

Programación Avanzada

Diseño

Guías para el Abordaje del Diseño

Contenido

- Introducción
- Caso de Estudio
- Guías para el Abordaje del Diseño

Introducción

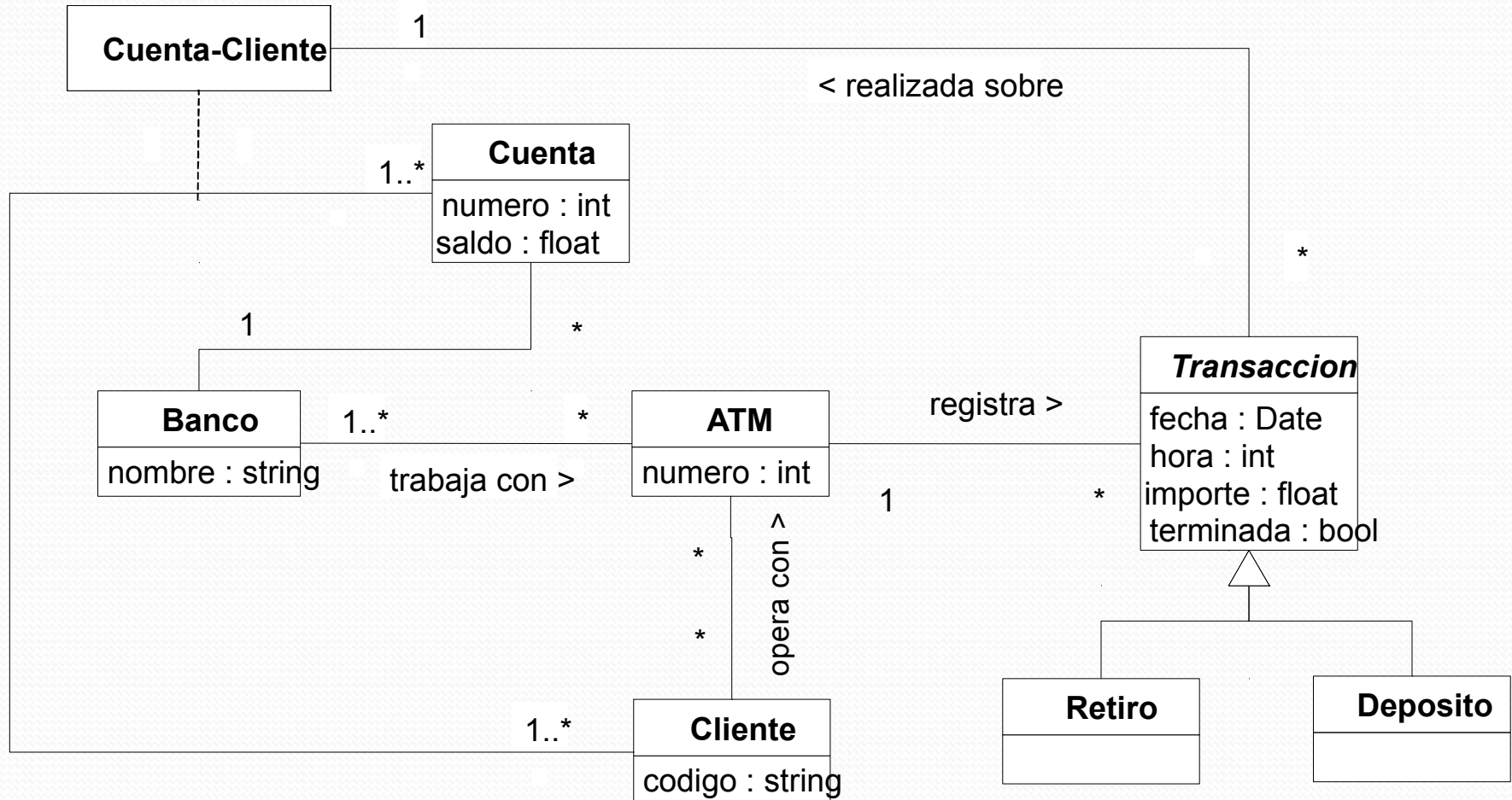
- Se desea abordar la etapa de diseño con un enfoque sistemático
- Se presentarán pautas para organizar de mejor forma la tarea
- Se ejemplificarán las mismas por medio de un caso de estudio

Caso de Estudio

- Gestión de cuentas en bancos a través de ATMs (cajeros automáticos):
 - A través de una red de ATMs, los clientes acceden a sus cuentas sobre las cuales realizan transacciones (depósitos y retiros)
 - Las cuentas pueden ser compartidas por más de un cliente

Caso de Estudio

Modelo de Dominio



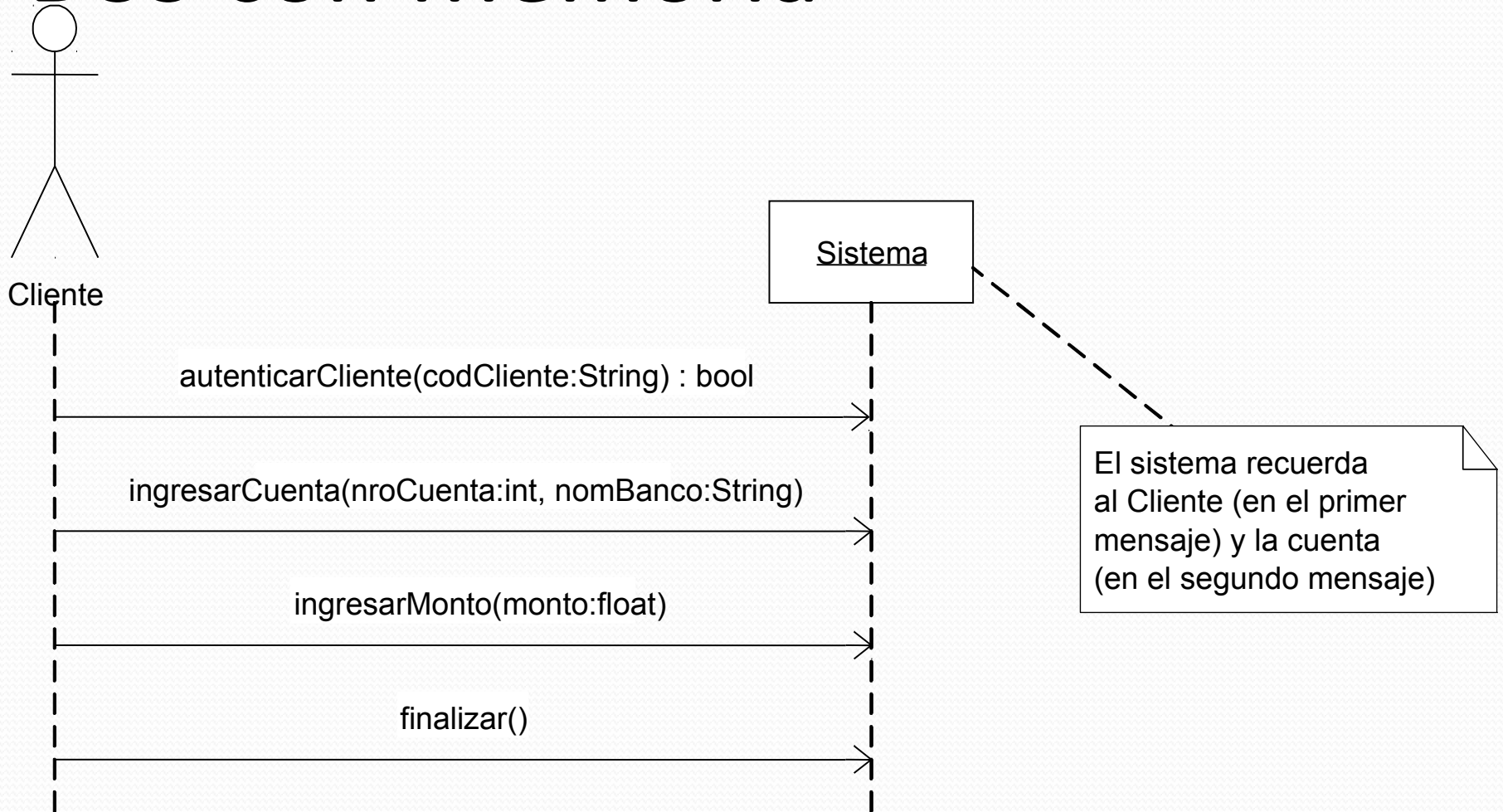
Caso de Estudio

Caso de Uso

Nombre	Retiro de Cuenta	Actores	Cliente
Sinopsis	El caso de uso comienza cuando el cliente inserta su tarjeta en el cajero e ingresa su clave de usuario. Tras validar al cliente, el sistema recibe el nombre del banco y el número de cuenta para iniciar la transacción de retiro correspondiente. El cliente ingresa el monto que desea retirar de la cuenta y el sistema realiza el débito. Finalmente, el cliente retira su tarjeta.		

Caso de Estudio

DSS con Memoria



Caso de Estudio

Descripción de Operaciones

- **autenticarCliente** (**codCliente:String**):**bool**
 - Valida la existencia del cliente
- **ingresarCuenta** (**nroCuenta:int**,
nomBanco:String)
 - Obtiene la cuenta **nroCuenta** del banco **nomBanco** sobre la cual se realizará la transacción
- **ingresarMonto** (**monto:float**)
 - Realiza la transacción de débito por el monto indicado sobre la cuenta del cliente
- **finalizar** ()
 - Finaliza la operativa del sistema

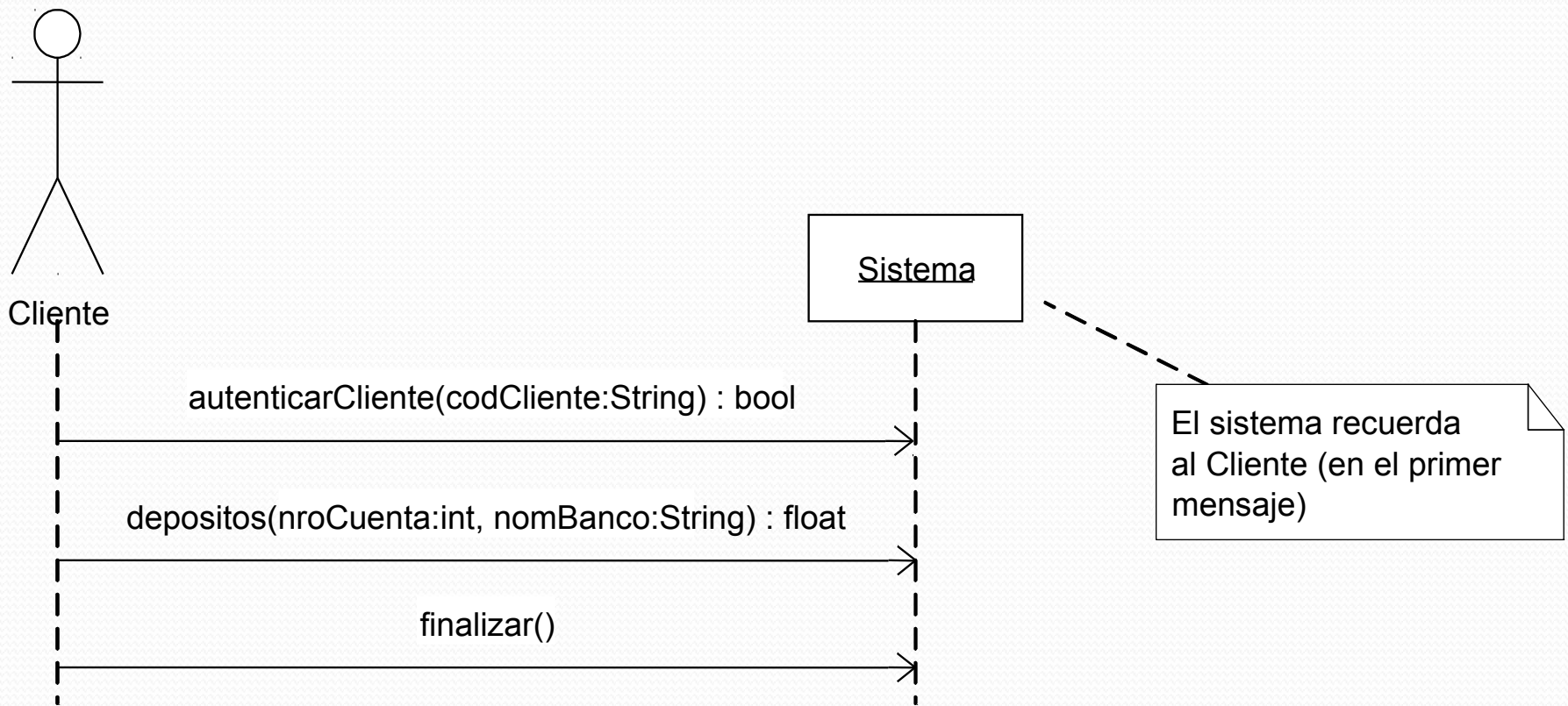
Caso de Estudio

Caso de Uso (2)

Nombre	Consulta de Depósitos	Actores	Cliente
Sinopsis	<p>El caso de uso comienza cuando el cliente inserta su tarjeta en el cajero e ingresa su clave de usuario. Tras ingresar los datos de validación (igual que en el caso de uso Retiro de Cuenta), el cliente indica el nombre del banco y el número de cuenta sobre la cual desea consultar el total de depósitos (históricos).</p> <p>Posteriormente, el sistema calcula el total de depósitos histórico y lo devuelve. Finalmente, el cliente retira su tarjeta.</p>		

Caso de Estudio

DSS con Memoria (2)



Descripción de Operaciones (2)

- **autenticarCliente** (**codCliente:String**):**bool**
 - Misma operación que en el DSS anterior
- **depositos** (**nroCuenta:int**,
nomBanco:String):**float**
 - Devuelve la suma de los montos de todos los depósitos realizados en la cuenta **nroCuenta** del banco **nomBanco**
- **finalizar** ()
 - Misma operación que en el DSS anterior

Guías para el Abordaje del Diseño

- El abordaje de la etapa de diseño puede realizarse sistemáticamente
- Por ejemplo, considerando estos pasos:
 1. Organizar Operaciones
 2. Definir Ubicación de Instancias
 3. Definir Colaboraciones
 4. Diseñar Colaboraciones

Organizar Operaciones

- Definir los Controladores a utilizar
- Definir las Interfaces del Sistema que contendrán las operaciones del sistema
- Organizar operaciones según:
 - Afinidad temática (según dominio)
 - Afinidad funcional (según objetivos)
 - Casos de Uso
- Definir la Fábrica de controladores

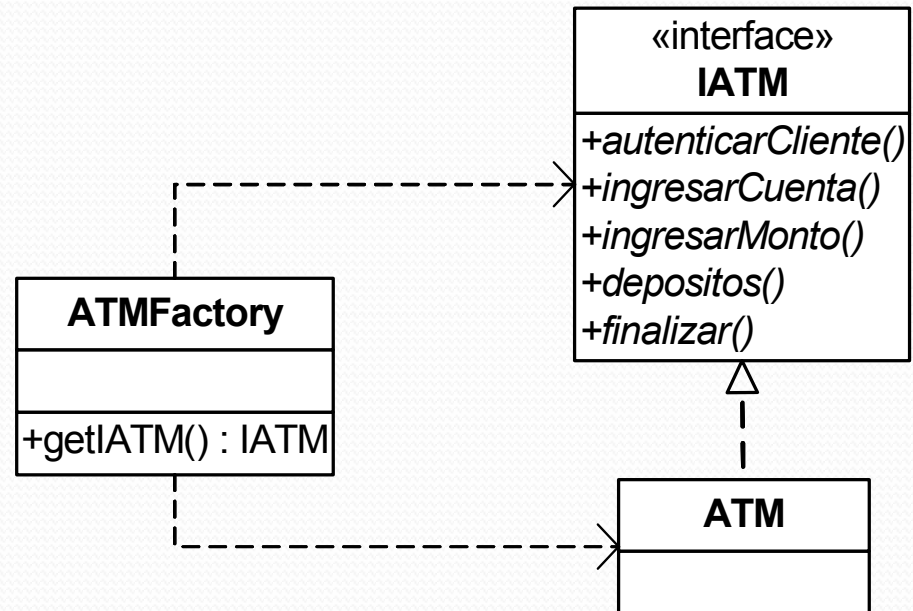
Organizar Operaciones (2)

- Al definir Controladores, considerar
 - Operaciones repetidas en casos de uso
 - Memoria del Sistema
- Si un Controlador realiza una Interfaz del Sistema, asigna un método a todas las operaciones presentes en ella.

Ejemplo (Organizar Operaciones)

- ¿Algún concepto del dominio podría ser un Controlador?, ¿ATM?, ¿Banco?

Una posible opción



Definir Ubicación de Instancias

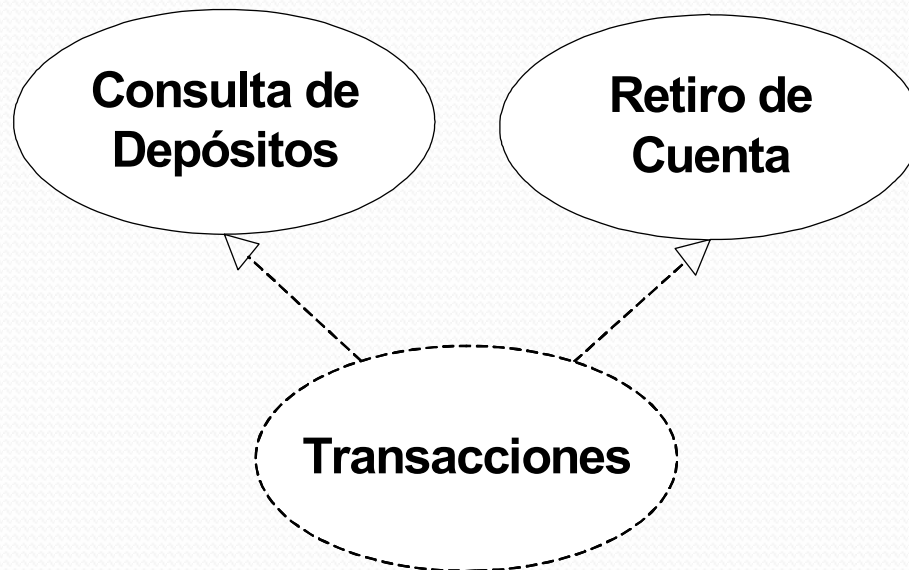
- Diferenciar:
 - Las colecciones que pueden ser alojadas en un controlador (ej: ATM, Banco)
 - Las que serán accedidas únicamente a través de otra clase (ej: Cuenta accesible a través de Banco)
 - En caso de ser necesario, alojar separadamente una colección que sea compartida entre varios controladores

Definir Colaboraciones

- Una colaboración realiza uno o más casos de uso
- Agrupar casos de uso con cierta afinidad
 - Comúnmente afinidad temática pero no hay una regla estricta
- Definir una colaboración por cada grupo de casos de uso, asignándole un nombre
- Priorizar las colaboraciones según el impacto esperado sobre el diseño

Ejemplo (Definir Colaboraciones)

- Definir una sola colaboración para ambos casos de uso es beneficioso ya que están relacionados



Diseñar Colaboraciones

- Diseñar cada colaboración en orden de prioridad:
 - Realizar diagramas de comunicación para las operaciones del sistema involucradas
 - Considerar (a) criterios de asignación de responsabilidades, (b) decisiones tomadas en iteraciones anteriores (consistencia) y (c) nuevos problemas de diseño
 - Realizar el diagrama de clases de diseño

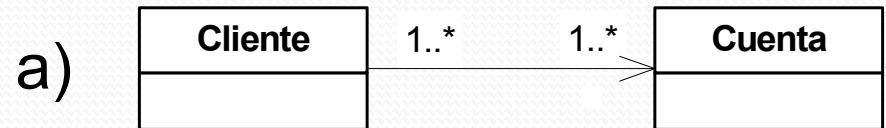
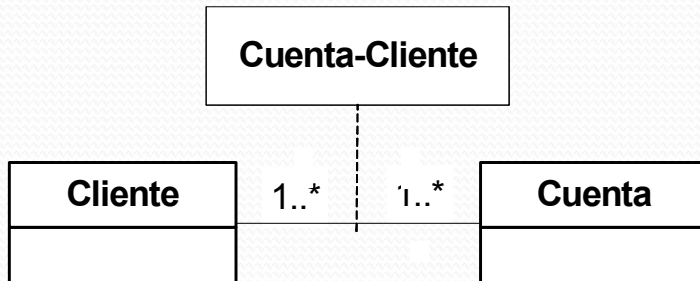
Ejemplo (Diseñar Colaboraciones)

- ¿Cómo se asignan responsabilidades?
 - ¿Quién crea las transacciones?
 - ¿ATM?, ¿Banco?, ¿Cuenta?
 - ¿Quién es el experto en calcular el total de depósitos realizados?
 - ¿Banco?, ¿Cuenta?, ¿Cliente?
 - ¿Qué visibilidades se necesitan?
 - ¿ATM → Transacción?, ¿Banco → Cuenta?, ¿Cliente → Transacción?, ¿ATM → Cuenta?

Guías para el Abordaje del Diseño

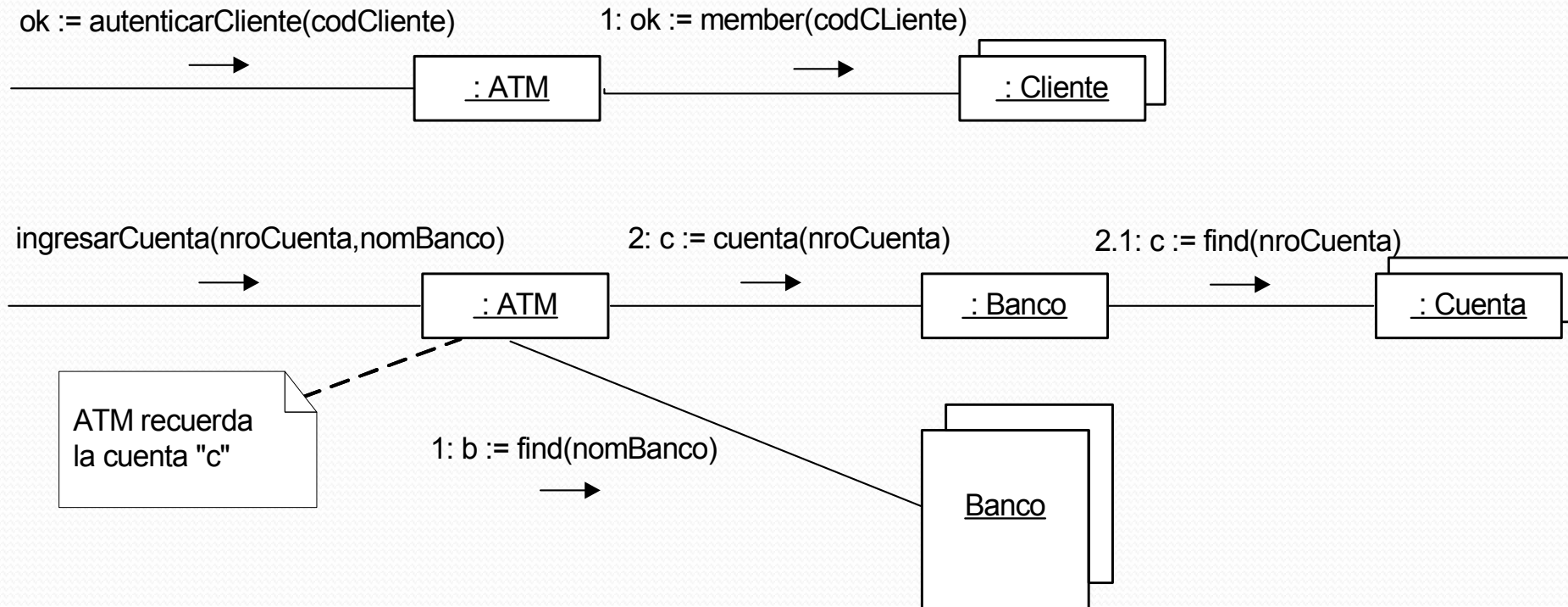
Ejemplo (Diseñar Colaboraciones) (2)

- ¿Qué sucede con los tipos asociativos?
 - Si poseen información relevante puede convenir mantenerlas



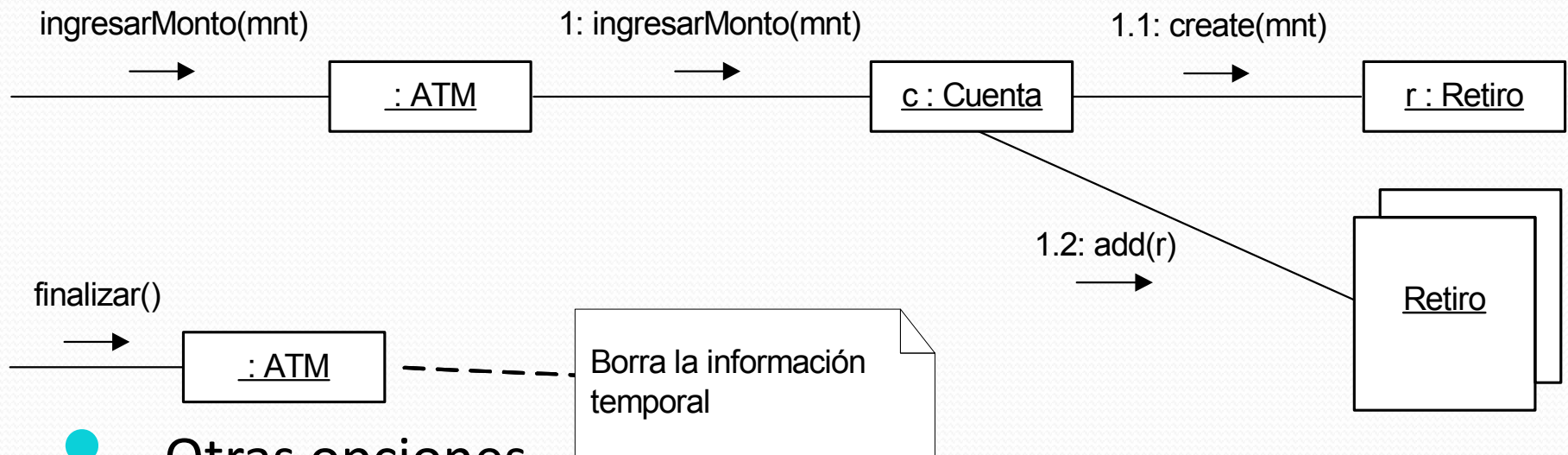
Ejemplo (Diseñar Colaboraciones) (3)

- Una posible solución



Guías para el Abordaje del Diseño

Ejemplo (Diseñar Colaboraciones) (4)



Otras opciones

- Asociar la transacción con el cliente
- Delegar las transacciones al banco

Guías para el Abordaje del Diseño

Ejemplo (Diseñar Colaboraciones) (5)

