Programación Avanzada

Diseño

Diseño de la Estructura de una Colaboración

Contenido

- Introducción
- Diagrama de Clases de Diseño

Introducción

- La asignación de responsabilidades ha sido completada
- La parte dinámica de la colaboración que se está diseñando ha sido determinada
- Habiendo finalizado la construcción de los diagramas de comunicación es posible especificar la parte estructural de la colaboración

Introducción (2)

- Esta especificación se realizará mediante los diagramas de clases de UML
- Estos diagramas:
  - Ilustran la estructura de la solución
  - Están anotados con información de diseño, como por ejemplo operaciones y navegabilidades
- Al artefacto resultante lo llamamos **Diagrama de Clases de Diseño** (DCD) y será incluido en el Modelo de Diseño

Diagrama de Clases de Diseño

- Un Diagrama de Clases de Diseño especifica la estructura de una colaboración
- Los elementos que contiene son representaciones gráficas de algunos elementos de diseño contenidos en el modelo
- Los elementos a incluir son solamente aquellos que sean necesarios para solucionar el/los caso/s de uso realizado/s por la colaboración

Diagrama de Clases de Diseño (2)

- Elementos de diseño a incluir:
  - Clases, asociaciones y atributos
  - Navegabilidades de asociaciones
  - Operaciones de clases y existencia de métodos
  - Interfaces con sus operaciones
  - Información acerca del tipo de los atributos y de los valores devueltos por las operaciones (incluyendo datatypes)
  - Generalizaciones entre clases o interfaces
  - Dependencias entre elementos

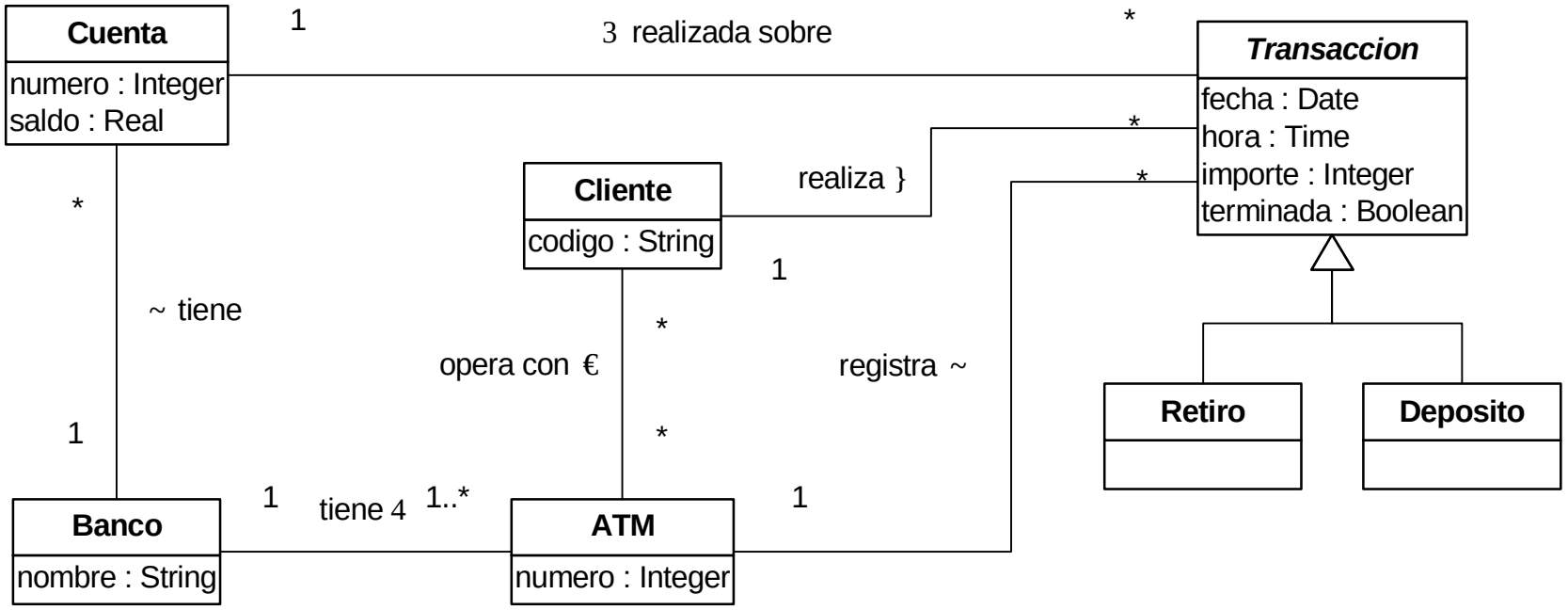
Construcción de un DCD

- Para la construcción de un DCD:
  1. Identificar todas las clases que participan de la solución de los casos de uso. Hacer esto analizando los diagramas de comunicación
  2. Incluirlas en un el diagrama de clases
  3. Replicar los atributos de los conceptos correspondientes en el Modelo de Dominio, agregando aquellos nuevos que sean necesarios
  4. Agregar las operaciones correspondientes a cada clase analizando los diagramas de comunicación

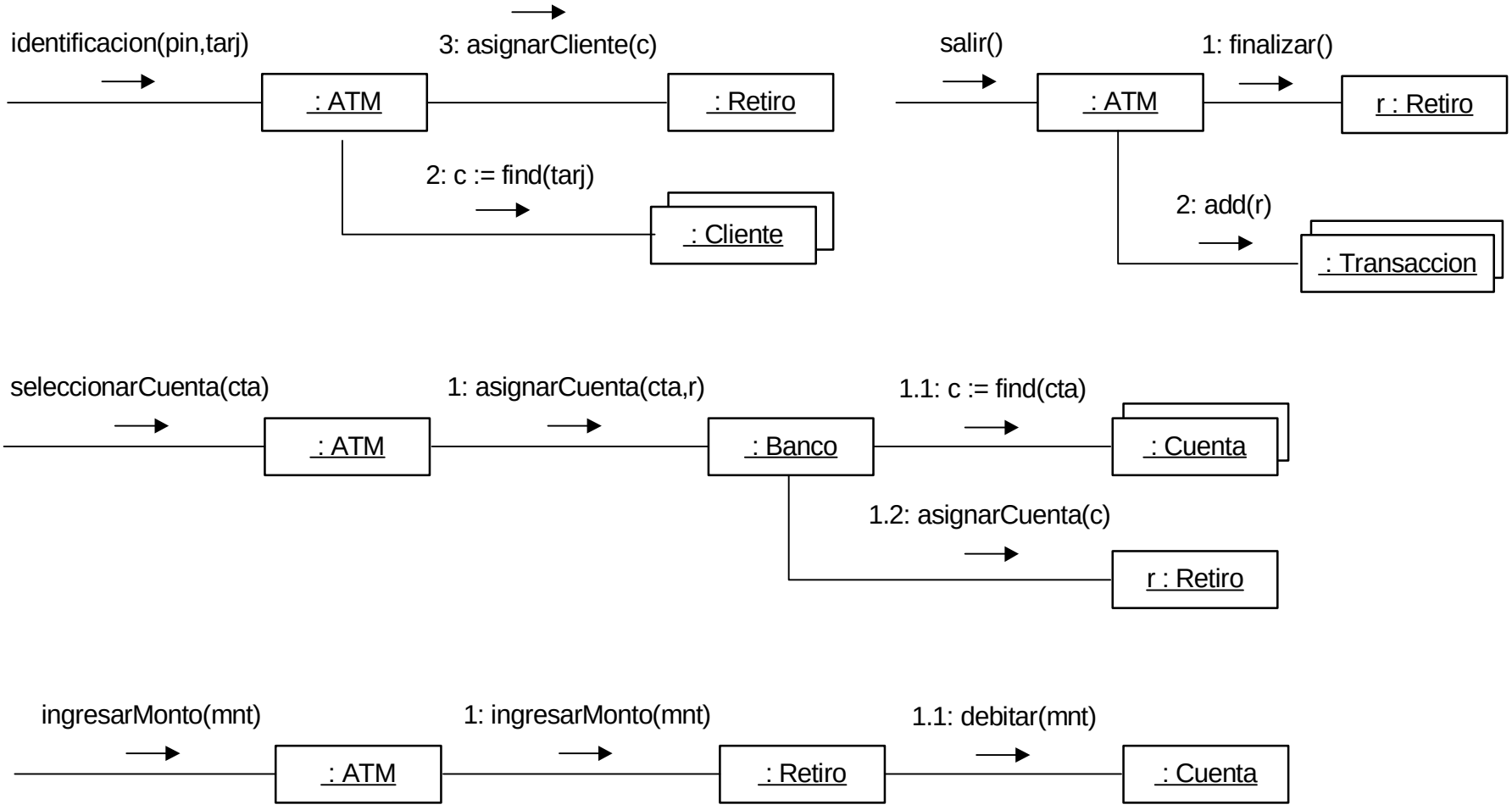
Construcción de un DCD (2)

- Para la construcción de un DCD (cont.):
  5. Agregar la información de tipos a los atributos y operaciones
  6. Agregar las asociaciones necesarias para permitir las visibilidades por atributo requeridas en los diagramas de comunicación
  7. Agregar navegabilidades para indicar la dirección de cada visibilidad por atributo
  8. Agregar dependencias para reflejar los demás tipos de visibilidades existentes
  9. Agregar interfaces, fábricas y datatypes

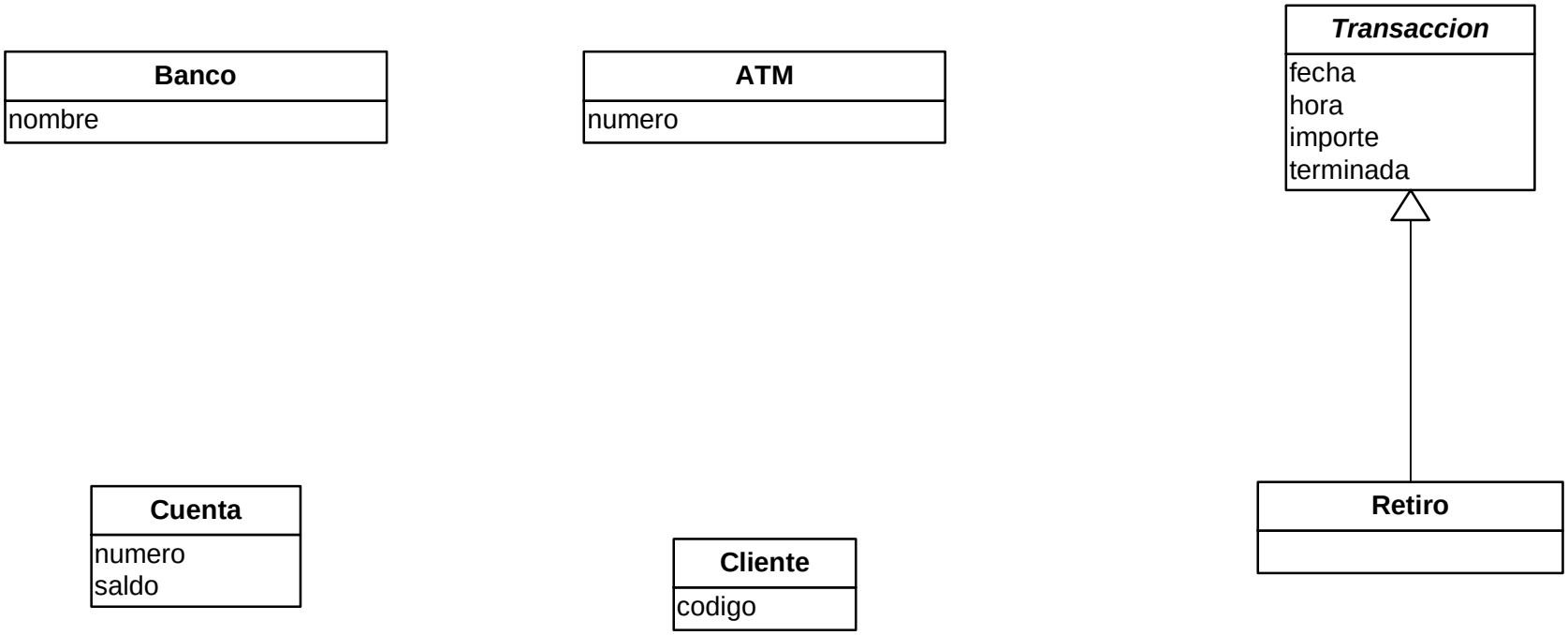
Construcción de un DCD
Información Previa (Dominio)



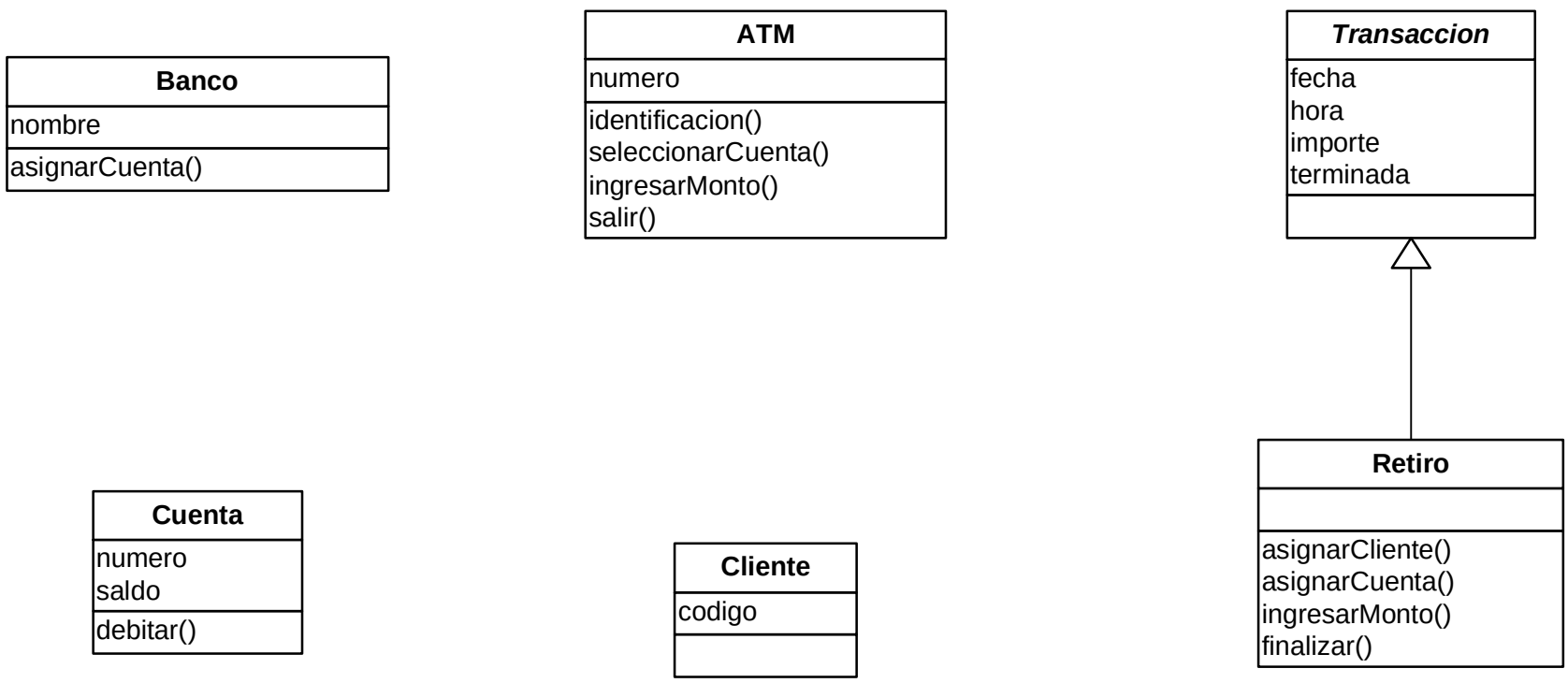
Construcción de un DCD
Información Previa (Interacciones)



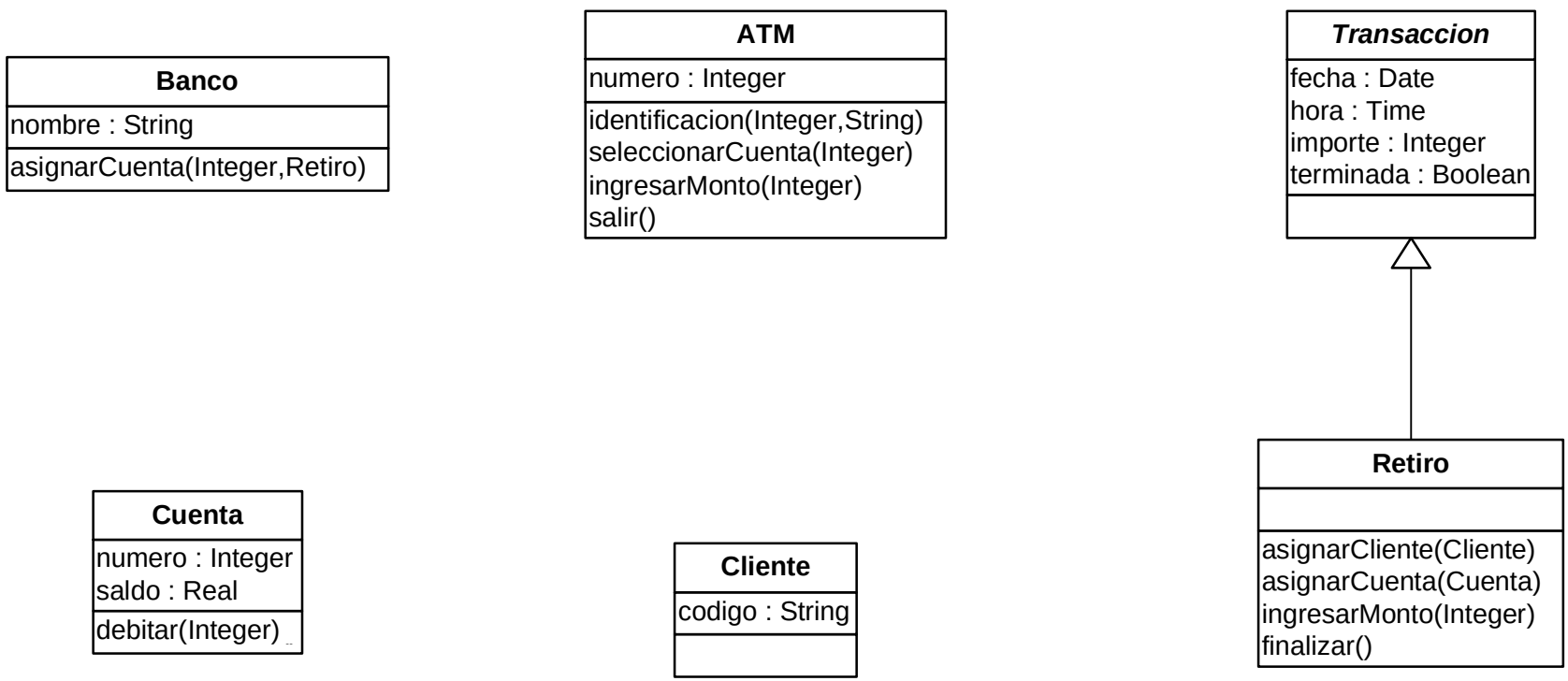
Identificar las Clases e Ilustrarlas
Pasos 1, 2 y 3



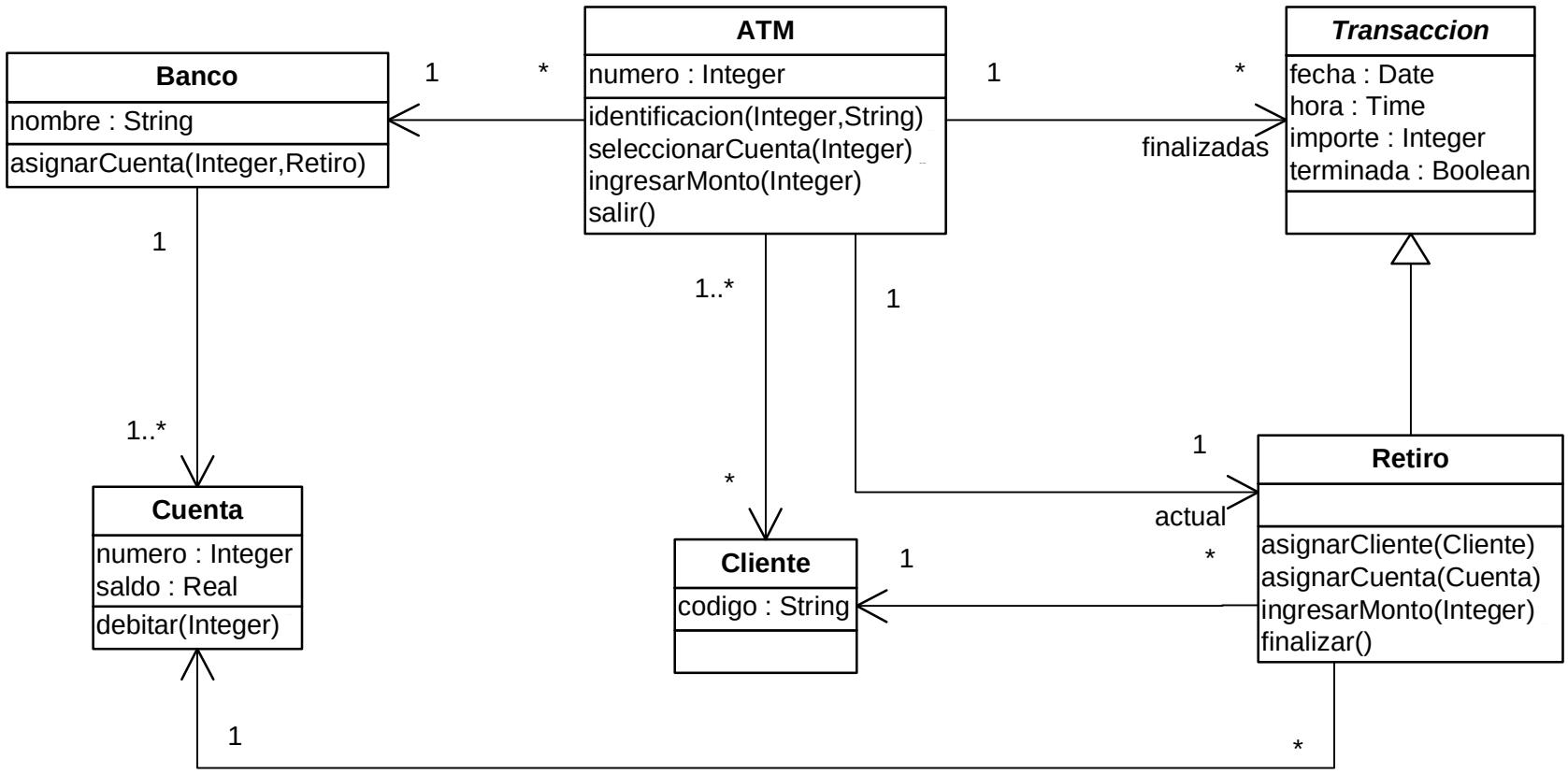
Agregar Operaciones y Métodos
Paso 4



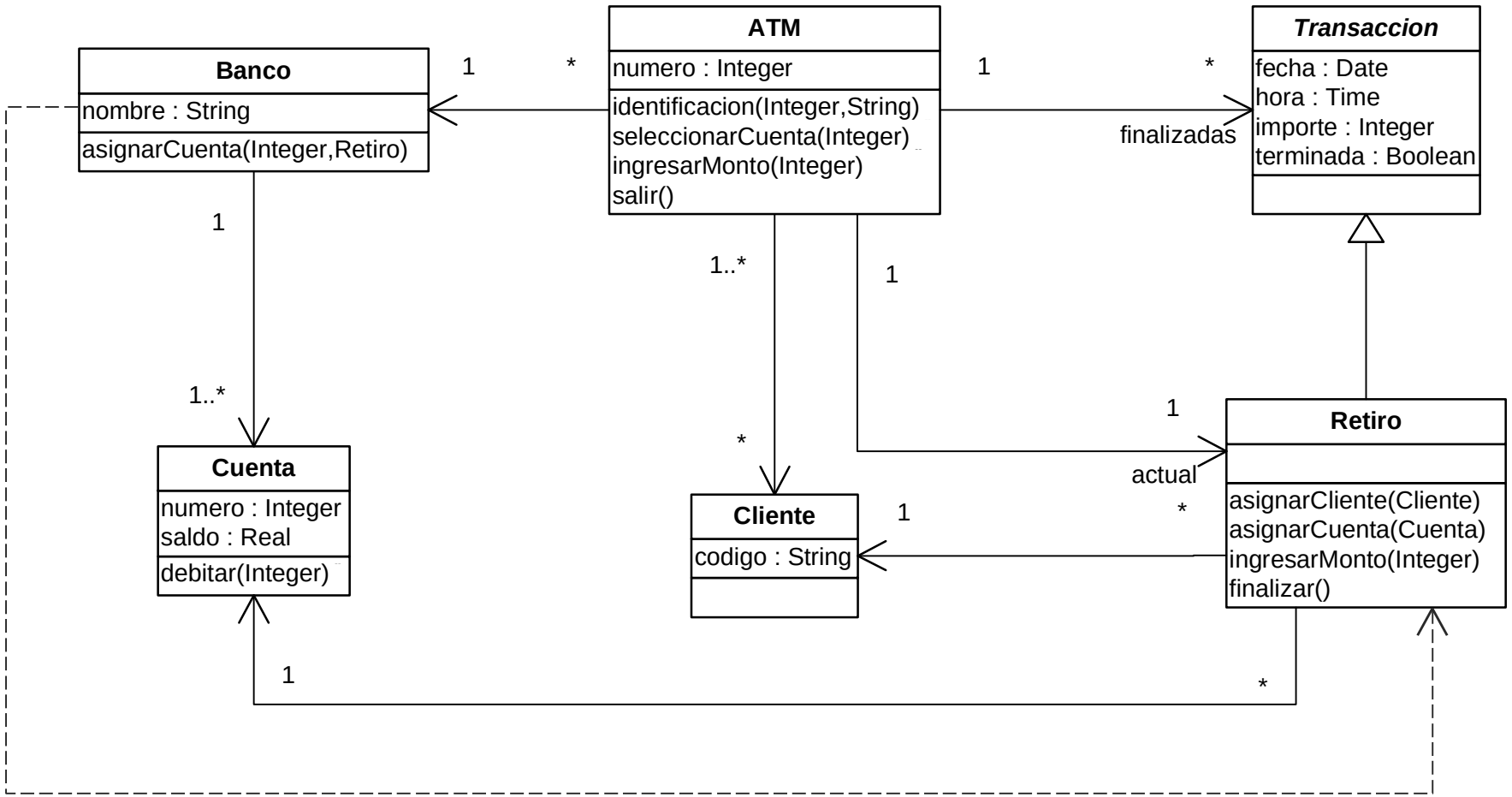
Agregar Información de Tipos
Paso 5



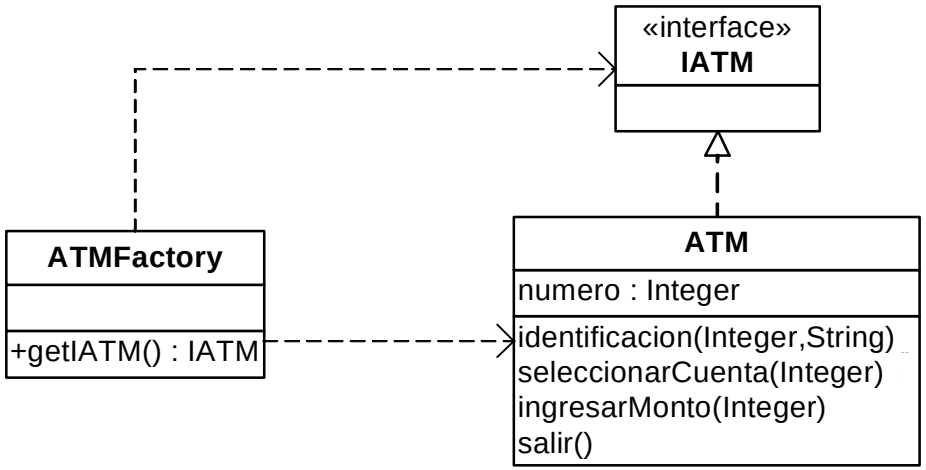
Agregar Asociaciones y Navegabilidad
Pasos 6 y 7



Agregar Dependencias
Paso 8



Agregar interfaces, fábricas y datatypes
Paso 9



Inclusión de Operaciones

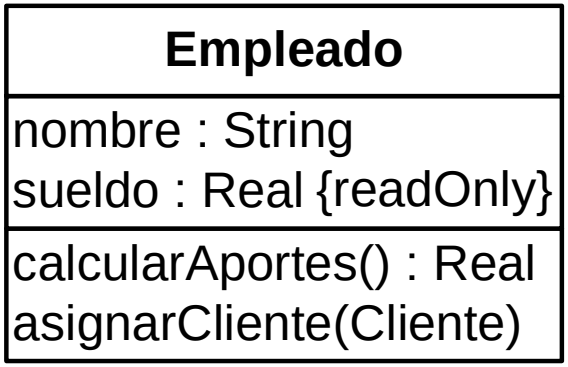
- Operación *create*:
  - La operación *create* es utilizada para la creación de instancias
  - Esta forma es propia de UML e independiente de todo lenguaje de programación
  - Este mensaje se corresponde con los constructores de clases
  - Los constructores están siempre presentes en las clases por lo que es común omitirlos en los diagramas de clases de diseño

Inclusión de Operaciones (2)

- Operaciones de acceso:
  - Son utilizados para obtener el valor de un atributo (*get*) o para modificarlo (*set*)
  - Lo usual es declarar los atributos como privados y necesitar este tipo de operaciones
  - Sin embargo se las excluye de los diagramas
  - Por defecto se asume que un atributo tiene su *get* y *set* asociado
  - Es posible indicar que para un atributo no se brindará la operación set correspondiente aplicándole la restricción {readOnly}

Inclusión de Operaciones (3)

- Operaciones de acceso (cont.)
  - Ejemplo: la implementación de la clase Empleado contendrá las operaciones
    - `calcularAportes()`
    - `asignarCliente()`
    - `getNombre()`
    - `setNombre()`
    - `getSueldo()`

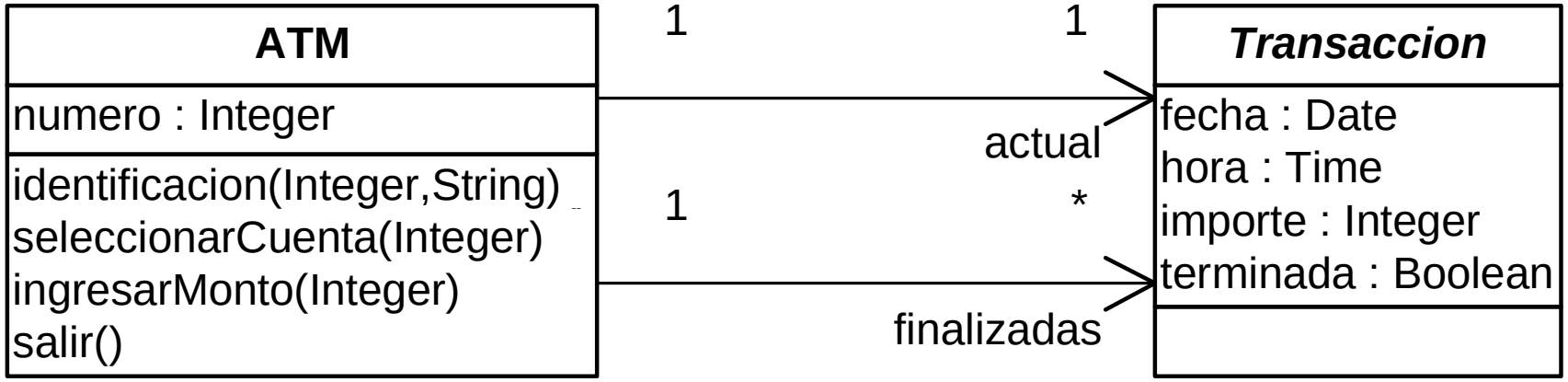


Inclusión de Colecciones

- Las colecciones (tratadas como fuera indicado) usualmente disponen todas de las mismas operaciones
- Por tal razón no aportarían mayor información al diagrama y es común omitirlas
- La necesidad de una colección se deriva de las multiplicidades

Inclusión de Colecciones (2)

- Ejemplo



Un ATM tendrá asociado:
- Una sola transacción actual
- Una colección de transacciones finalizadas

Diseño de la Estructura
Errores Comunes

- No incluir las dependencias existentes
- Omitir la definición de los datatypes
- No incluir interfaces, controladores ni fábricas
- Sobrecargar el diagrama con operaciones omitibles (create, set, etc.)
- Incluir colecciones como clases innecesariamente