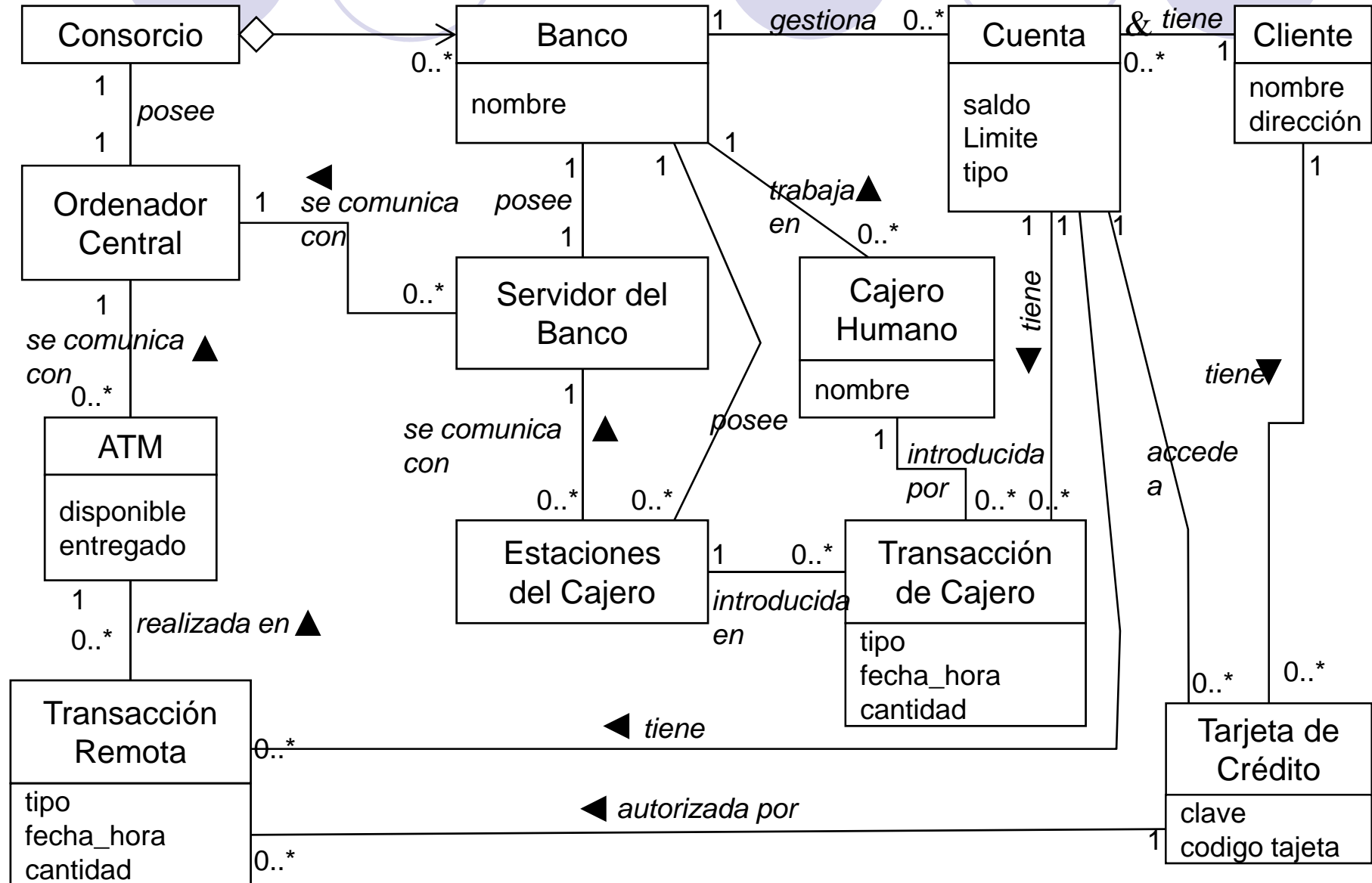
# Identificar Atributos de Objetos y Relaciones

- Atributos de los objetos
  - Del *Banco*: Nombre.
  - De la *Cuenta*: Saldo, Límite de crédito, Tipo de cuenta.
  - Del *Cliente*: Nombre, Dirección.
  - Del *Cajero*: Nombre.
  - De una *Transacción del cajero*: Tipo, Fecha y hora, Cantidad.
  - Del *Cajero automático*: Efectivo disponible, Cantidad entregada.
  - De una *Transacción remota*: Tipo, Fecha y hora, Cantidad.
  - De la *Tarjeta de crédito*: Clave, Código de la tarjeta.
- Atributos de las relaciones (la multiplicidad de la relación queda sobreentendida al usar un "código")
  - 8 y 9: Código de la estación de cajero.
  - 15: Código del cajero automático.
  - 16a: Código del banco.
  - 23: Código del banco.
  - 25: Código de la cuenta.
  - 29: Código de empleado.



# Modelo de Objetos

## Diagrama de Clases, atributos

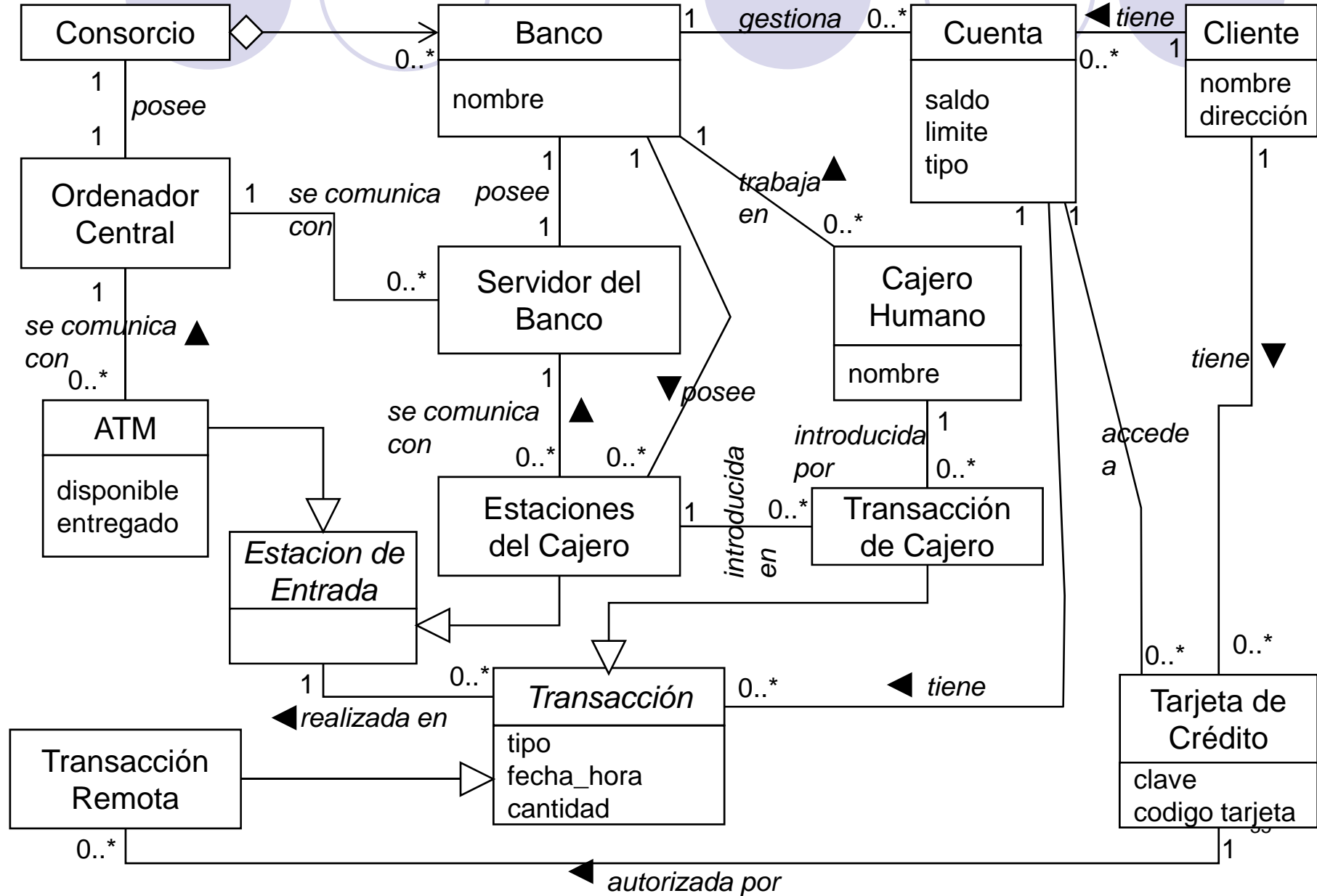


# Añadir Herencia

- Introducimos clases nuevas (quizá abstractas) que contienen información común a dos o más clases preexistentes.
- Procurar evitar la herencia múltiple, a menos que sea estrictamente necesaria.
- Resultado: Primer diagrama de clases
- En el ejemplo:
  - La clase *Estación de entrada* será superclase de *Cajero automático* y de *Estación de cajero*.
  - La clase *Transacción* será superclase de *Transacción de cajero* y de *Transacción remota*.
  - Podrían refinarse los tipos de cuentas

# Modelo de Objetos

## Diagrama de Clases, herencia



# Comprobar los Casos de Uso (iterar)

Para localizar fallos que deben corregirse fijarse en:

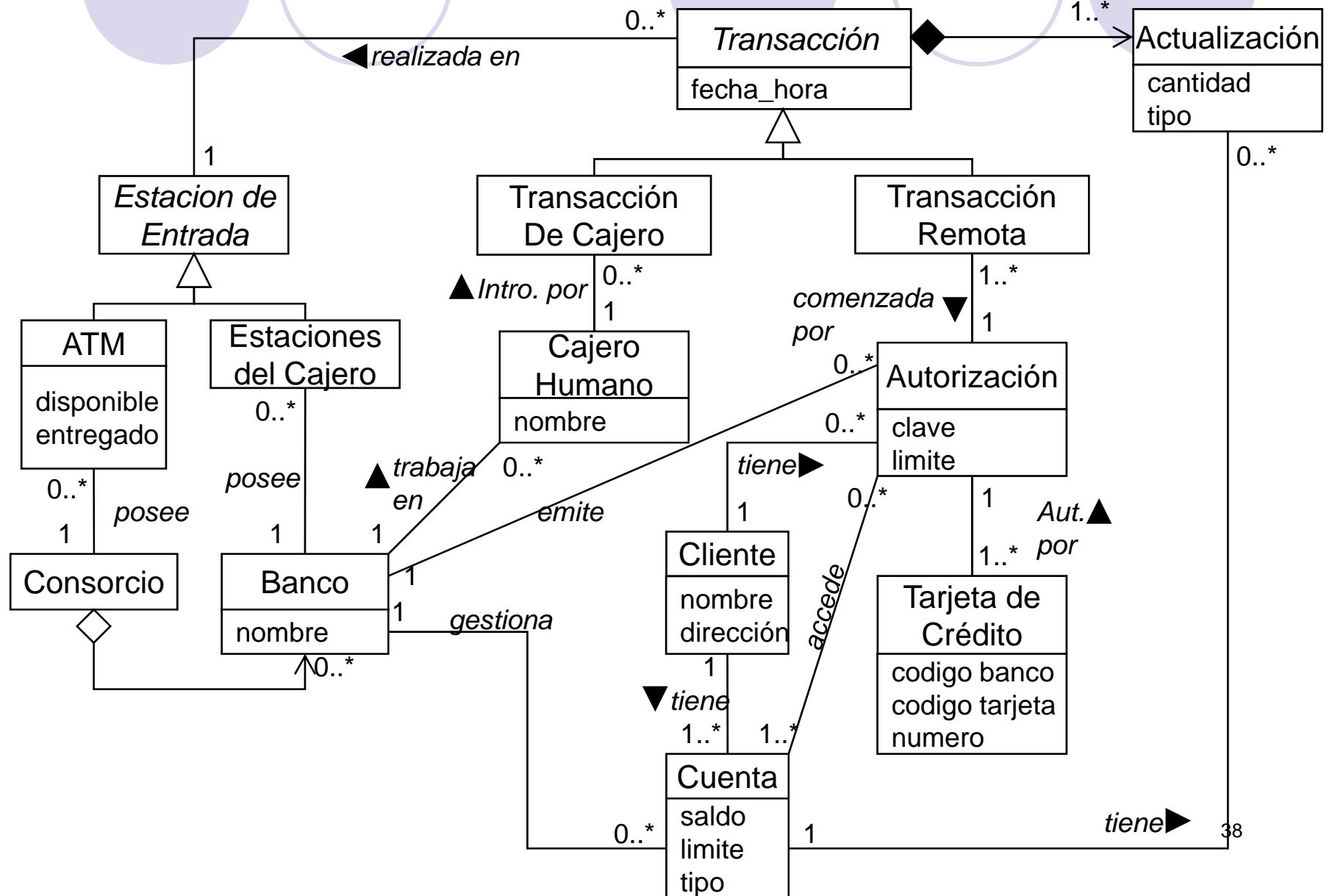
- Atributos muy dispares (discordantes): descomponer una clase en dos.
- Operaciones sin objetivo: añadir clase con estas operaciones como métodos de clase.
- Conversión de relaciones en clases: por ejemplo, clase *Empleado* (clase asociación para una relación entre las clases *Persona* y *Compañía*, que representa la forma en que una compañía contrata a una persona)
- Operaciones que no encuentran camino para realizarse: añadir relaciones.
- Relaciones redundantes: eliminarlas.
- Relaciones demasiado detalladas o demasiado vagas: subirlas a una superclase o bajarlas a una subclase.
- Clases sin atributos, sin métodos o sin relaciones: eliminarlas.
- Relaciones que nadie atraviesa: eliminarlas.
- Atributos de clase necesarios en un acceso: pasarlos a atributos de relación (por ejemplo el código).

# Comprobar los Casos de Uso (iterar)

- En el ejemplo de los cajeros automáticos:
- *Tarjeta de crédito* desempeña dos roles: la tarjeta física, que se introduce y que permite al cajero automático conectarse con el banco, con información sobre el mundo real (banco, número de la tarjeta) y las autorizaciones concedidas por éste, que sólo son números en la memoria de un ordenador y se pueden cambiar con facilidad (contraseña, límite de crédito). Se puede descomponer en *Tarjeta de crédito* y *Autorización* de la tarjeta. Una sola autorización puede afectar a más de una tarjeta física. Una misma autorización puede permitir acceder a más de una cuenta (y viceversa).
- Introducimos la clase *Actualización de cuenta* para refinar el concepto de *Transacción*. Una misma transacción puede estar compuesta de varias actualizaciones de cuenta (por ejemplo, transferencia entre cuentas son dos actualizaciones).
- No hay distinción significativa entre *Banco* y *Ordenador del banco*, por una parte, y entre *Consortio* y *Ordenador central*, por otra. Fusionamos esas clases.

# Modelo de Objetos

## Diagrama de Clases, Iteración

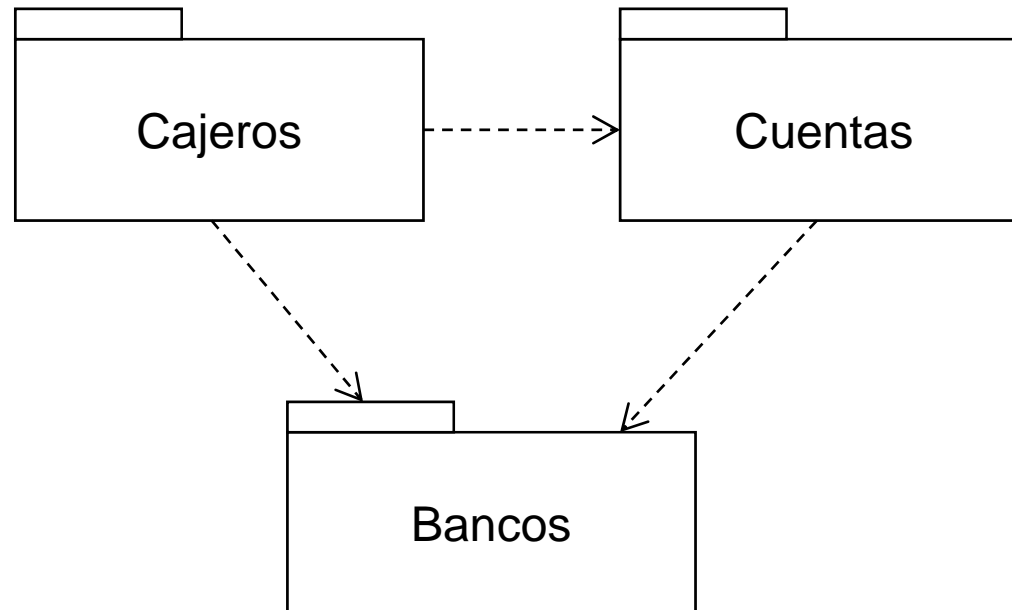


# Modularizar



- Agrupar clases en módulos.
- En el ejemplo de los cajeros automáticos.  
Posibles módulos:
  - Cajeros en general: *Cajero, Estación de cajero, ATM, Estación de entrada.*
  - Cuentas en general: *Cuenta, Tarjeta de crédito, Autorización, Cliente, Transacción, Transacción de cajero, Transacción remota.*
  - Bancos: *Banco, Consorcio.*
- Resultado: Diagrama de Paquetes

# Diagrama de Paquetes





# Modelo Dinámico



Consta de los siguientes pasos:

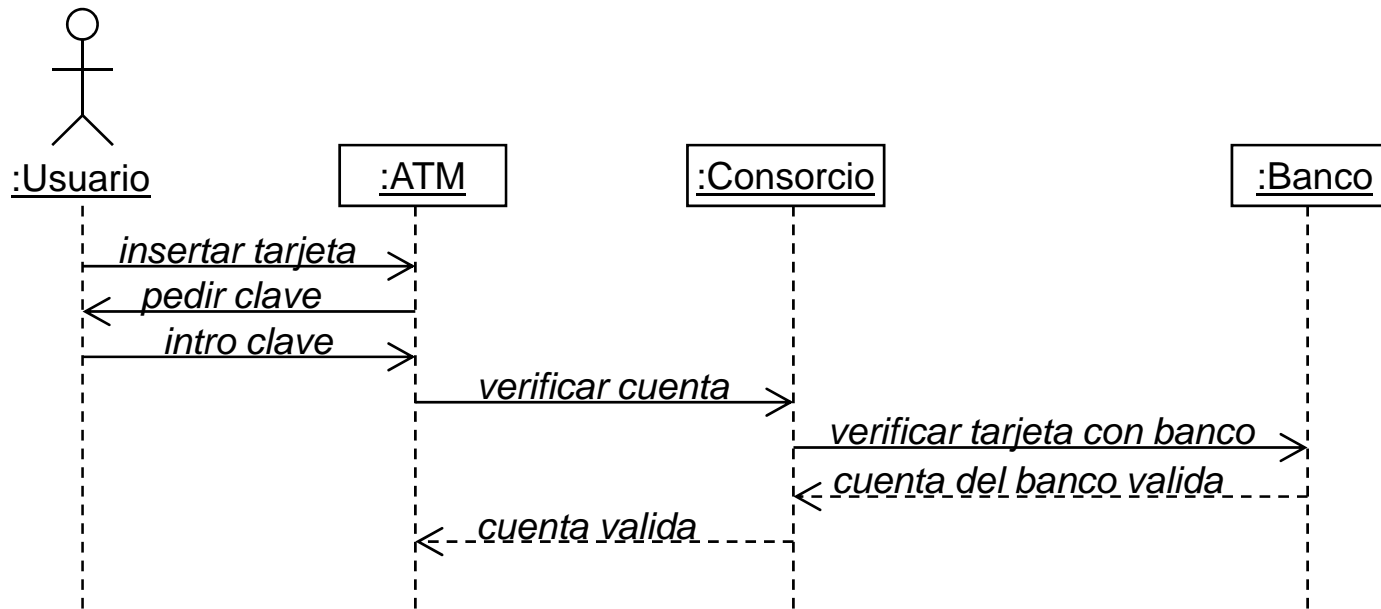
- Identificar sucesos
- Construir diagramas de estados
- Comprobar consistencia (iterar)
- Añadir métodos

# Identificar Mensajes

- Los mensajes se extraen de los casos de uso (escenarios). Pueden ser de los siguientes tipos:
  - Señales
  - Entradas
  - Decisiones
  - Interrupciones
  - Transiciones
  - Acciones externas
  - Condiciones de error
- Resultados: Diagramas de secuencia y de colaboración.

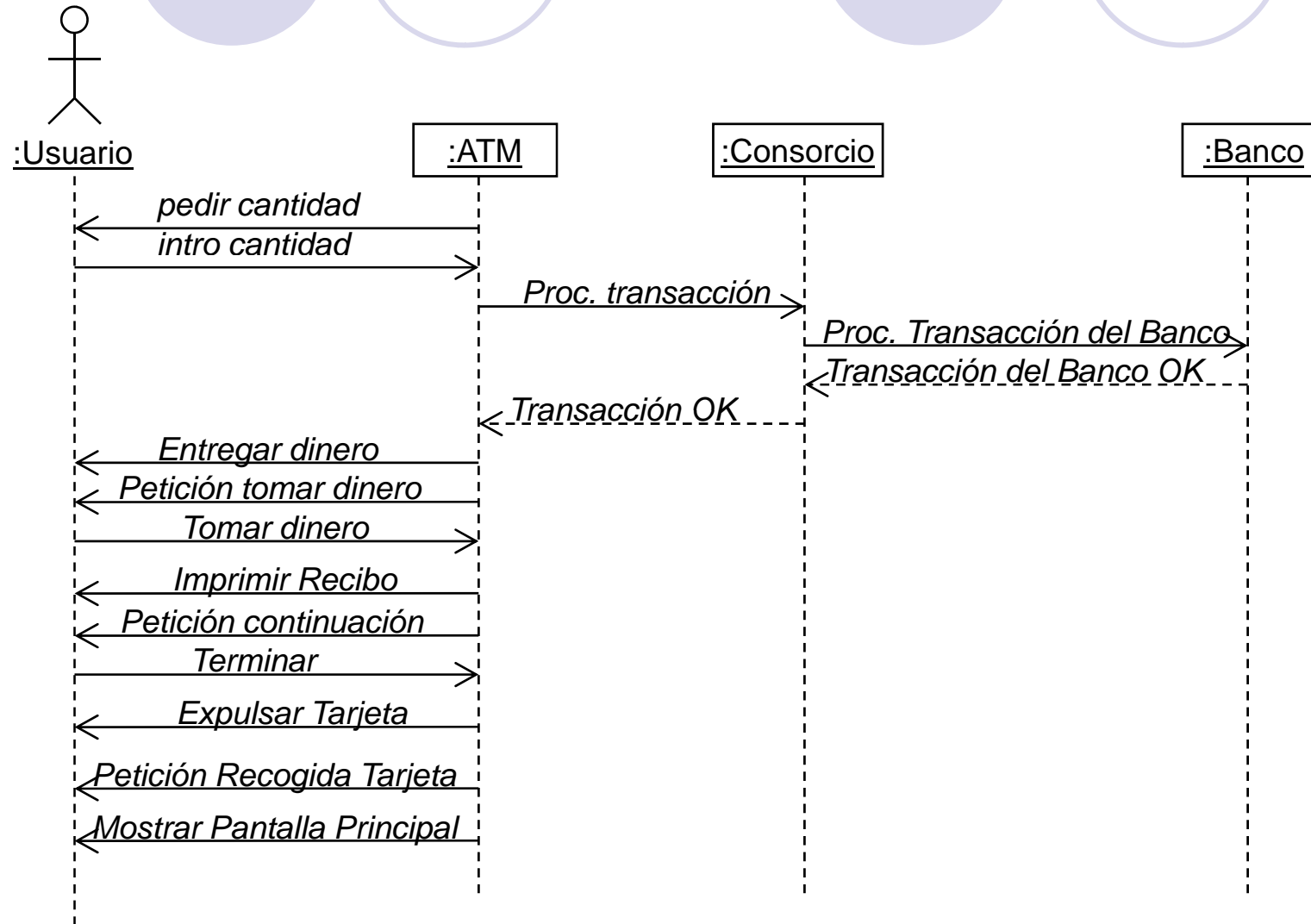
# Diagrama de Secuencia

## Validar Tarjeta y Clave



# Diagrama de Secuencia

## Retirar Efectivo



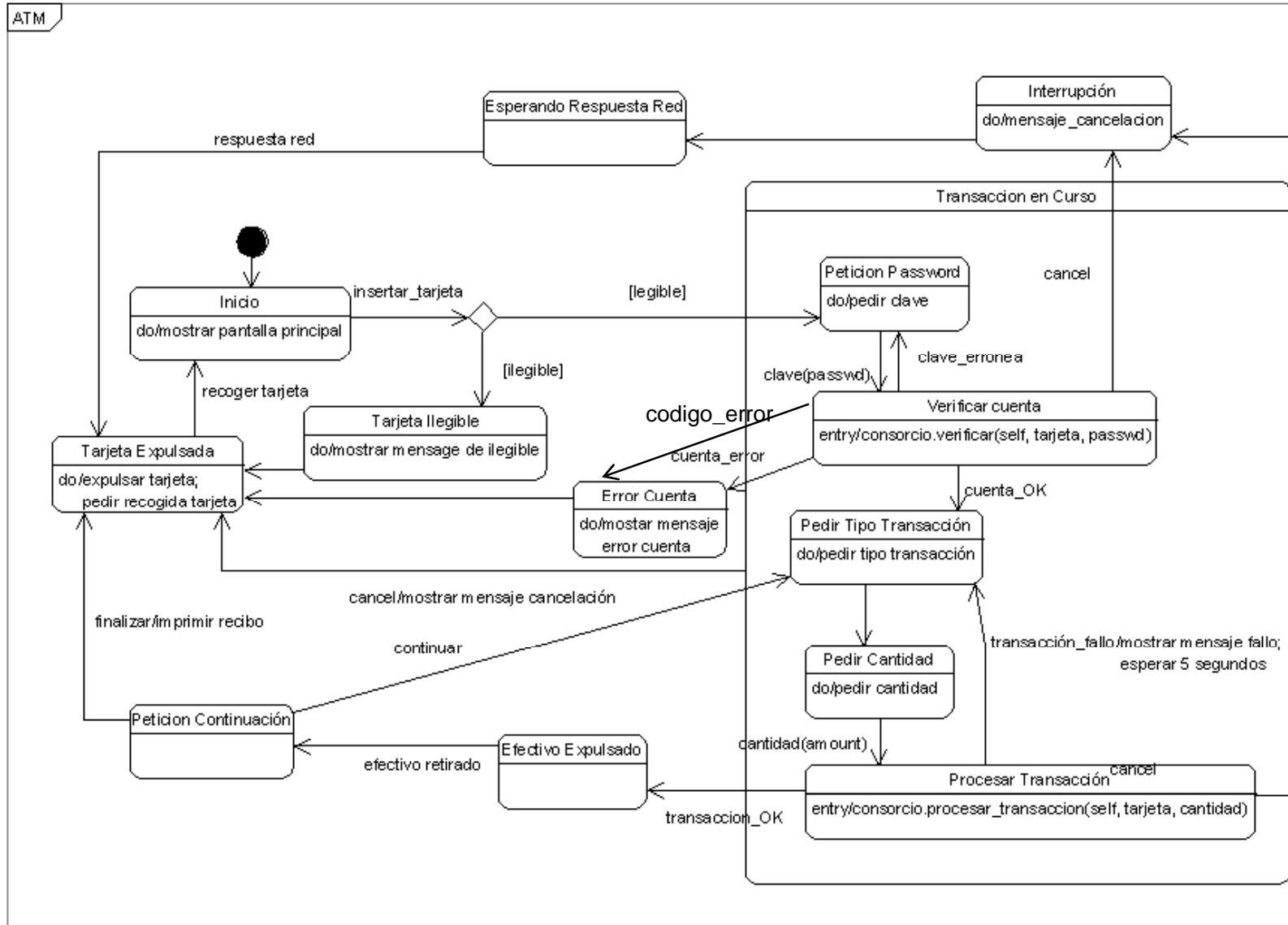
# Identificar Mensajes

- Los casos de uso (escenarios) se convierten en diagramas de secuencia. Estas se compactan en diagramas de colaboración.
- En el ejemplo de los cajeros automáticos:
  - *El cliente introduce la contraseña* define un mensaje de entrada que el objeto *Cliente* envía al objeto *Cajero automático*. *El cajero automático entrega el dinero al cliente* es un evento que el objeto *Cajero automático* envía al objeto *Cliente*.
- Agrupar los mensajes equivalentes:
  - *El cliente introduce la contraseña* es el mismo evento independientemente de la contraseña introducida. *El cajero automático entrega el dinero al cliente* es el mismo mensaje independientemente de la cantidad entregada.
- No agrupar los mensajes no equivalentes: *El banco autoriza la transacción* es distinto evento que *El banco rechaza la transacción*.

# Construir Diagramas de Estado

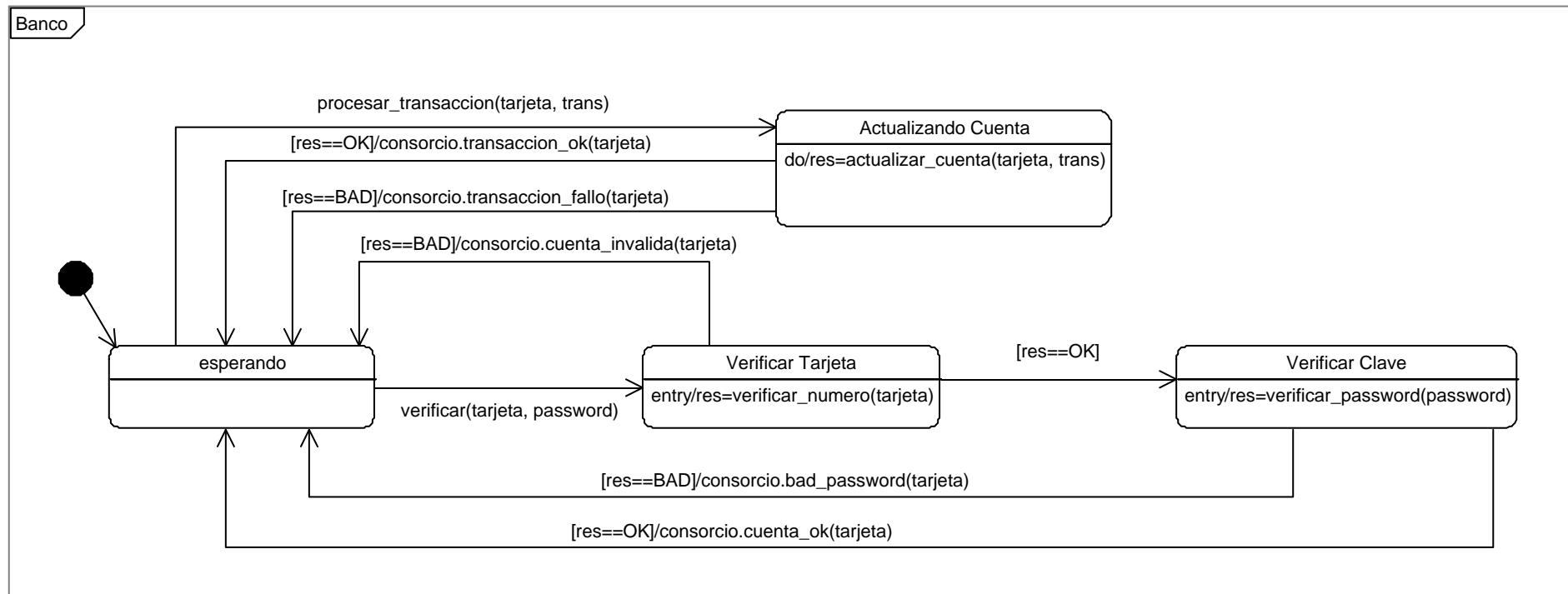
- Uno por clase. Determinar los eventos que provocan transiciones entre estados.
- En el ejemplo de los cajeros automáticos centrarse en las clases dinámicas, que cambian de estado:
  - Cajero automático
  - Banco
  - Consorcio
  - Estación de cajero
- No hace falta construir diagramas de estado de las clases pasivas, que no cambian de estado de modo significativo:
  - Tarjeta de crédito
  - Transacción
  - Cuenta
- Tampoco hace falta considerar a fondo los objetos externos, que no forman parte del sistema informático:
  - Cliente
  - Cajero humano

## Diagrama de Transición Estados, clase ATM



# Modelo de Objetos

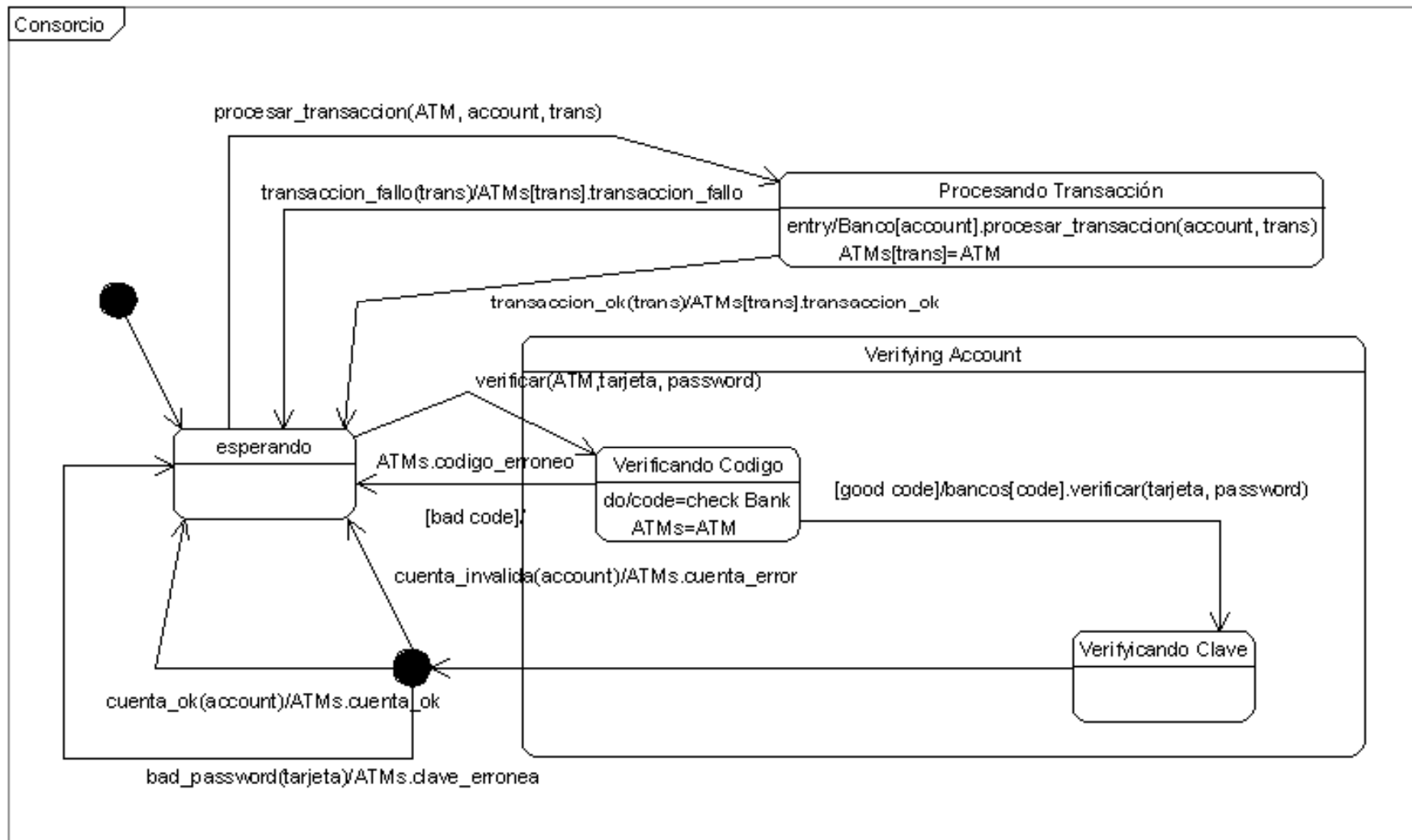
## Diagrama de Transición Estados, clase Banco





# Modelo de Objetos

## Diagrama de Transición Estados, clase Consorcio



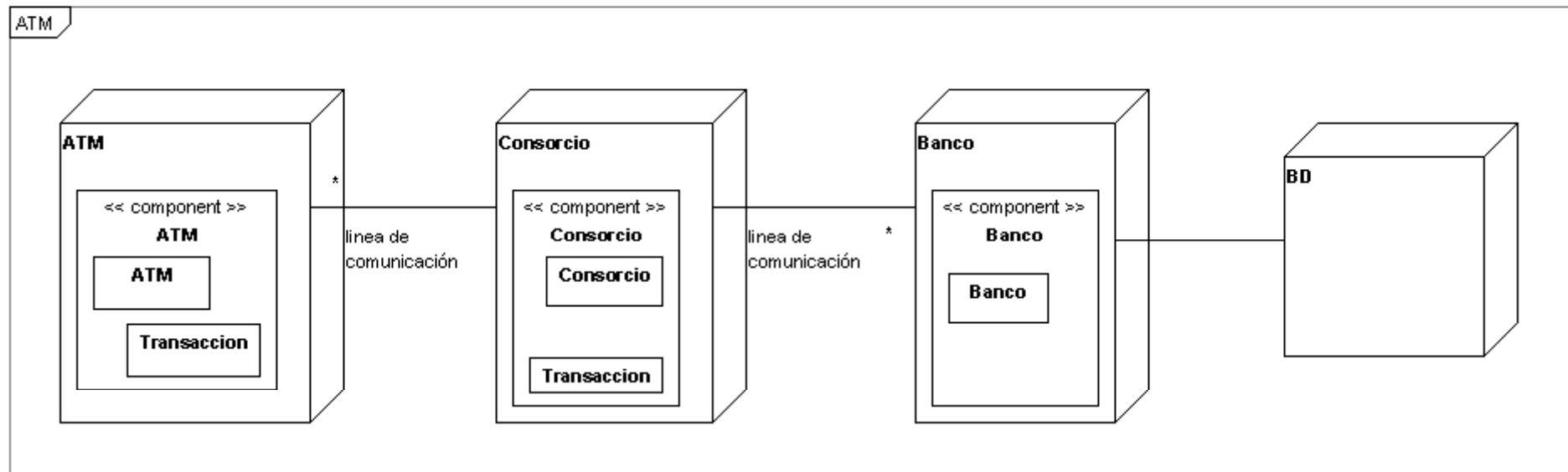
# Ejercicio



- ¿Son consistentes los diagramas anteriores entre sí?
- ¿Son consistentes con los casos de uso?
- Añadir la información de los casos alternativos y excepciones (timeouts, etc.)

# Arquitectura

## Diagrama de Despliegue



# Sistema de Control de Tráfico Ferroviario (SCTF)

- Sistema para el control de tráfico ferroviario (de pasajeros y carga), que permita incrementar el tráfico de trenes, así como la programación predecible de horarios.
- Automatización del enrutado de trenes y monitorización de todos los elementos del sistema del tren.
- Objetivos: Reducir costes de operación y mejorar el uso de recursos.

# Sistema de Control de Tráfico Ferroviario

## Requisitos

- Problema: requisitos poco claros y contradictorios.
- Se hace necesario un modelo de desarrollo iterativo e incremental. Metodología RUP.
- Sistema complejo, varios años de desarrollo: permitir cierto grado de cambio en los requisitos, para aprovechar avances en el hardware.
- Riesgo de “*parálisis en el análisis*”, dado que el número de requisitos es muy grande.

# Sistema de Control de Tráfico Ferroviario

*Requisitos: Comienzo (“Inception”)*

- Dos funciones principales: enrutado de trenes y monitorización.
- Otras funciones relacionadas:
  - Planificación del tráfico.
  - Predicción de fallos.
  - Seguimiento de la posición de los trenes.
  - Evitar colisiones.
  - Registro de mantenimiento.

# Sistema de Control de Tráfico Ferroviario

## Casos de Uso

- **Enrutar Tren**: Establecer un plan para un tren, que define el recorrido de un tren particular
- **Planificar Tráfico**: Establecer un plan de tráfico que provea una guía en el desarrollo de rutas para trenes en un periodo de tiempo para una región geográfica.
- **Controlar los Sistemas del Tren**: Controlar los sistemas de a bordo del tren para verificar que funcionan correctamente.
- **Predicción de Fallos**: Realizar un análisis del estado de los sistemas del tren para predecir la probabilidad de fallo relativa al plan del tren.
- **Seguimiento de Trenes**: Seguir la posición de los trenes usando los recursos del SCTF, así como GPS.
- **Seguimiento del tráfico**: Monitorización del tráfico de trenes en una región geográfica.
- **Evitar colisiones**: Proporcionar los medios, automáticos y manuales para evitar colisiones de trenes.
- **Registro de Mantenimiento**: Proporcionar los medios para anotar el mantenimiento realizado en los trenes.

# Sistema de Control de Tráfico Ferroviario

## *Requisitos no Funcionales y Restricciones*

- Requisitos no Funcionales:
  - Transporte de manera segura de pasajeros y cargamento.
  - Soporte de velocidades de tren de hasta 250 millas por hora (400 km/hora).
  - Interoperar con sistemas de gestión de tráfico en las fronteras del SCTF.
  - Asegurar la máxima reutilización y compatibilidad con el equipamiento existente.
  - Proporcionar una disponibilidad del sistema al nivel del 99.99%.
  - Proporcionar redundancia funcional completa para las capacidades del SCTF.
  - Proporcionar precisión en la posición del tren de 10 yardas (9 metros).
  - Proporcionar precisión en la velocidad del tren de 1.5 millas/hora (2.5 km/hora).
  - Respuesta a las órdenes del operador en menos de 1.0 segundos.
  - Facilidad de mantenimiento y evolución del SCTF.
- Restricciones:
  - Seguimiento de los estándares nacionales, gubernamentales e industriales.
  - Maximizar el uso de componentes COTS (commercial-off-the-shelf) hardware y software.



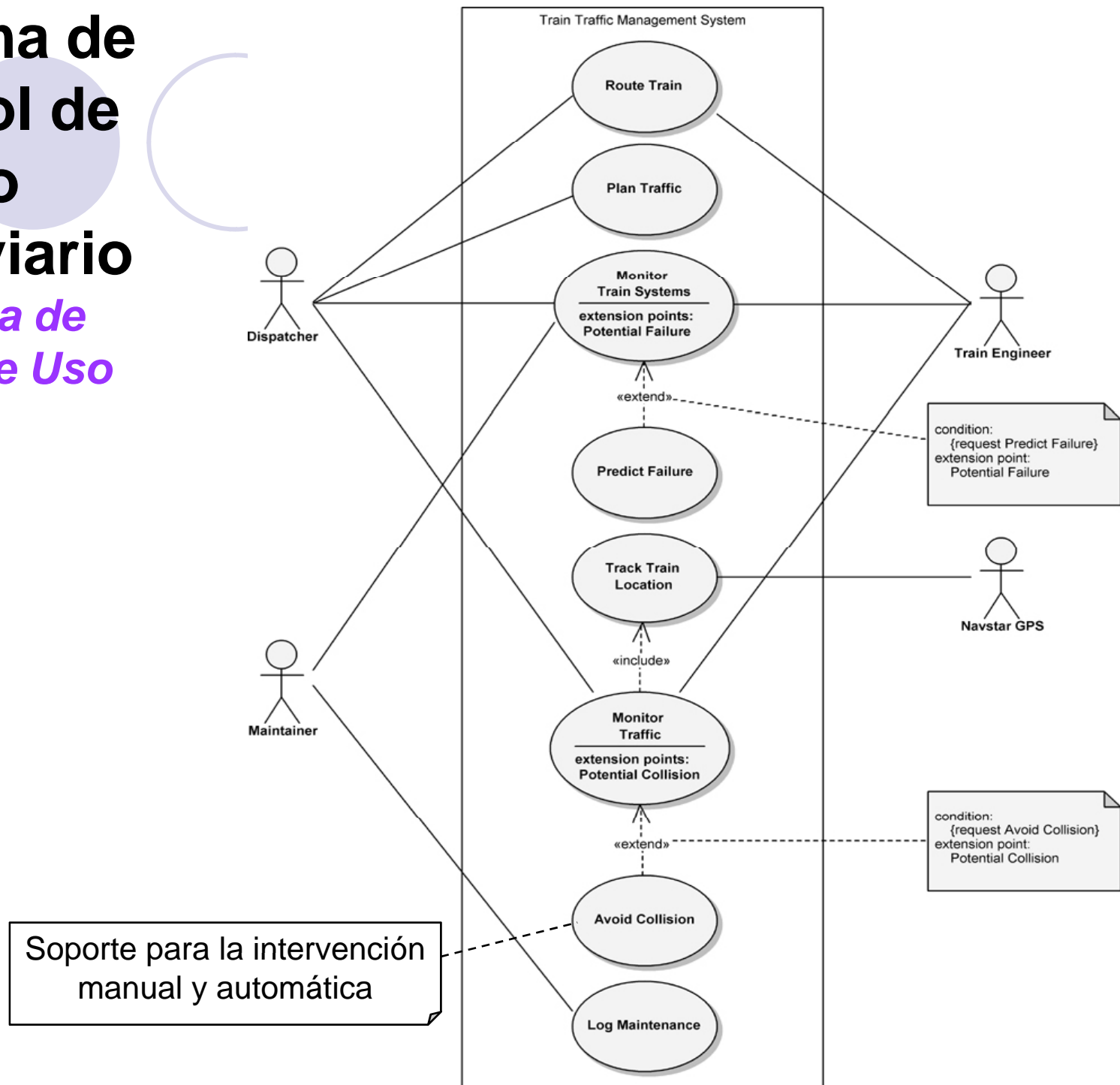
# Sistema de Control de Tráfico Ferroviario

## Actores

- **Controlador (Dispatcher)**: Establece las rutas de los trenes y sigue el progreso de los trenes individuales.
- **Maquinista (Train Engineer)**: Monitoriza el estado del tren y opera el mismo.
- **Operario de Mantenimiento (Maintainer)**: Monitoriza el estado y mantiene los sistemas del tren.
- **GPS Navstar**: Proporciona los servicios de localización para el seguimiento de los trenes.

# Sistema de Control de Tráfico Ferroviario

## Diagrama de Casos de Uso



# Caso de Uso: Enrutar Tren

**Propósito:** Establecer un plan para el tren que actúe como repositorio para toda la información asociada con la ruta de un tren específico y las acciones que sucedan en el camino.

**Contacto:** Pedro Pérez

**Fecha de modificación:** 9/5/06

**Pre-condiciones:** Existe un plan de tráfico para el intervalo de tiempo y la región geográfica relevante al plan que se está elaborando.

**Post-condiciones:** Se estableció el plan para el tren, que detalla su ruta de viaje.

**Limitaciones:** Cada tren tiene un ID único en el sistema. Los distintos recursos pueden no ser usables por más de un plan de tren en un cierto intervalo de tiempo.

**Suposiciones:** Un plan de tren es accesible por los controladores para su consulta y modificación y accesible a los ingenieros ferroviarios para consulta.

## **Escenario Principal:**

1. El SGTF presenta al controlador una lista de opciones.
2. El controlador selecciona desarrollar un nuevo plan para un tren.
3. El SGTF presenta una plantilla para un plan de tren al controlador.
4. El controlador completa la plantilla, dando información sobre el ID de la locomotora, los ingenieros ferroviarios y puntos de paso con tiempos.
5. El controlador introduce el plan completado en el SGTF.
6. El SGTF asigna un ID único al plan de tren y lo almacena. El SGTF hace accesible el plan para consulta y modificación.

# Caso de Uso: Enrutar Tren

## *Escenarios Alternativos*

### **Desarrollo de un plan basado en uno existente:**

- 2a. El controlador selecciona desarrollar un nuevo plan de tren, basado en uno existente.
3. El controlador proporciona unos criterios de búsqueda para encontrar planes existentes.
4. El SGTF proporciona los resultados de la búsqueda.
5. El controlador selecciona un plan.
6. El controlador completa un plan.
7. Se sigue en el escenario principal en el paso 5.

### **Modificación de un plan existente**

- 2b. El controlador selecciona modificar un plan existente.
3. El controlador proporciona unos criterios de búsqueda para encontrar planes existentes.
4. El SGTF proporciona los resultados de la búsqueda.
5. El controlador selecciona un plan.
6. El controlador modifica el plan.
7. El controlador introduce el plan modificado en el SGTF.
8. El SGTF almacena el plan modificado y lo hace accesible para consulta y modificación.

# Caso de Uso: Controlar los Sistemas del Tren

**Propósito:** Controlar los dispositivos a bordo del tren para asegurar su funcionamiento correcto.

**Contacto:** Pedro Pérez

**Fecha de modificación:** 10/5/06

**Precondiciones:** La locomotora está funcionando.

**Postcondiciones:** El sistema muestra información sobre el funcionamiento de los sistemas a bordo del tren.

**Limitaciones:** Ninguna identificada.

**Suposiciones:** La visualización del estado de los sistemas se proporciona cuando la locomotora está funcionando. El sistema proporciona señales audibles y visibles (resalta en amarillo los sistemas problemáticos) sobre los problemas del sistema.

## **Escenario Principal:**

1. El SCTF presenta al maquinista una serie de opciones.
2. El maquinista elige controlar los sistemas del tren.
3. El SCTF presenta al maquinista información de estado de los sistemas de tren.
4. El maquinista revisa la información de estado.

# Caso de Uso: Controlar los Sistemas del Tren

## *Escenarios Alternativos*

### **Pedir visualización detallada del sistema**

5. El maquinista elige visualizar de manera detallada un sistema que muestra su estado en amarillo.
6. El SCTF presenta al maquinista información detallada del estado del sistema seleccionado.
7. El maquinista revisa la información detallada proporcionada.
8. Se sigue en el paso 2 del escenario principal

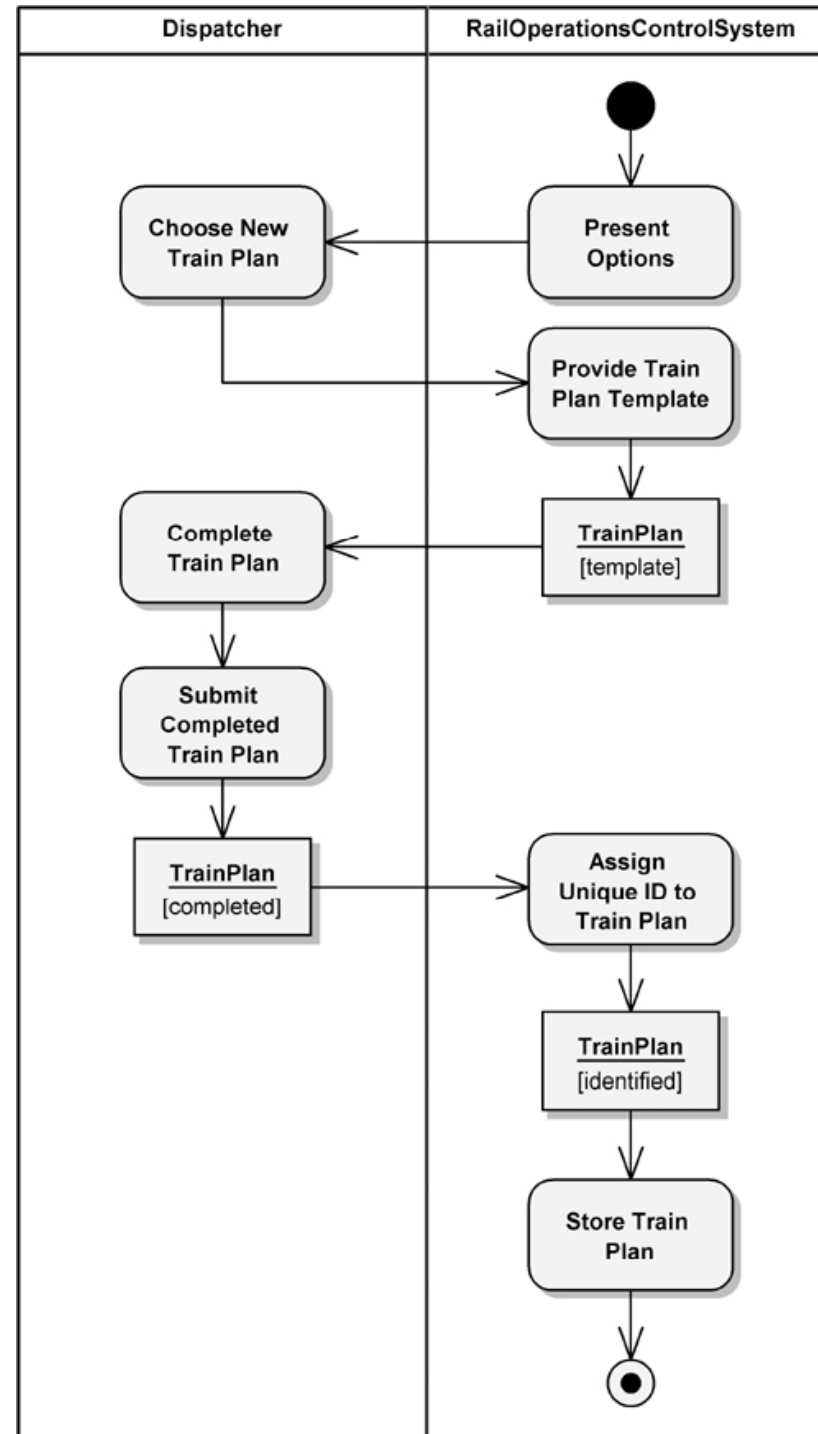
### **Extensión: Solicitar un análisis de predicción de fallos para un sistema.**

- 7a. El maquinista solicita un análisis de predicción de fallos para el sistema.
8. El SCTF realiza un análisis de predicción de fallos para el sistema.
9. El SCTF presenta al maquinista el resultado del análisis.
10. El maquinista revisa el análisis.
11. El maquinista pide al SCTF que alerte al mantenedor del sistema que puede fallar.
12. El SCTF avisa al mantenedor del sistema.
13. El mantenedor solicita revisar los resultados del análisis.
14. El SCTF le presenta la información del análisis de la predicción.
15. El mantenedor revisa el análisis y determina que la condición de color amarillo no es lo suficientemente grave como para requerir acción inmediata.
16. El mantenedor solicita al SCTF que alerte al maquinista de esta decisión.
17. El SCTF muestra la decisión al maquinista.
18. El maquinista elige realizar una visualización del sistema seleccionado.
19. El escenario alternativo "Pedir Visualización Detallada del Sistema" se continua en el paso 6.

# Análisis de la Funcionalidad del Sistema

*RUP: Elaboración*

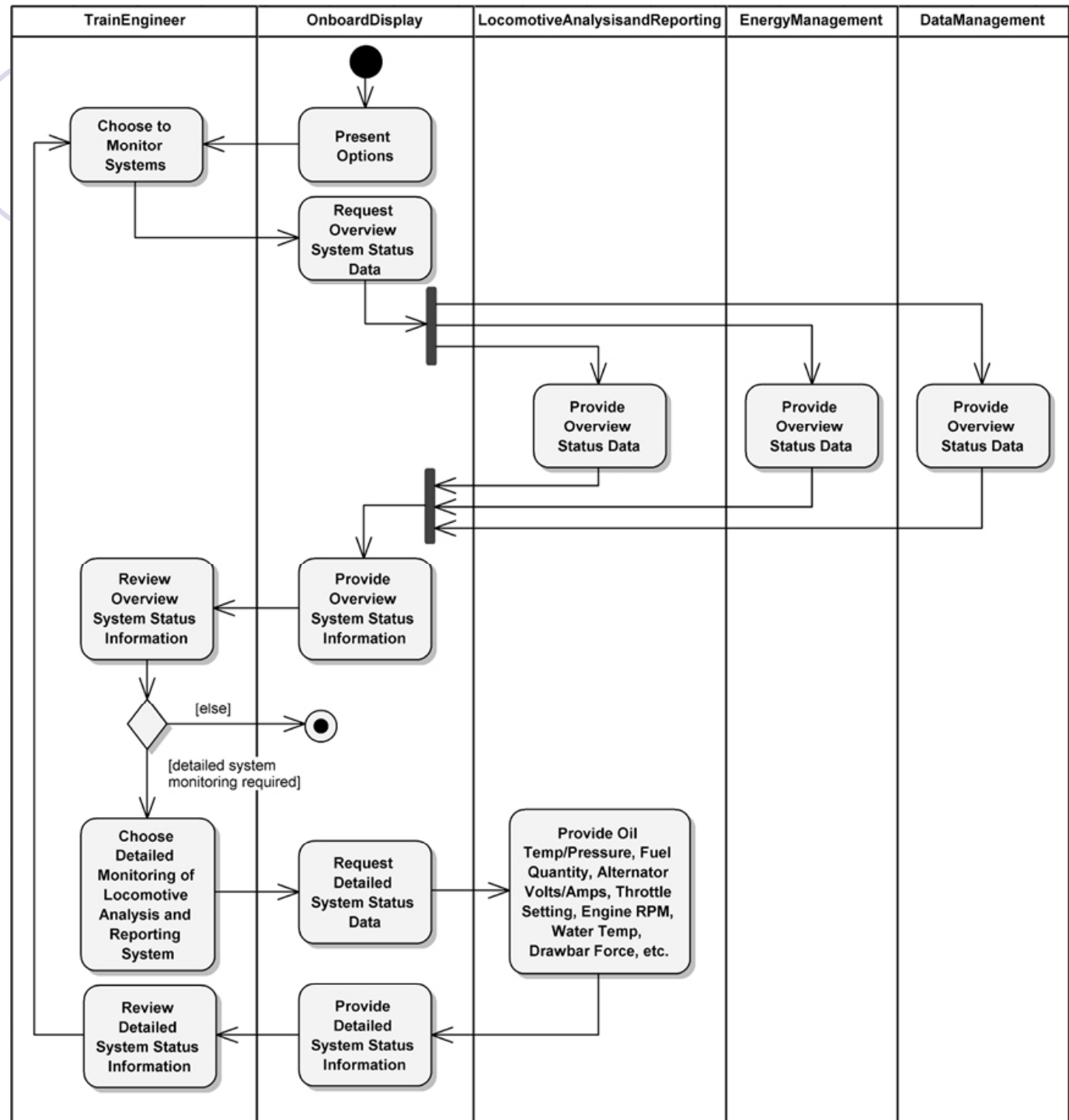
Caso de uso enrutar tren



# Análisis de la Funcionalidad del Sistema

*RUP: Elaboración*

Caso de uso  
controlar los  
sistemas del  
tren y escenario  
alternativo

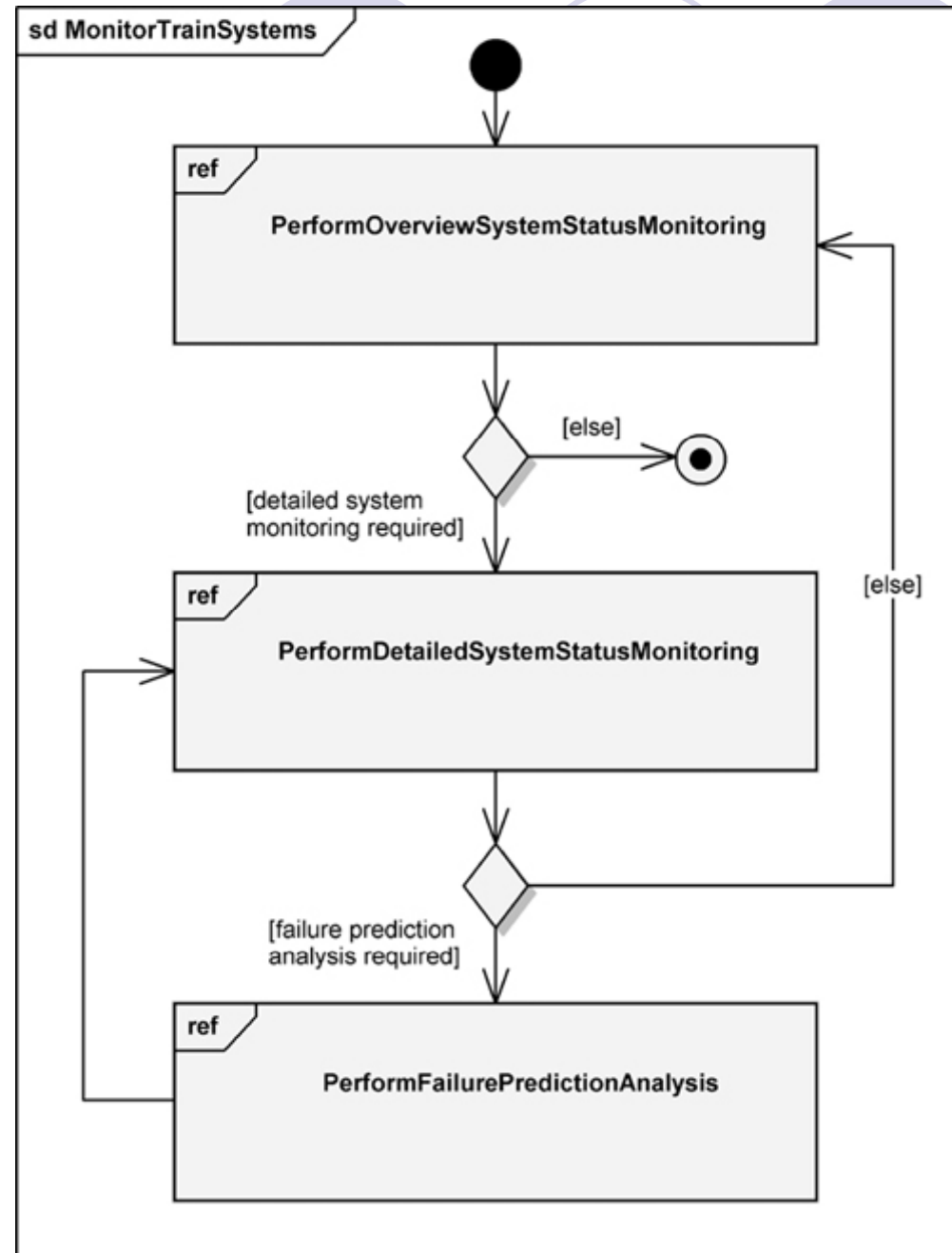




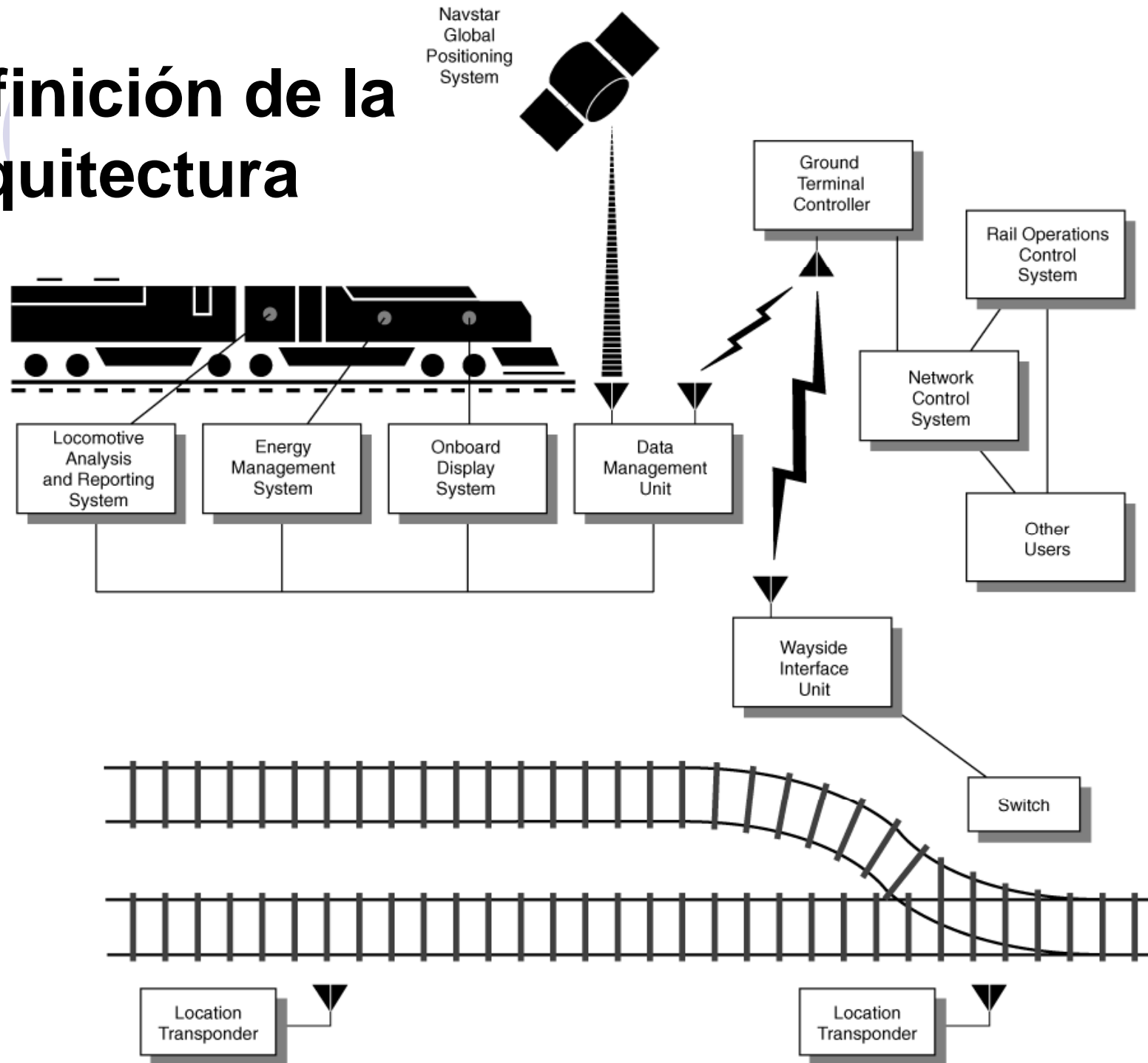
# Análisis de la Funcionalidad del Sistema

## *Elaboración*

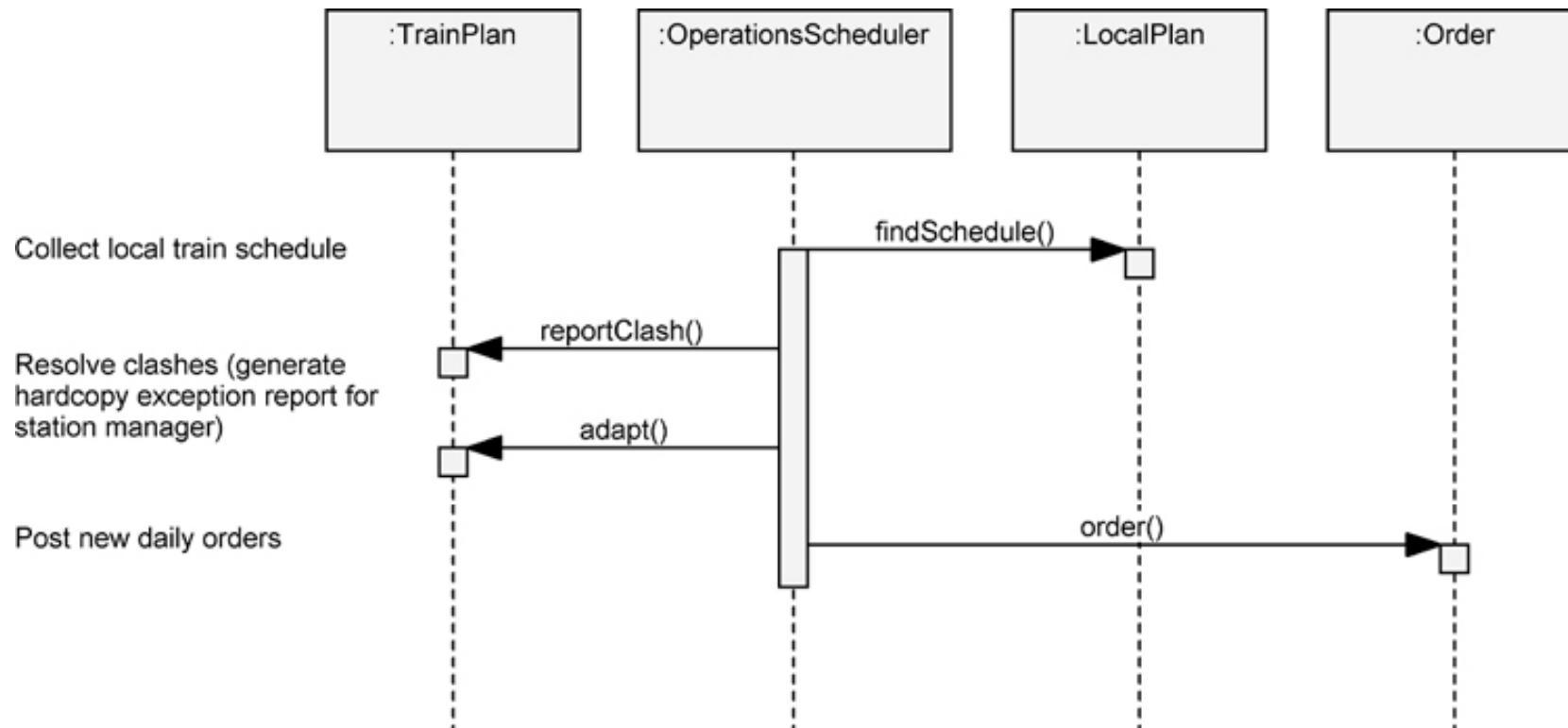
Diagrama de visión conjunta de la interacción que muestra la relación entre los distintos escenarios del caso de uso “controlar los sistemas del tren”



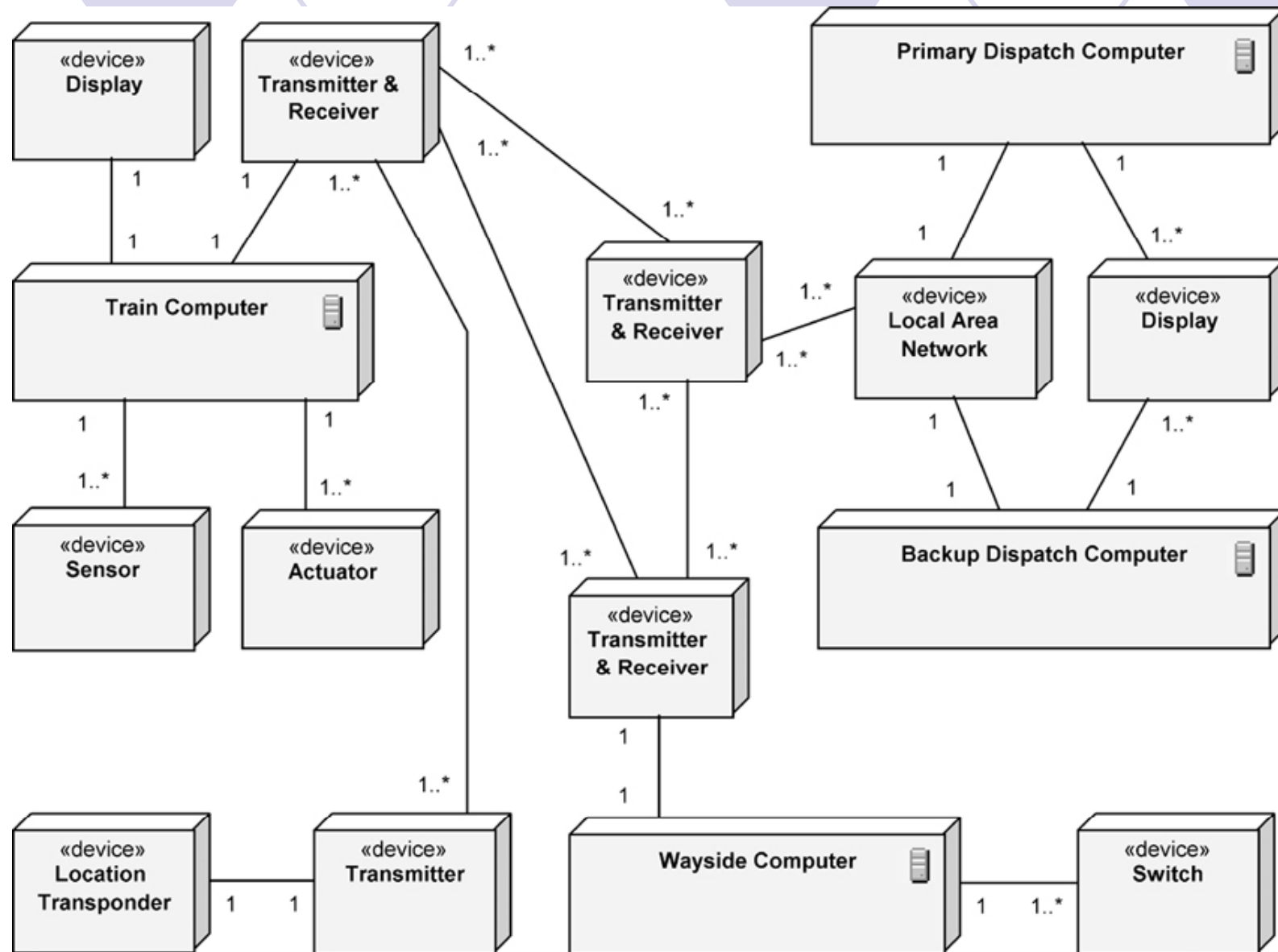
# Definición de la Arquitectura



# Ingeniería de Sistemas



# Definición de la Arquitectura



# Abstracciones y Mecanismos

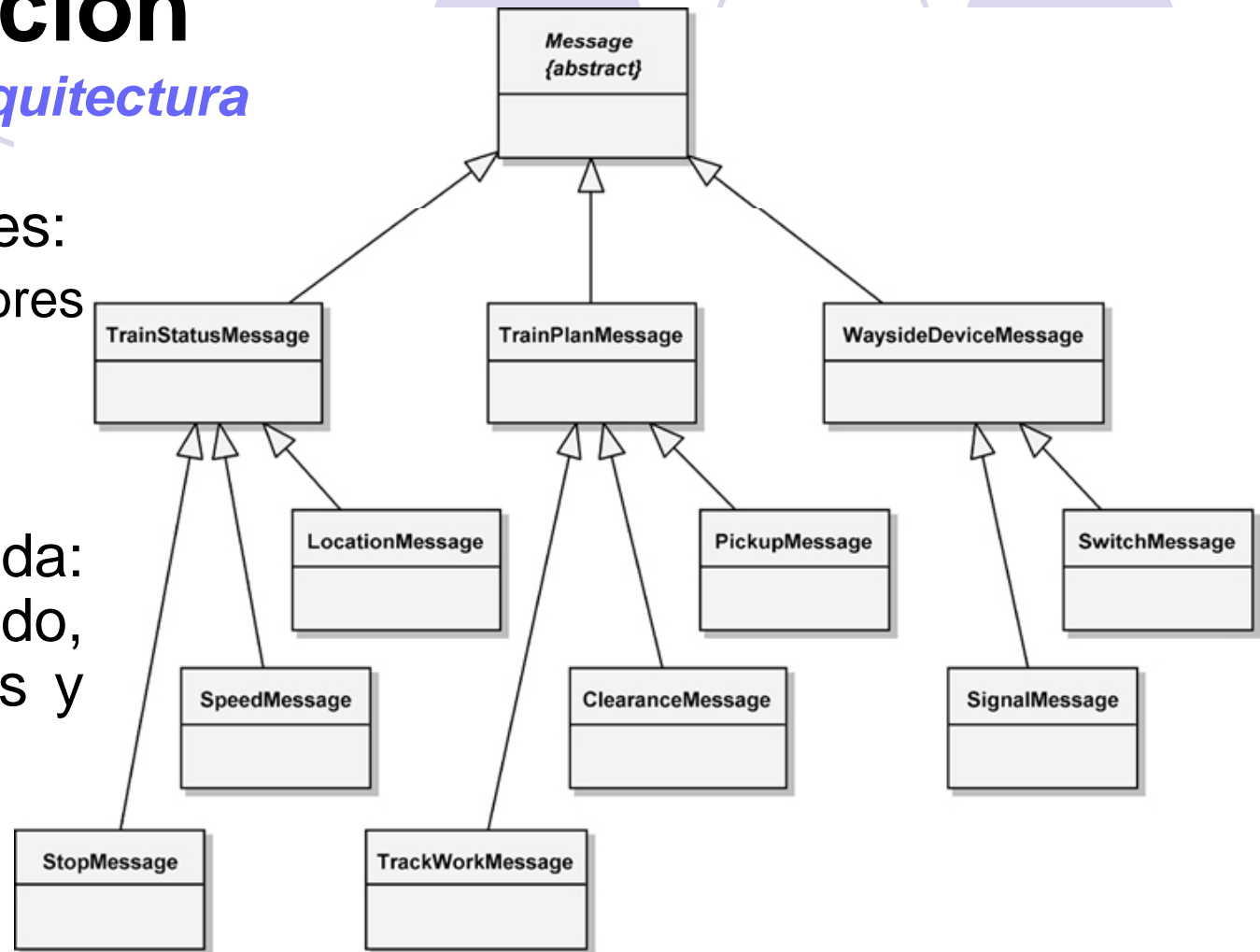
## *Análisis de dominio*

- El sistema comprende cuatro abstracciones o mecanismos principales:
  - Red y Comunicaciones.
  - Base de Datos.
  - Interfaz hombre-máquina.
  - Control en tiempo real de dispositivos analógicos y digitales.
- Hay tres abstracciones comunes:
  - Trenes: incluye vagones y locomotoras.
  - Vías de tren: perfil, grado, dispositivos de rail.
  - Planes: horarios, órdenes, permisos, autoridad y asignación de personal.
- Mecanismos para las abstracciones:
  - Paso de mensajes.
  - Planificación de los horarios del tren.
  - Visualización de información.
  - Adquisición de datos de los sensores.

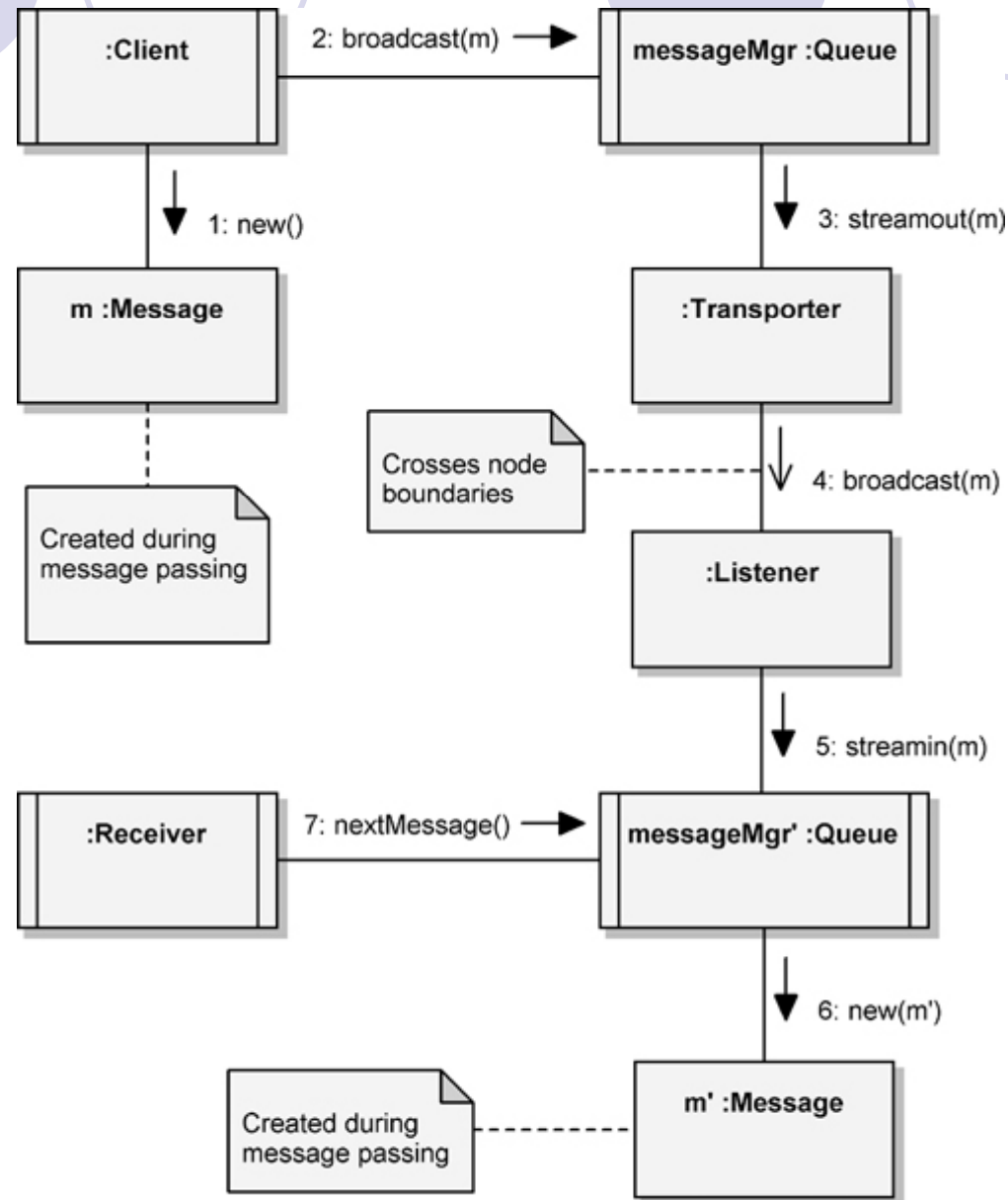
# Construcción

## *Diseño de la Arquitectura*

- Paso de mensajes:
  - Entre ordenadores y dispositivos.
  - Entre ordenadores.
- Red distribuida: contemplar ruido, fallos de equipos y seguridad.

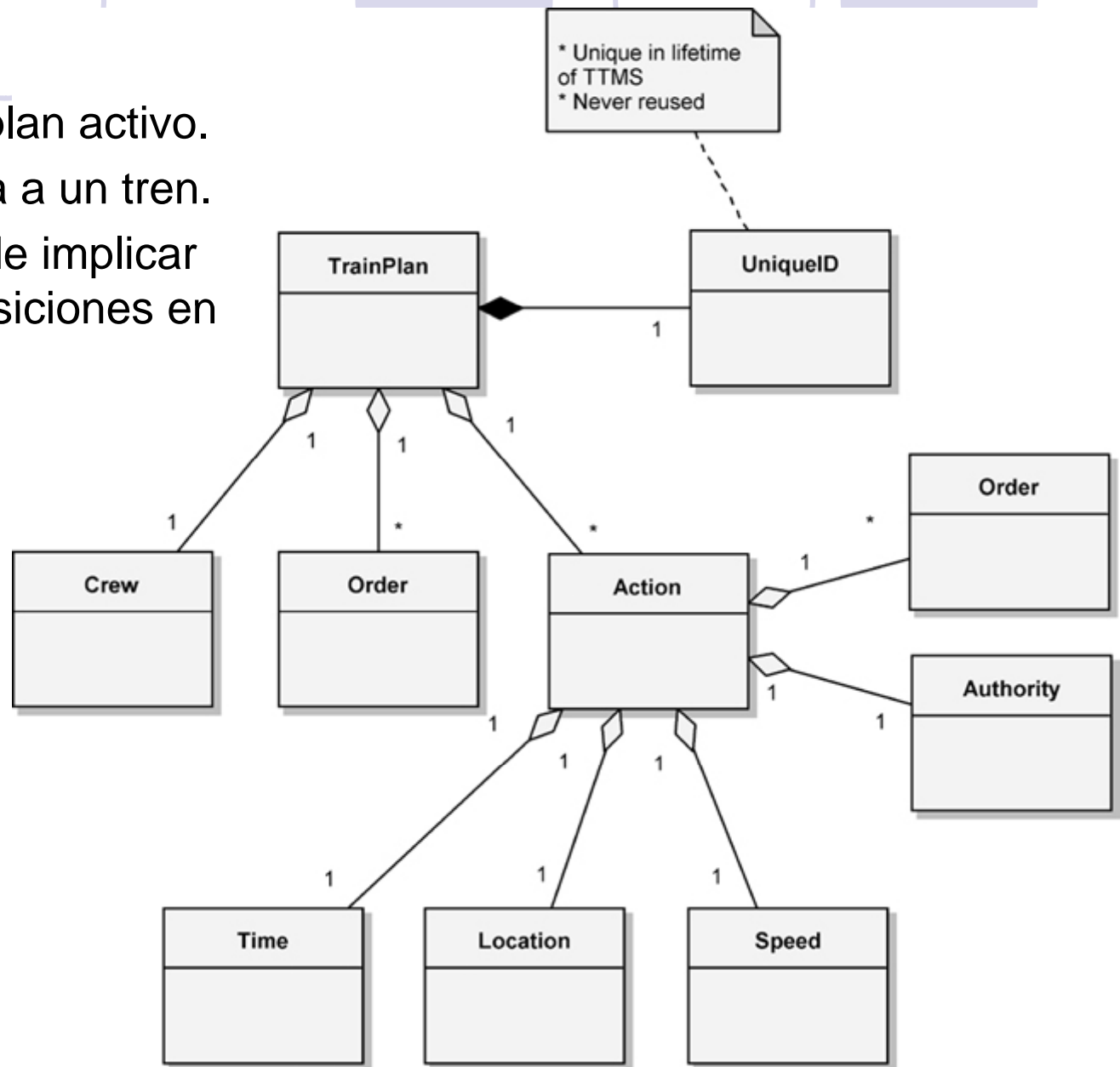


# Mecanismo de Paso de Mensajes



# Planificación de horarios

- Cada tren tiene un plan activo.
- Cada plan se asigna a un tren.
- Un plan puede implicar varias órdenes y posiciones en las vías.



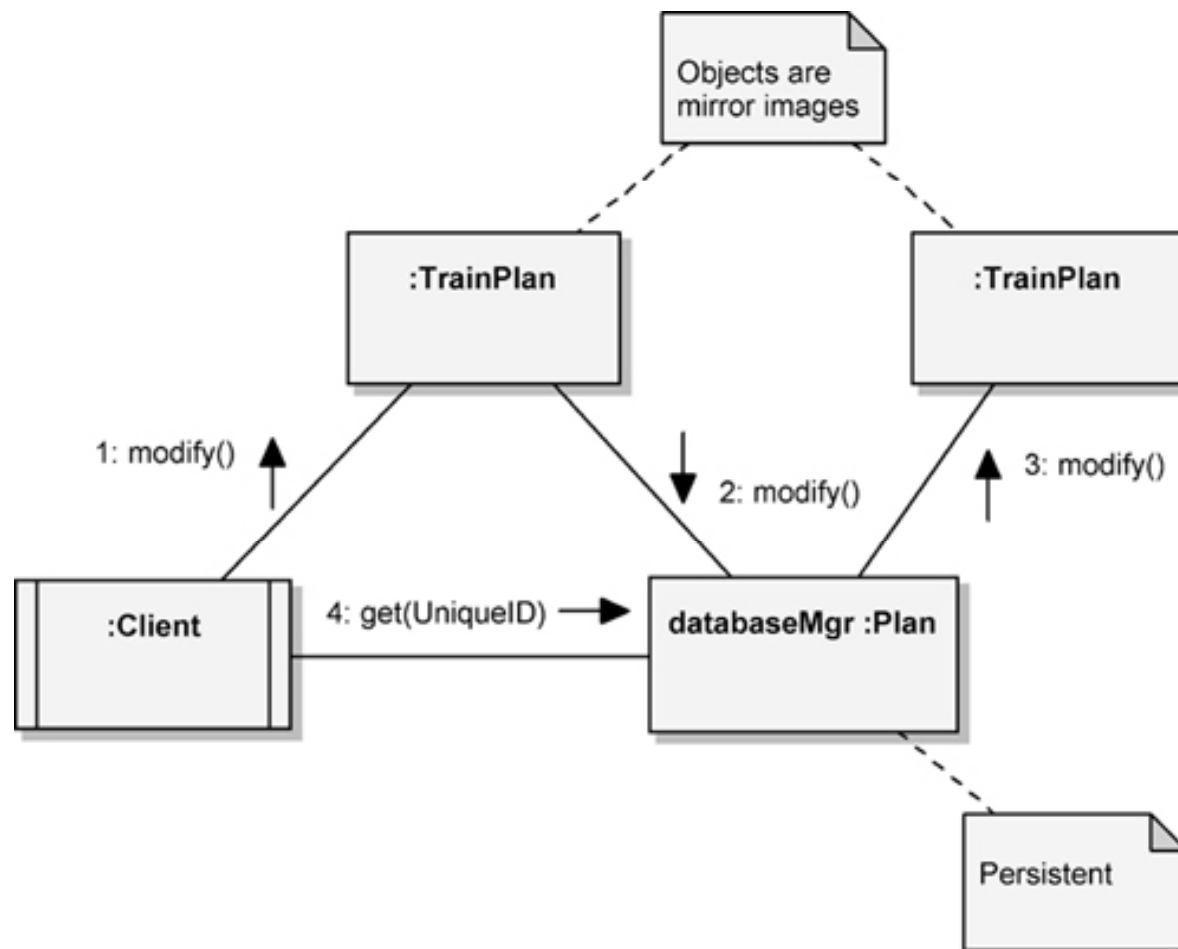


# Planificación de horarios

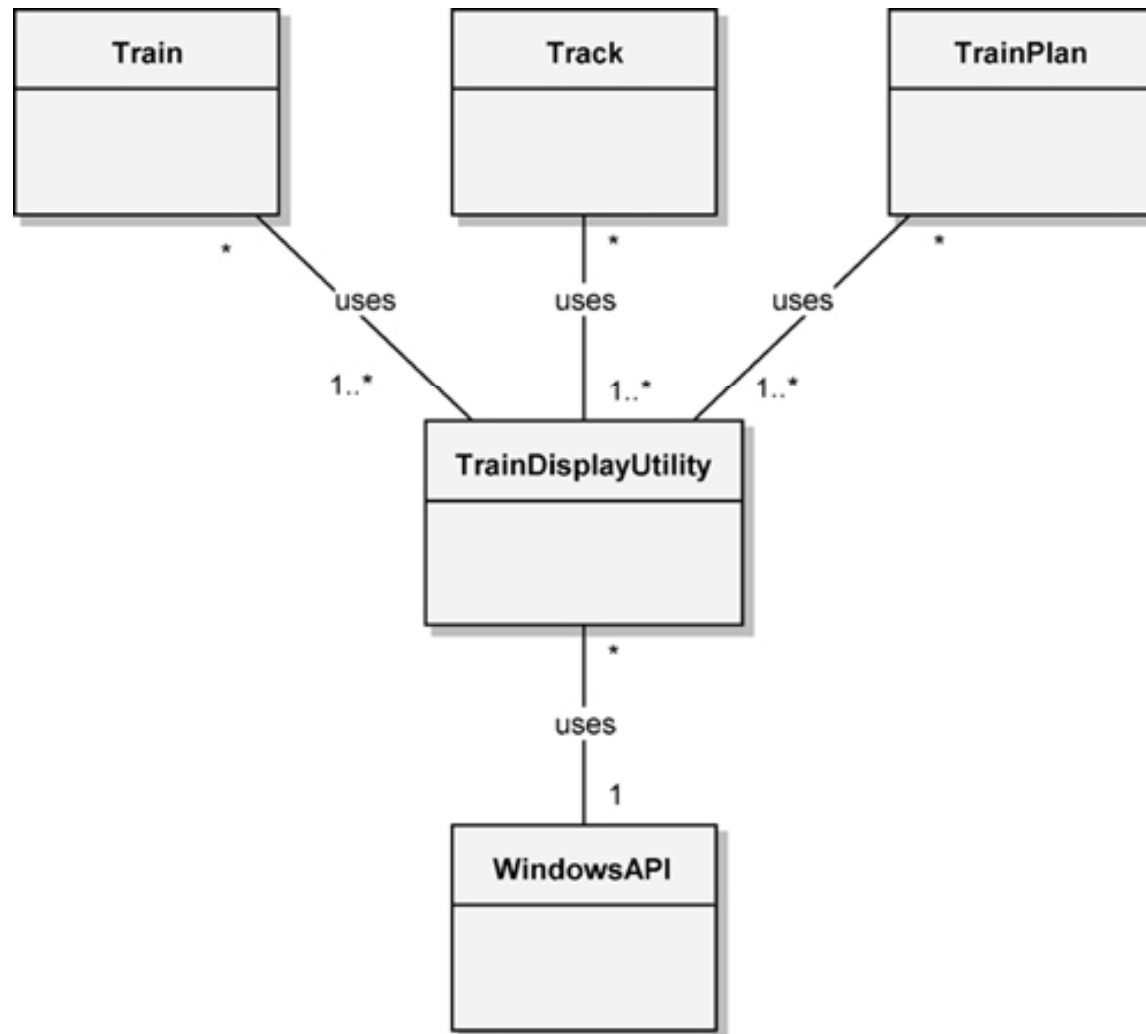
- Ejemplo de las acciones que puede contener un plan.

Time	Location	Speed	Authority	Orders
0800	Pueblo	As posted	See yardmaster	Depart yard
1100	Colorado Springs	40 mph		Set out 30 cars
1300	Denver	45 mph		Set out 20 cars
1600	Pueblo	As posted		Return to yard

# Planificación de horarios



# Visualización de Información



# Arquitectura del Sistema

