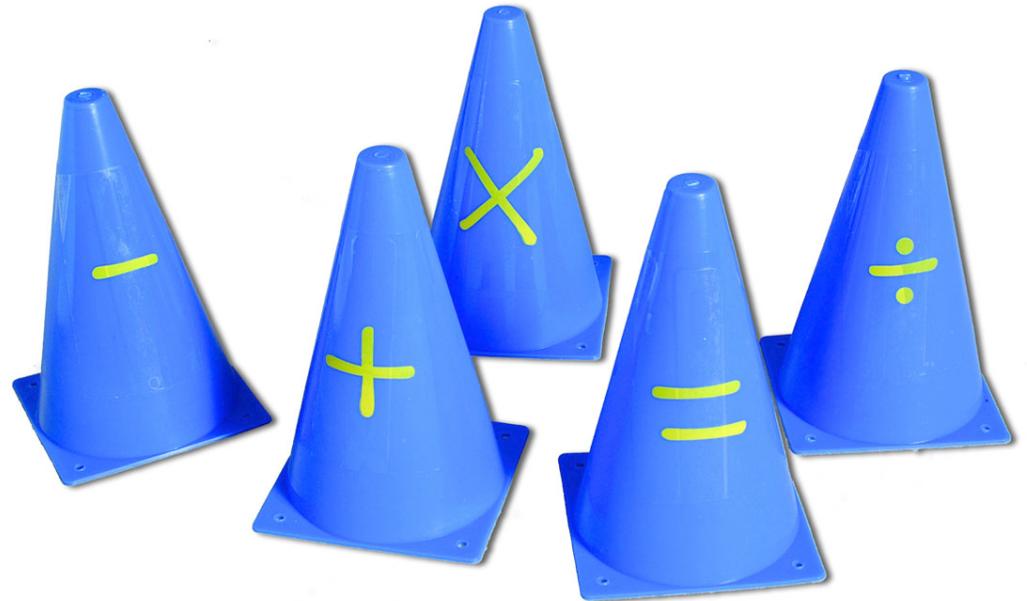


# La Notación

---

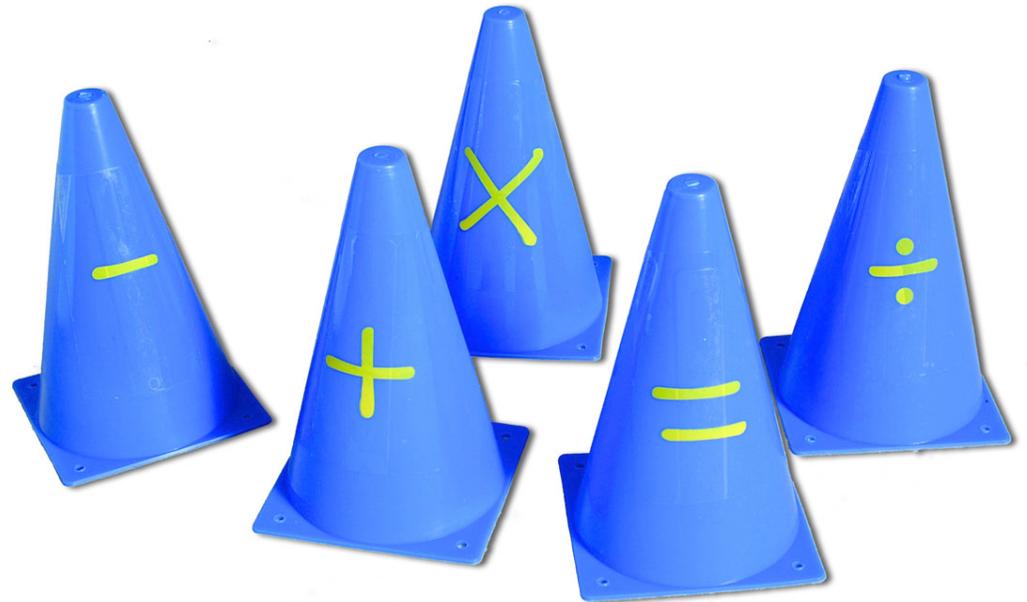
- Contenido
  - Elementos de la Notación
  - Diagramas de Clases
  - Diagramas de Objetos
  - Diagramas de Interacción o Secuencia
  - Diagrama de Estados



# La Notación

---

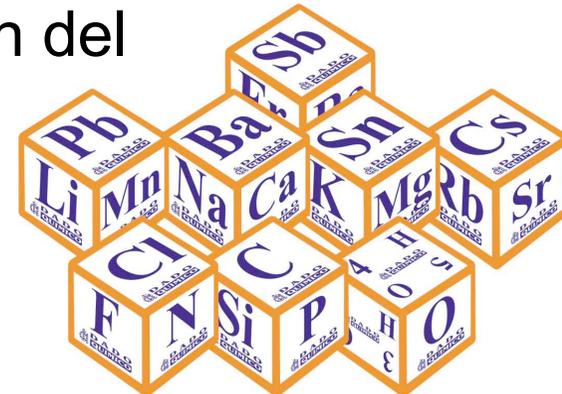
- Contenido
  - **Elementos de la Notación**
  - Diagramas de Clases
  - Diagramas de Objetos
  - Diagramas de Interacción o Secuencia
  - Diagrama de Estados



# Elementos de la Notación (1)

---

- Si se observa el trabajo de cualquier ingeniero, químico, arquitecto o cualquier otra disciplina, se advertirá pronto que el único y exclusivo lugar en el que se concibe un sistema es en la mente del diseñador.
- Puesto que este diseño no permanece en el tiempo, a menudo se captura mediante medios tan avanzados como pizarras, servilletas o el reverso de los envoltorios.
- La acción de dibujar un diagrama no constituye análisis ni diseño; un diagrama se limita a capturar una descripción del comportamiento del sistema.



# Elementos de la Notación (2)

---

- No obstante, el disponer de una notación expresiva y bien definida es importante para el proceso de desarrollo de software ya que:
  - Se posibilita a la persona a describir un universo y comunicar estas decisiones entonces a otros en forma no ambigua.
  - Una buena notación libera al cerebro de todo trabajo innecesario para concentrarse en problemas más avanzados.
  - Una notación expresiva hace posible eliminar la comprobación de consistencia y corrección de las decisiones adoptadas, ya que pueden utilizarse herramientas automáticas (CASE).



# Elementos de la Notación (3)

---

- Es imposible capturar todos los detalles sutiles de un sistema de software complejo en una sola vista. Uno debe comprender:
  - La estructura taxonómica de las clases.
  - Los mecanismos de herencia utilizados.
  - Los comportamientos individuales de los objetos.
  - El comportamiento dinámico del sistema en su conjunto.
- Durante el análisis deben plantearse las siguientes cuestiones:
  - ¿Cuál es el comportamiento que se desea del sistema?
  - ¿Cuáles son las misiones y responsabilidades de los objetos que llevan a cabo este comportamiento?
  - ¿Qué clases existen y cómo se relacionan estas clases?
  - ¿Qué mecanismos se utilizan para regular la forma en que los objetos colaboran entre sí?



# Elementos de la Notación (4)

---

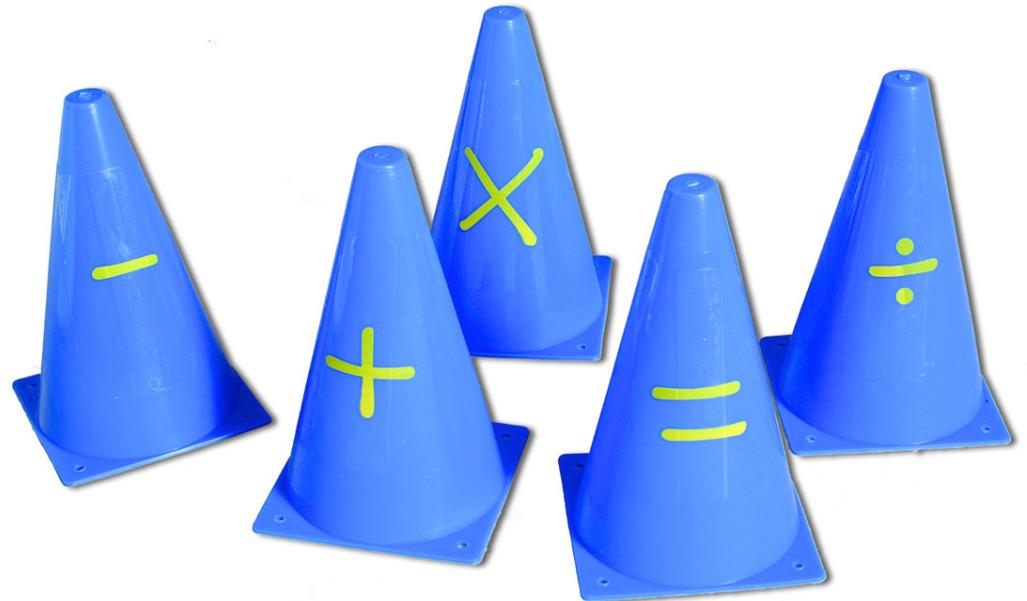
- Dado el soporte automático para cualquier notación, una de las cosas que las herramientas pueden hacer es ayudar a los malos diseñadores a crear diseños horribles mucho más rápido de lo que jamás han podido.
- Los grandes diseños provienen de los grandes diseñadores, no de las grandes herramientas; las herramientas son tan sólo eso: herramientas.
- Las herramientas simplemente fortalecen al individuo, ejecutando tareas de control de consistencia y corrección, liberándolo para concentrarse sobre los aspectos verdaderamente creativos del análisis o el diseño.
- Los diagramas de clases, de objetos, de transición de estados y de interacción son suficientes para expresar la semántica de un gran porcentaje de problemas de análisis.



# La Notación

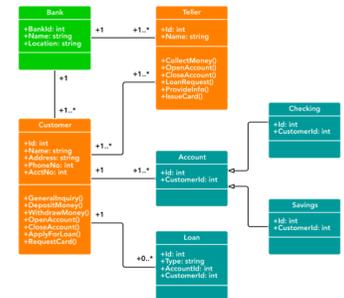
---

- Contenido
  - Elementos de la Notación
  - **Diagramas de Clases**
  - Diagramas de Objetos
  - Diagramas de Interacción o Secuencia
  - Diagrama de Estados

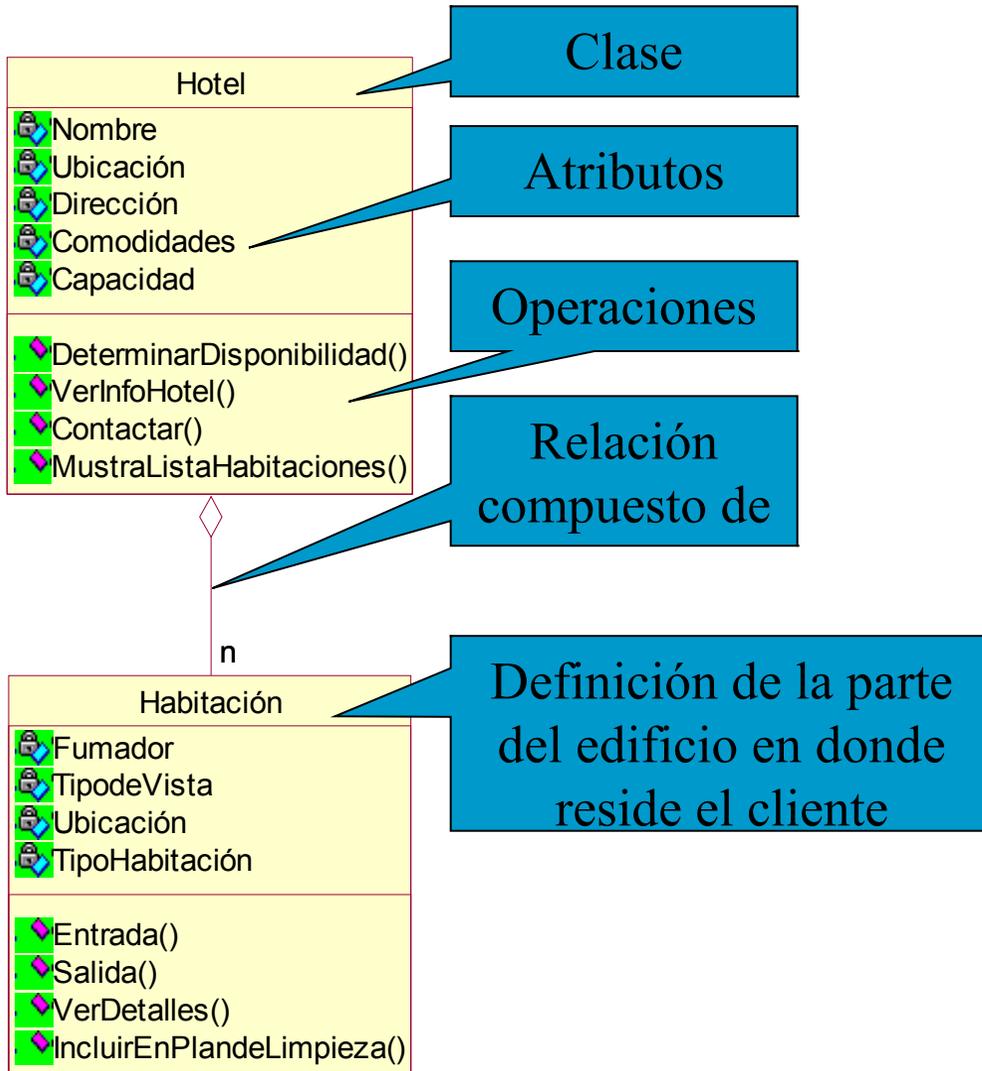


# Diagramas de Clases (1)

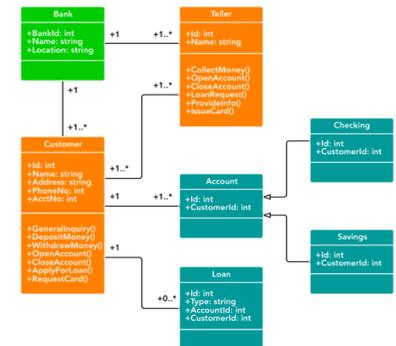
- Se utiliza un diagrama de clases para mostrar la existencia de clases y sus relaciones en la visión lógica del sistema.
- Un solo diagrama de clases representa una vista de la estructura de clases de un sistema.
- Durante el análisis, se utilizan diagramas de clases para indicar las misiones y responsabilidades comunes de las entidades que caracterizan el comportamiento de un sistema.
- Los dos elementos esenciales en un diagrama de clases son las clases y sus relaciones básicas.



# Diagramas de Clases (2)



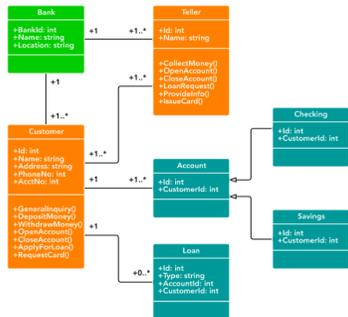
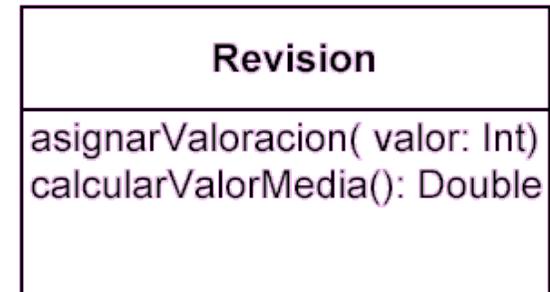
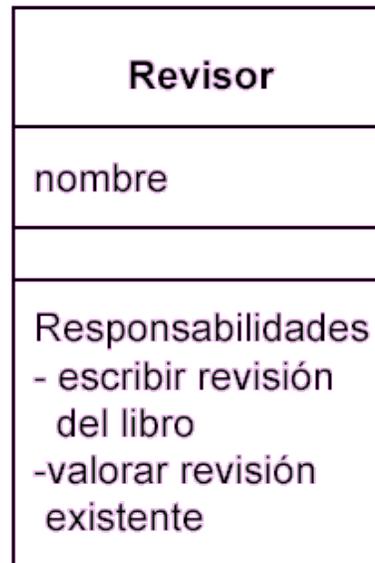
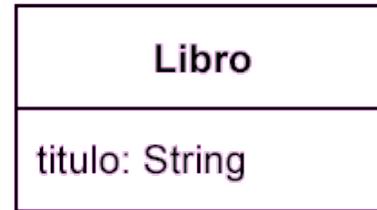
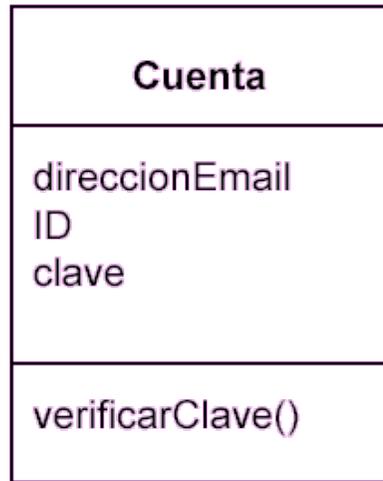
- Una clase es una plantilla que define una colección de cosas o conceptos con las mismas características.
- A un objeto que pertenece a una determinada clase se lo denomina **instancia** de una clase.



# Diagramas de Clases (2)

## Distintas formas de representar clases:

- Los nombres de las clases y de los atributos son nombres simples o frases nominales y los de las operaciones son verbos simples.

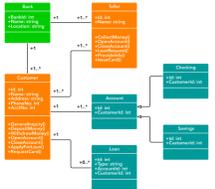


# Diagramas de Clases (3)

---

## Nombres:

- Se requiere un nombre para cada clase;
- Si el nombre es particularmente largo, puede abreviarse éste o bien utilizar un icono mayor.
- Todo nombre de clase debe ser único para la categoría de clases que engloba a esta.

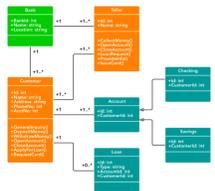


## Atributos:

- En ciertos diagramas de clases es útil exponer algunos de los atributos y operaciones asociados con una clase.
- El término “algunos” obedece a que para todas, excepto la clase más trivial, es impropio y en realidad innecesario mostrar todos esos miembros en un diagrama, incluso cuando se usa un icono rectangular.

# Diagramas de Clases (4)

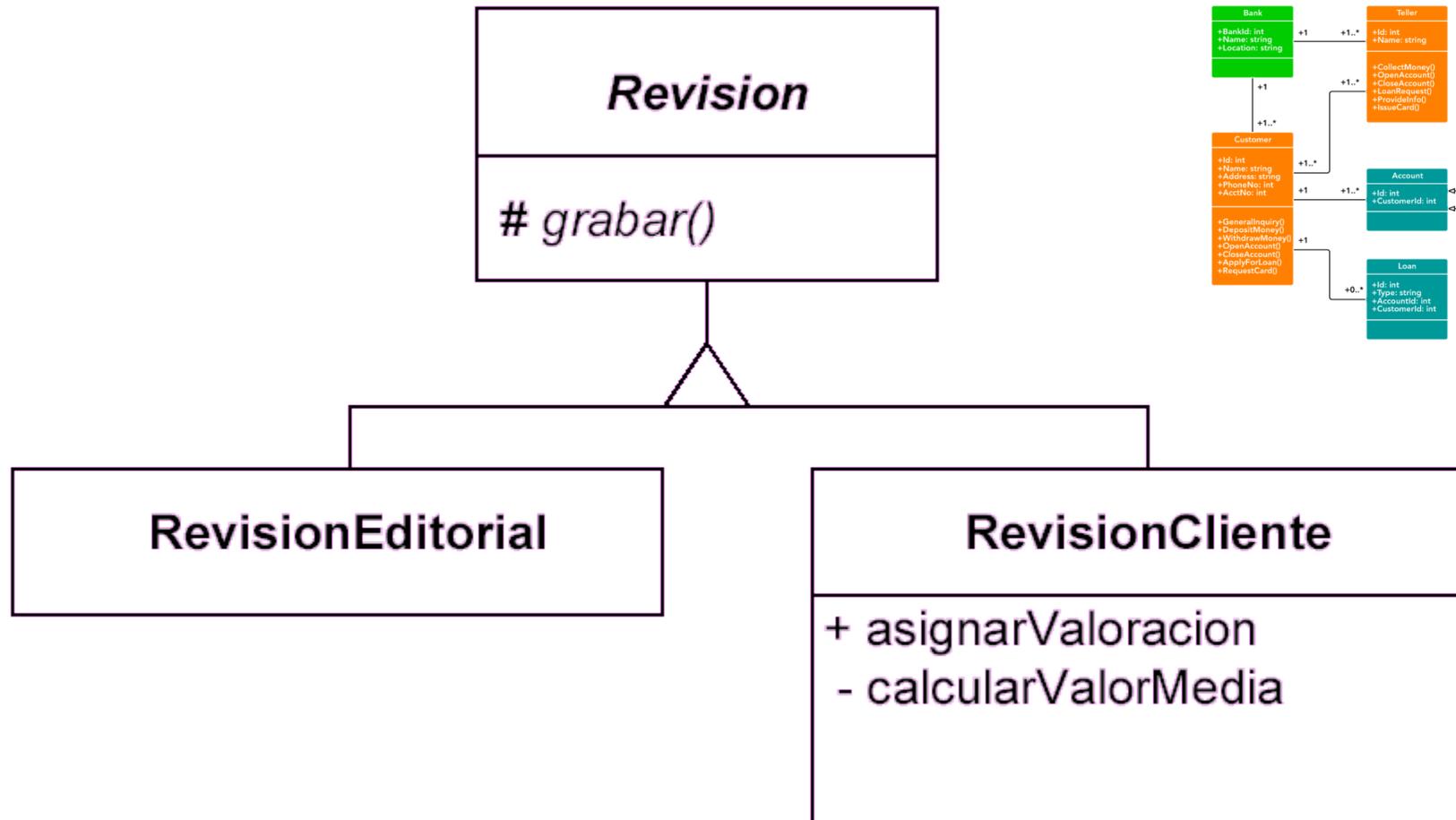
- UML ofrece una serie de elementos para especificar detalles para los atributos y las operaciones:
  - Visibilidad:** el encapsulado es el principio para ocultar datos (data hiding). Desde fuera de un objeto sólo se pueden manipular los datos por medio de llamadas a las operaciones (métodos) del objeto. La visibilidad puede ser:
    - De paquete:** indicada como [~] significa que los objetos pertenecientes a cualquier clase del mismo paquete al que pertenece la clase en cuestión puede verlo y utilizarlo.
    - Público:** (+) significa que los objetos pertenecientes a cualquier clase pueden utilizar el atributo u operación.
    - Protegido:** (#) significa que sólo los objetos de las subclases de la clase a la que pertenece el atributo u operación pueden utilizarlo.
    - Privado:** (-) significa que sólo se puede usar en los objetos donde está definido el atributo u operación.





# Diagramas de Clases (6)

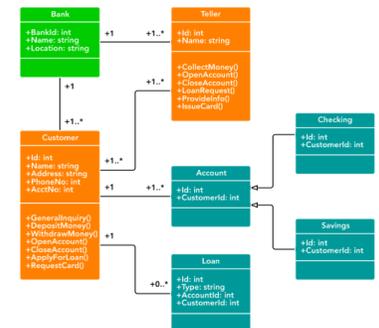
- La clase ***Revisión*** y su operación ***grabar*** son abstractas y se las diferencia porque los nombres están en cursiva. Una clase abstracta no puede tener instancias.



# Diagramas de Clases (7)

## Atributos:

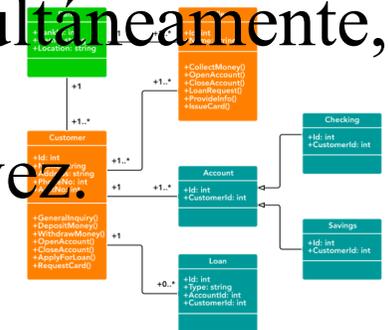
- En este sentido, los atributos y operaciones mostrados representan una visión abreviada de la especificación completa de la clase, que sirve como el único punto de declaración para todos sus miembros.
- Si se necesita mostrar muchos de éstos, se puede agrandar el icono de clase; si se decide no mostrar tales miembros, se puede eliminar la línea de separación y mostrar solamente el nombre de la clase.





# Diagramas de Clases (9)

- Una propiedad que se puede añadir a una declaración de operación es `IsQuery`, que indica que la operación no cambia el valor de ningún atributo.
- Hay tres propiedades relacionadas con la concurrencia, que tienen que ver con la manera de que un método que implementa una operación responde a los múltiples hilos de actividad:
  - **concurrent:** pueden entrar múltiples llamadas al método provenientes de otros tantos hilos y todas ellas se atienden concurrentemente.
  - **guarded:** pueden llegar varias llamadas simultáneamente, pero sólo puede atenderse una por vez.
  - **sequential:** sólo puede llegar una llamada por vez.

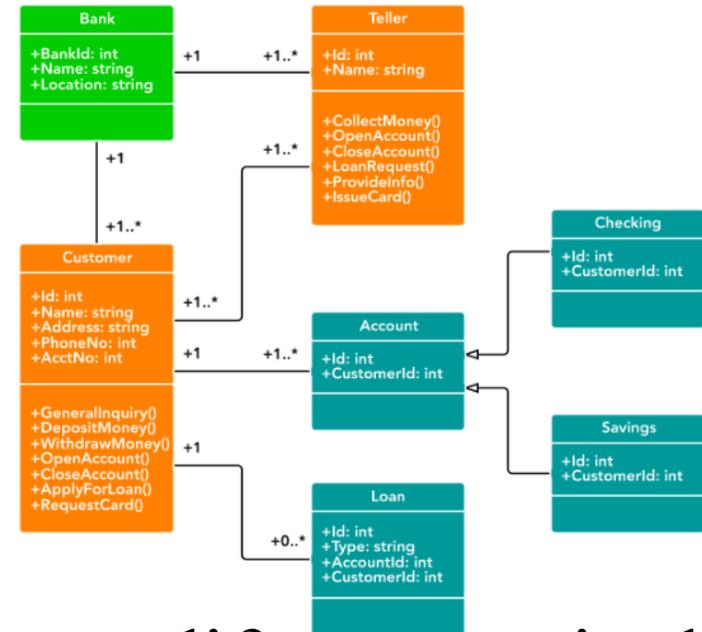




# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias



- Los diagramas de clases se crean a diferentes niveles

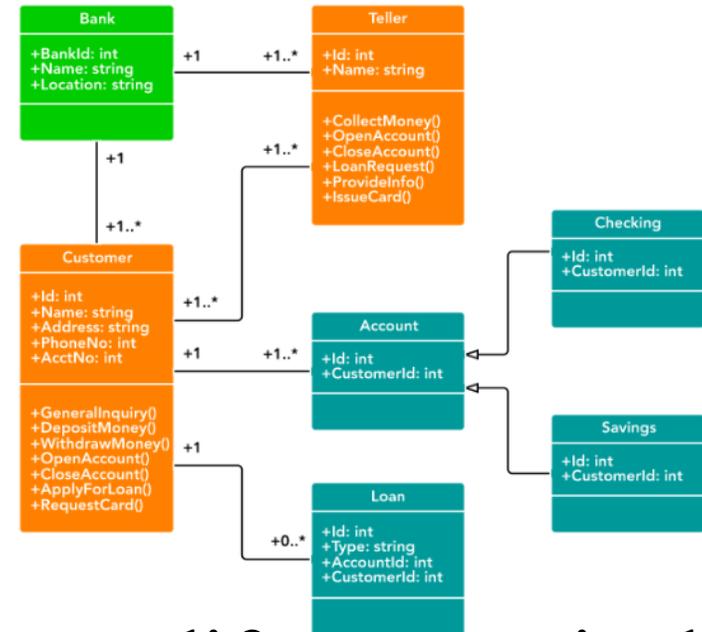
- Dominio
- Análisis

- Interfaces entre clases

# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias



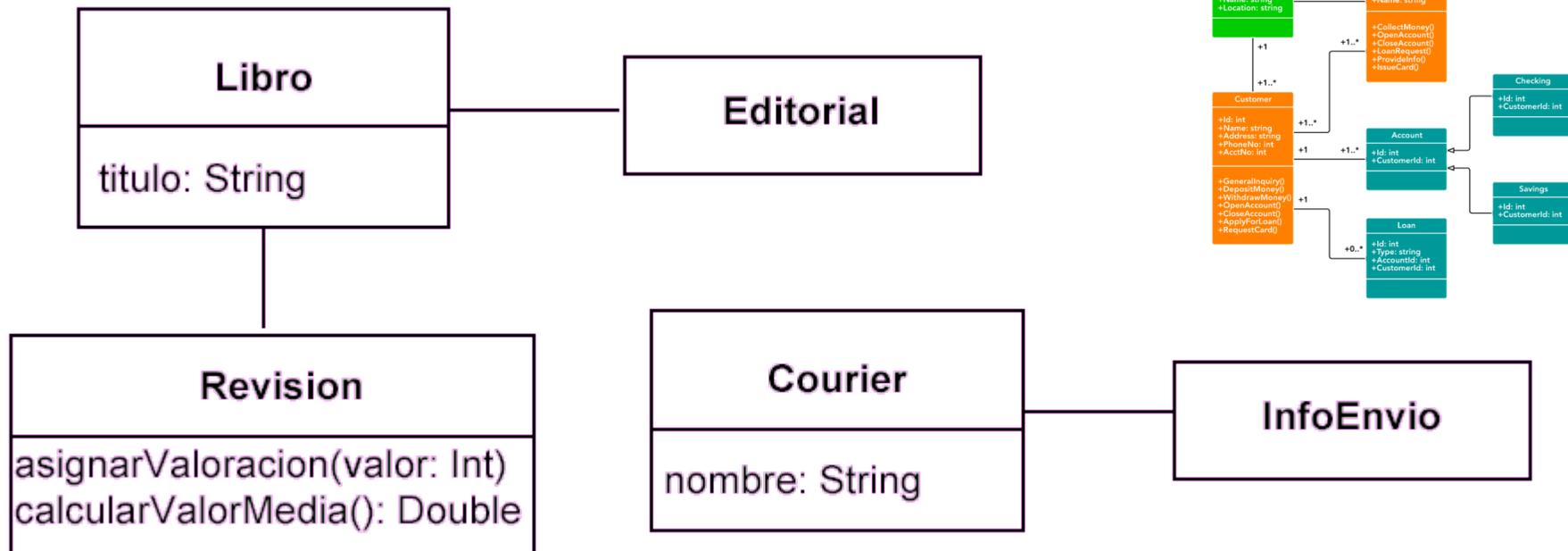
- Los diagramas de clases se crean a diferentes niveles

- Dominio
- Análisis

- Interfaces entre clases

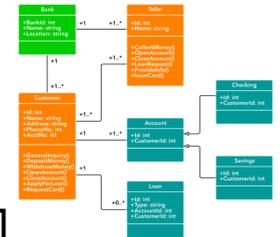
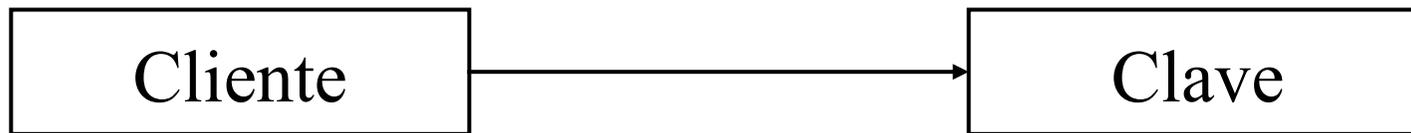
# Asociaciones (1)

- Una asociación es una conexión estructural simple entre clases. Las instancias de las clases implicadas en una asociación estarán probablemente comunicándose en el momento de ejecución.



# Asociaciones (2)

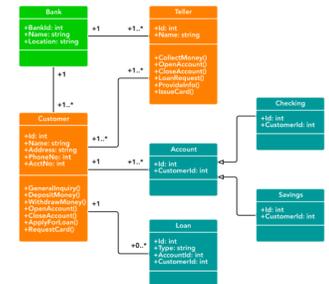
- Se asume que una asociación es bidireccional, es decir que se puede navegar desde cualquiera de las clases implicadas a la otra, pero es posible indicar que la navegación ocurrirá en una sola dirección.



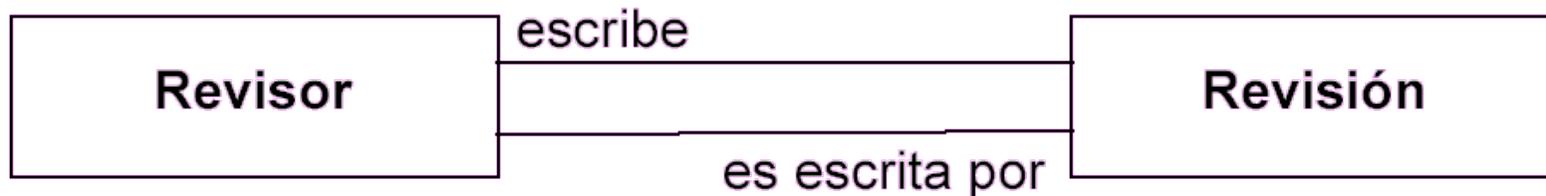
- Al establecer la direccionalidad de la asociación, indicamos que el *Cliente* puede acceder a su *Clave*, pero nadie puede usar la Clave para identificar a un Cliente.

# Asociaciones (3)

- Podemos agregar adiciones (detalles o adornos) a las asociaciones:
  - Un nombre que indica la naturaleza de la asociación y que puede estar acompañado de un triángulo que apunta en la dirección que debe leerse.

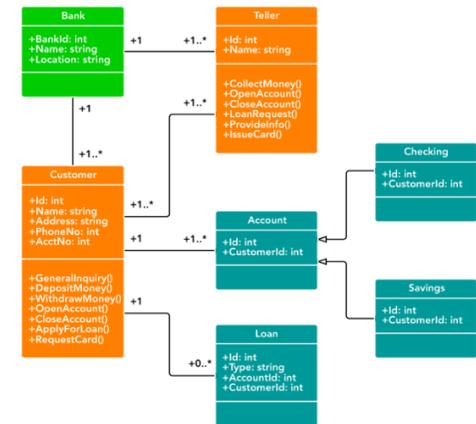
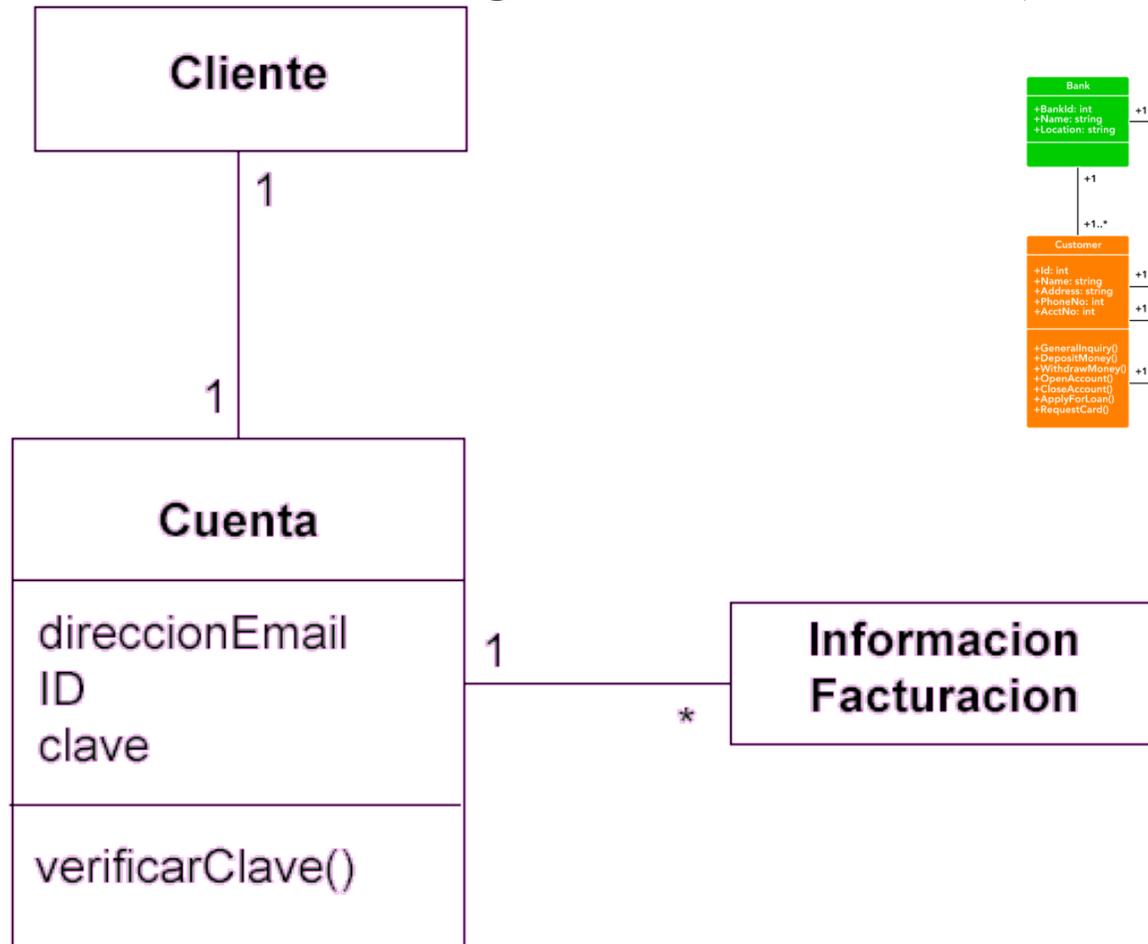


- La asociación también puede tener roles, que son las caras que presentan las clases a las demás.



# Asociaciones (4)

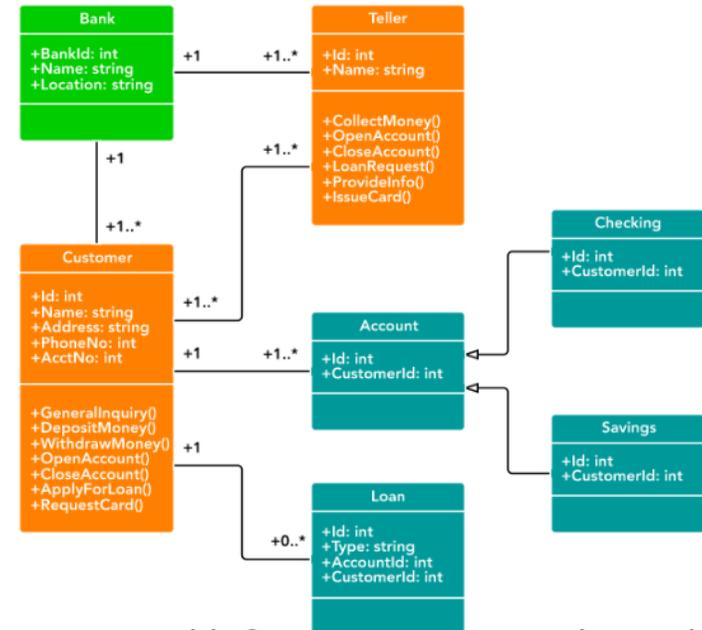
- Una asociación puede mostrar también la multiplicidad. Es igual a la cardinalidad en ingeniería de datos (salvo \* por N).



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- **Agregación**
- Composición
- Generalización
- Dependencias



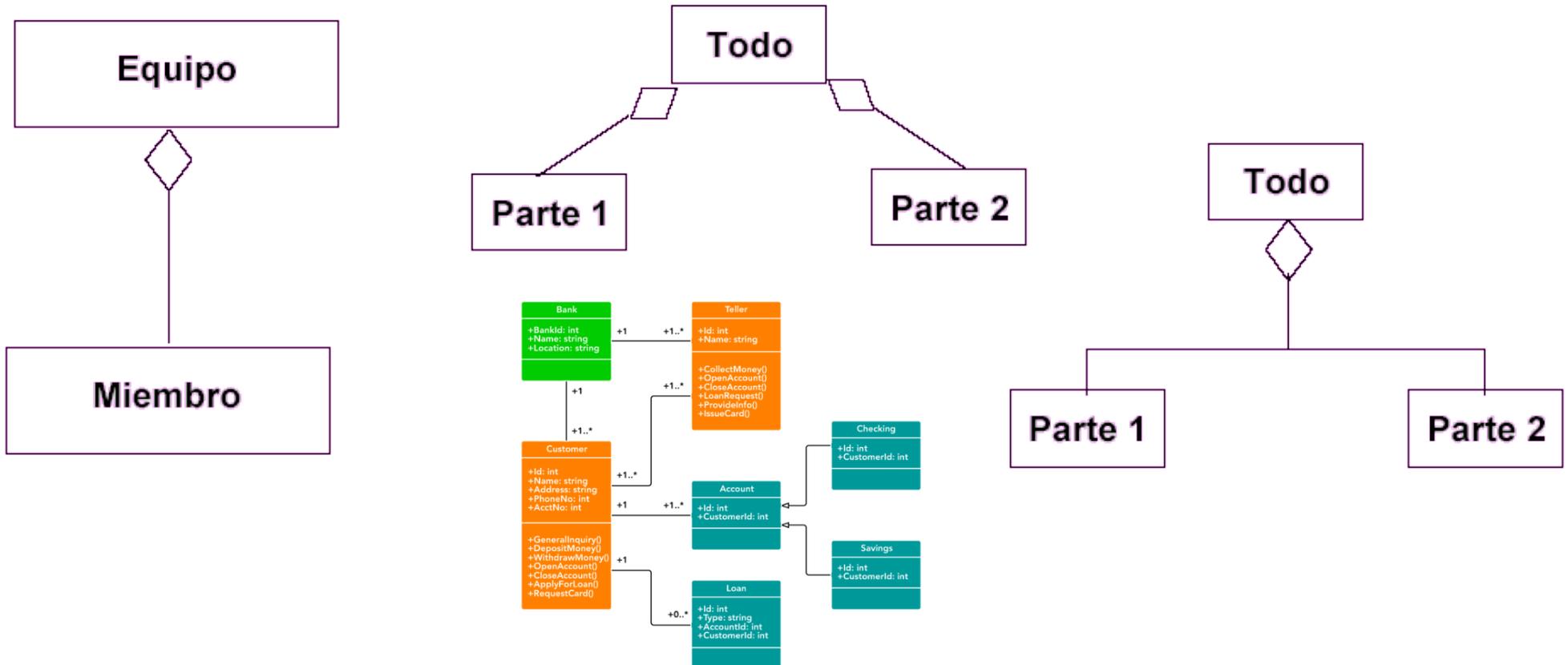
- Los diagramas de clases se crean a diferentes niveles

- Dominio
- Análisis

- Interfaces entre clases

# Agregación

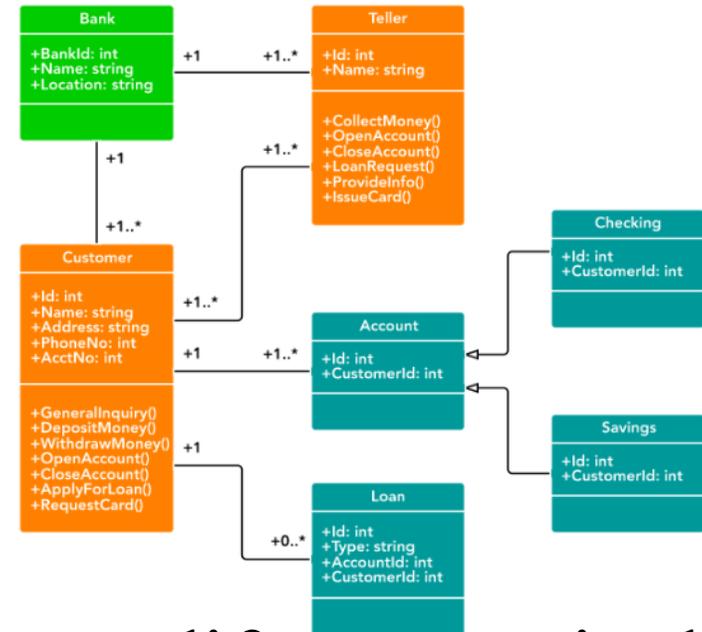
- Es una asociación especial, una relación del tipo “todo/parte” dentro de la cual una o más clases son partes de un conjunto.



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- **Composición**
- Generalización
- Dependencias



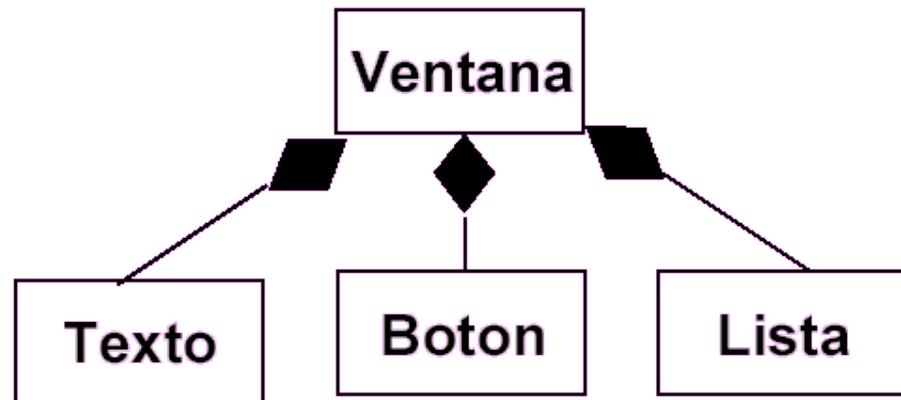
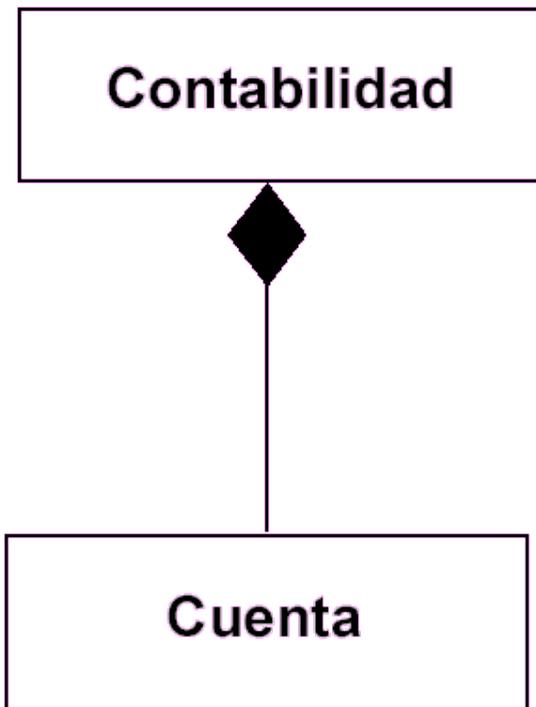
- Los diagramas de clases se crean a diferentes niveles

- Dominio
- Análisis

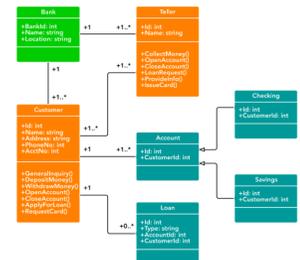
- Interfaces entre clases

# Composición

- La composición es una forma “fuerte” de agregación. Se diferencian en:
  - En la composición tanto el todo como las partes tienen el mismo ciclo de vida.
  - Un objeto puede pertenecer solamente a una composición.



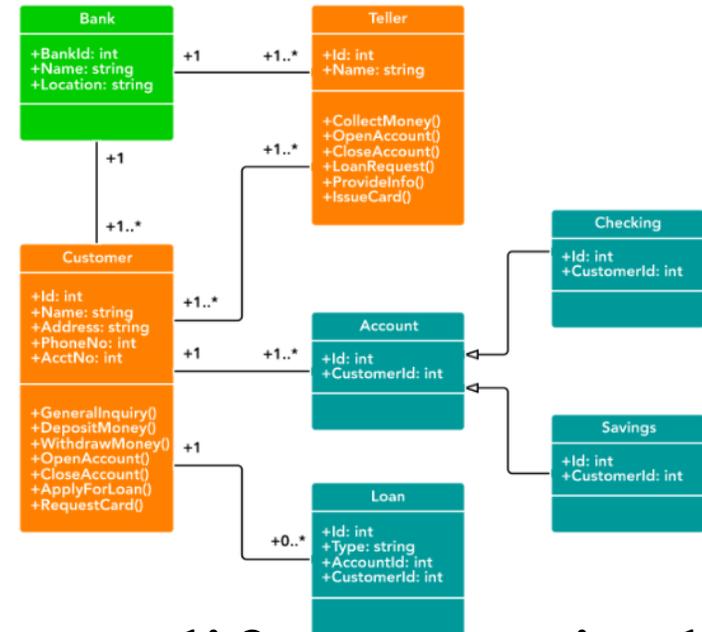
Contención Física



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- **Generalización**
- Dependencias



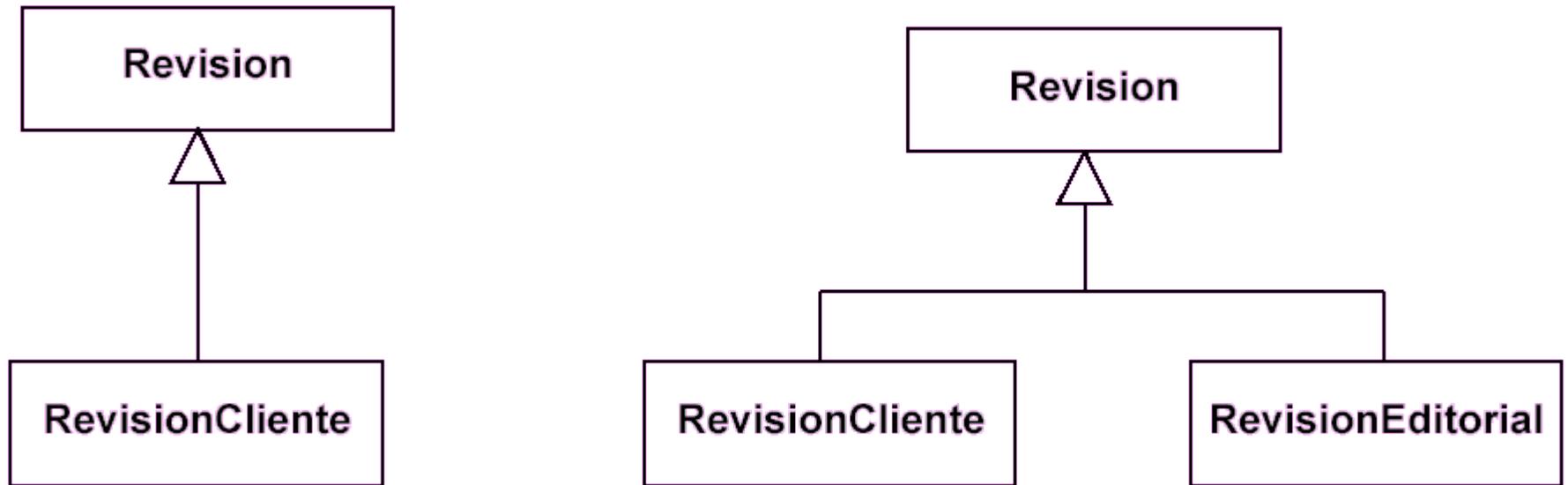
- Los diagramas de clases se crean a diferentes niveles

- Dominio
- Análisis

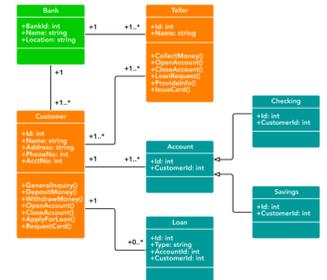
- Interfaces entre clases

# Generalización

- Una generalización/especialización se refiere a una relación entre una clase general (superclase o padre) y una versión más específica de dicha clase (subclase o hija).



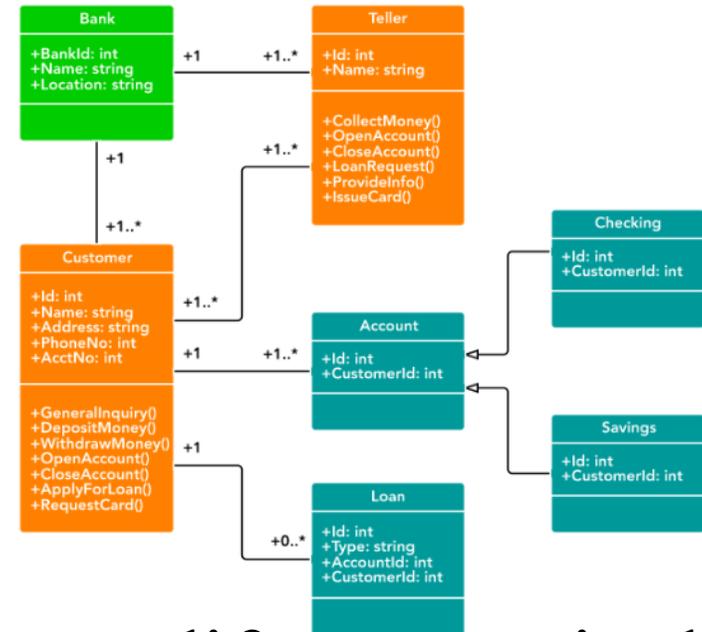
Herencia



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- **Dependencias**



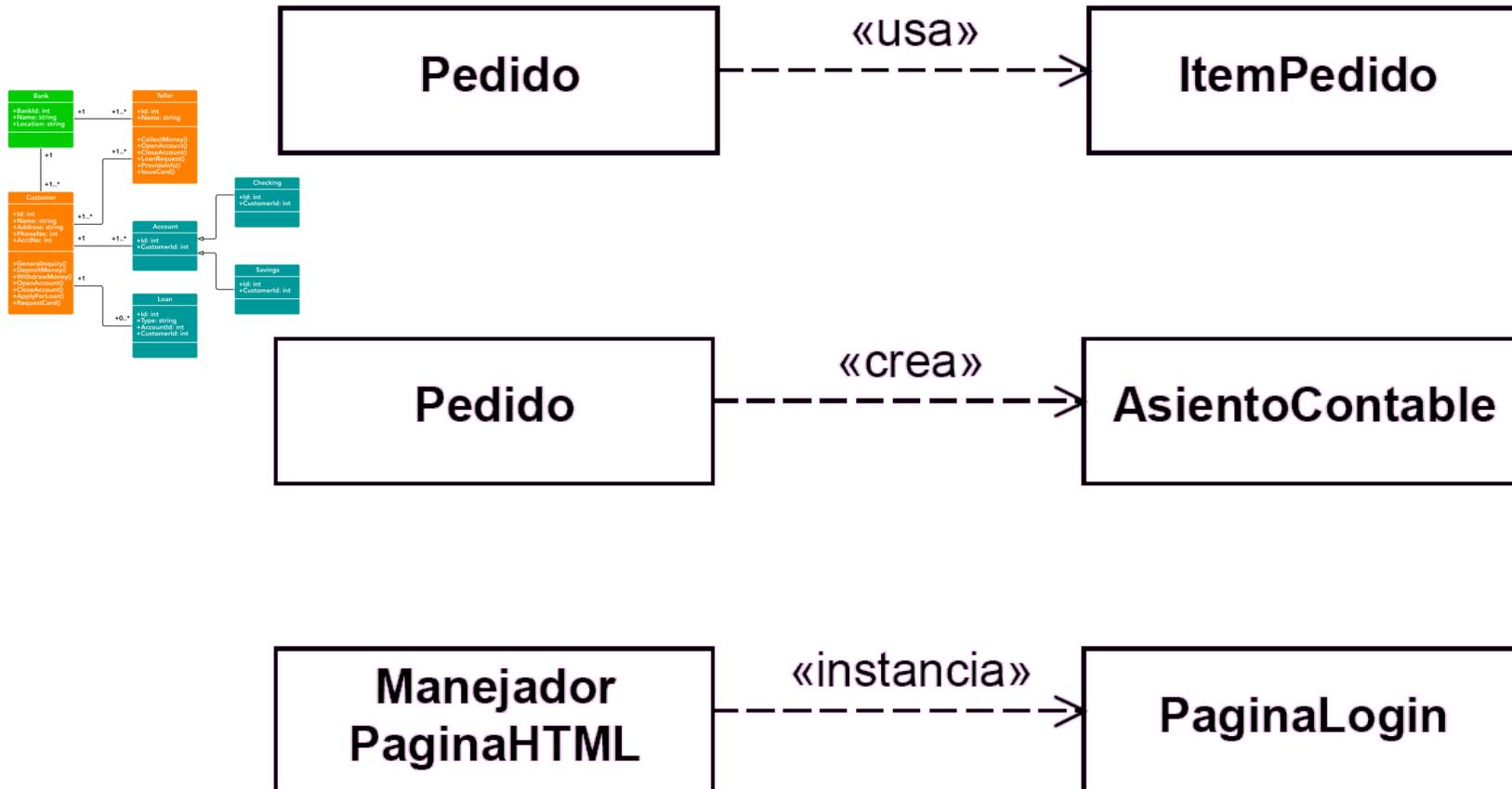
- Los diagramas de clases se crean a diferentes niveles

- Dominio
- Análisis

- Interfaces entre clases

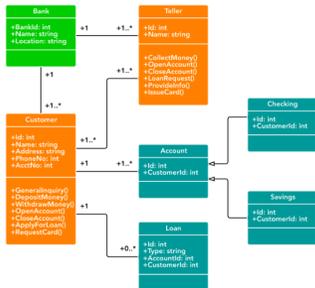
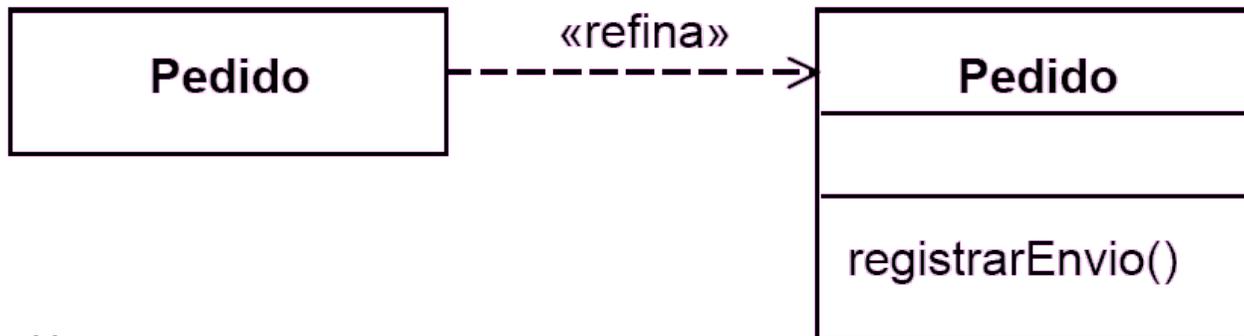
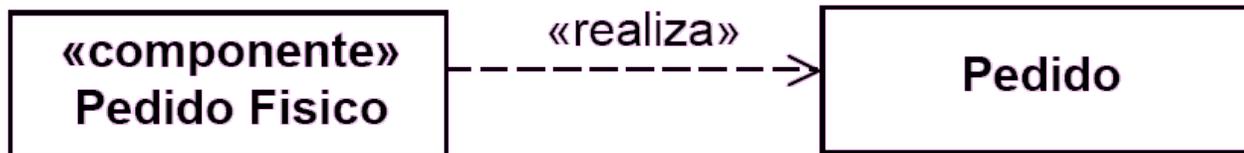
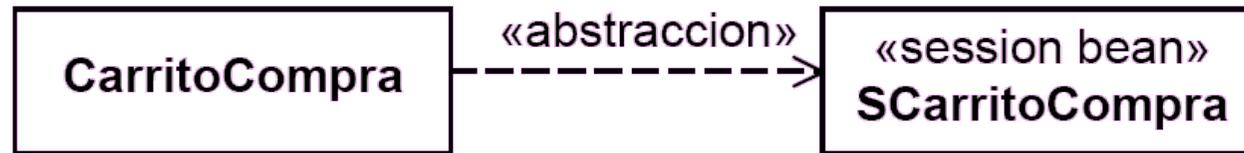
# Dependencias (1)

- Una dependencia es una relación de “uso” en la que un cambio en uno de los términos (por ejemplo, una clase) puede afectar a otro (otra clase).



# Dependencias (2)

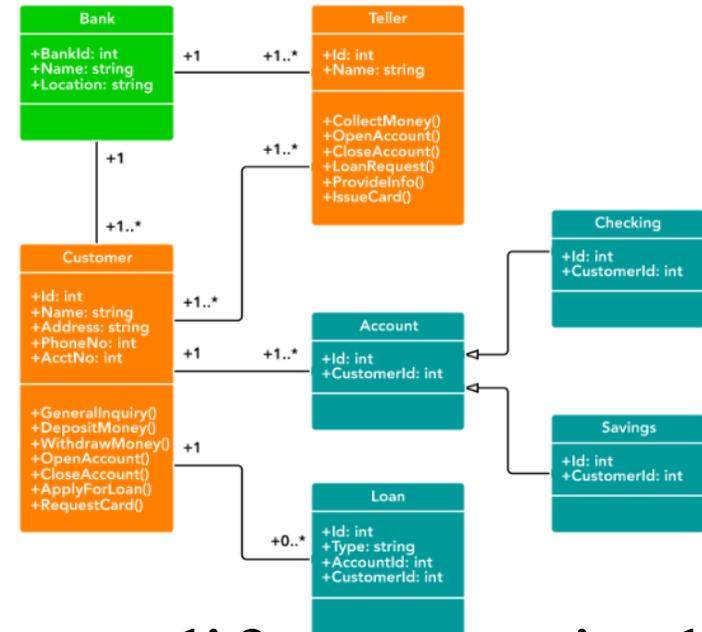
- Se usa también para dar trazabilidad, cuando se hace más concreto algo.



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias

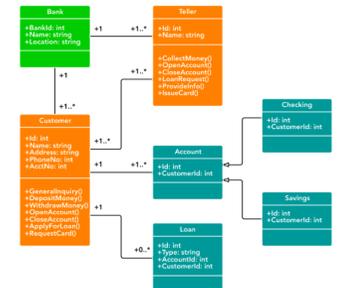
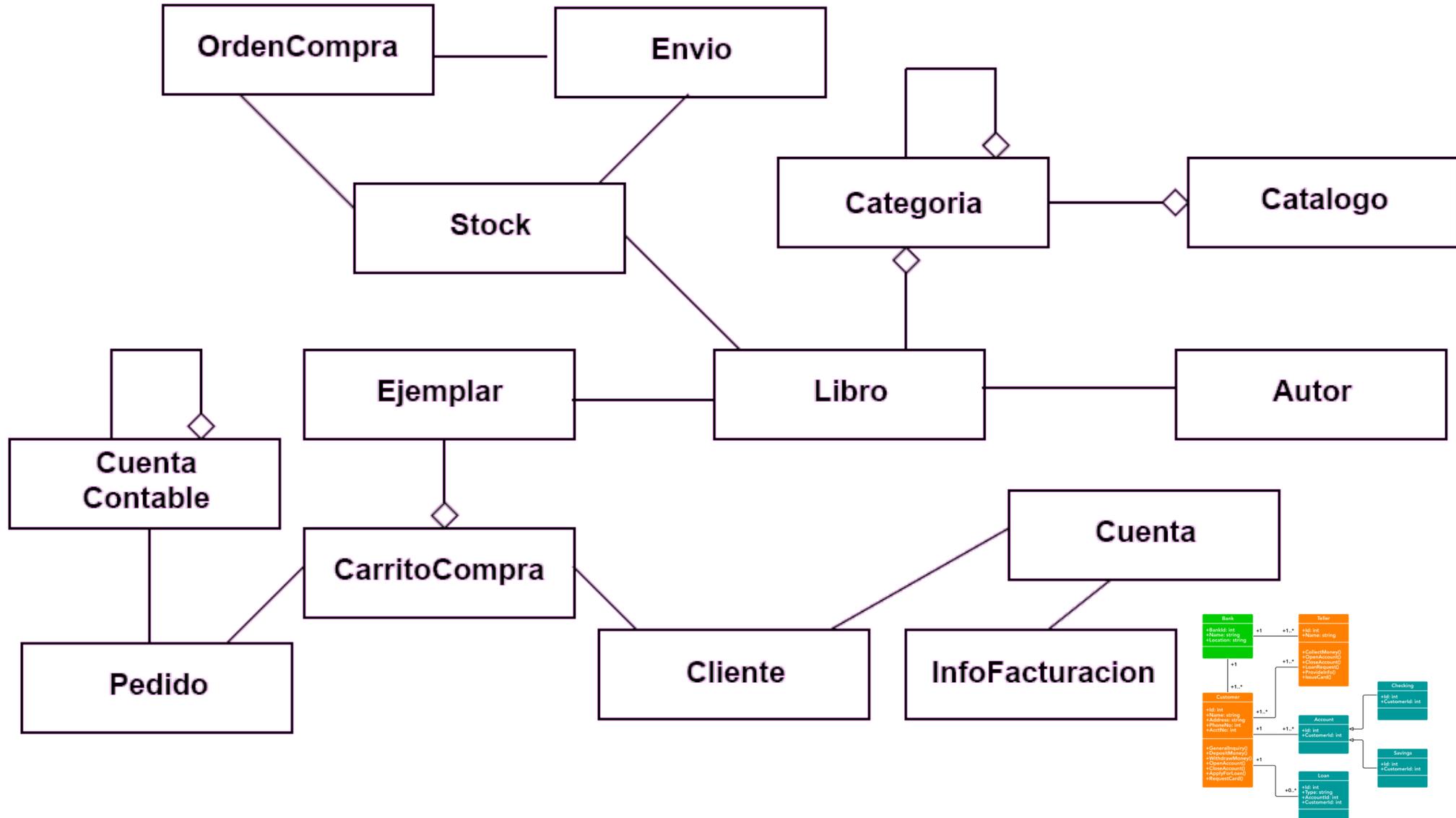


- Los diagramas de clases se crean a diferentes niveles

- **Dominio**
- Análisis

- Interfaces entre clases

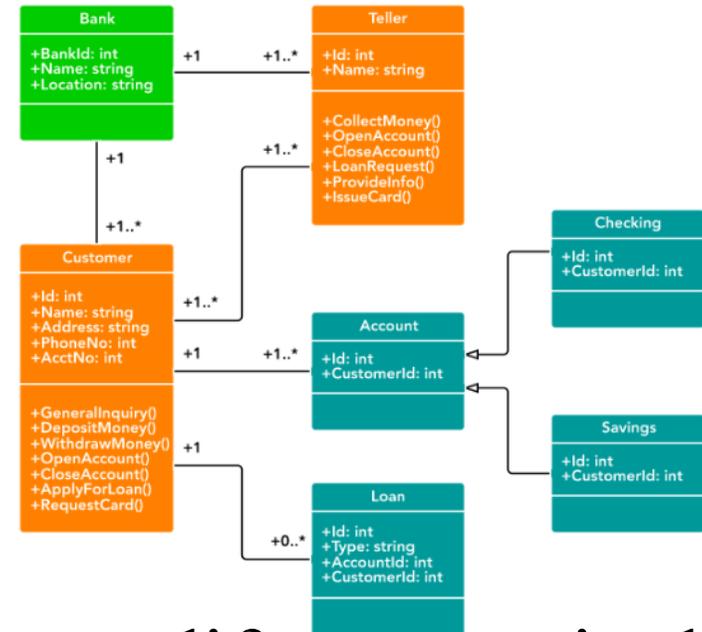
# Diagramas a Nivel de Dominio



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias

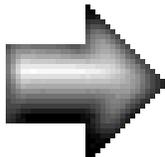
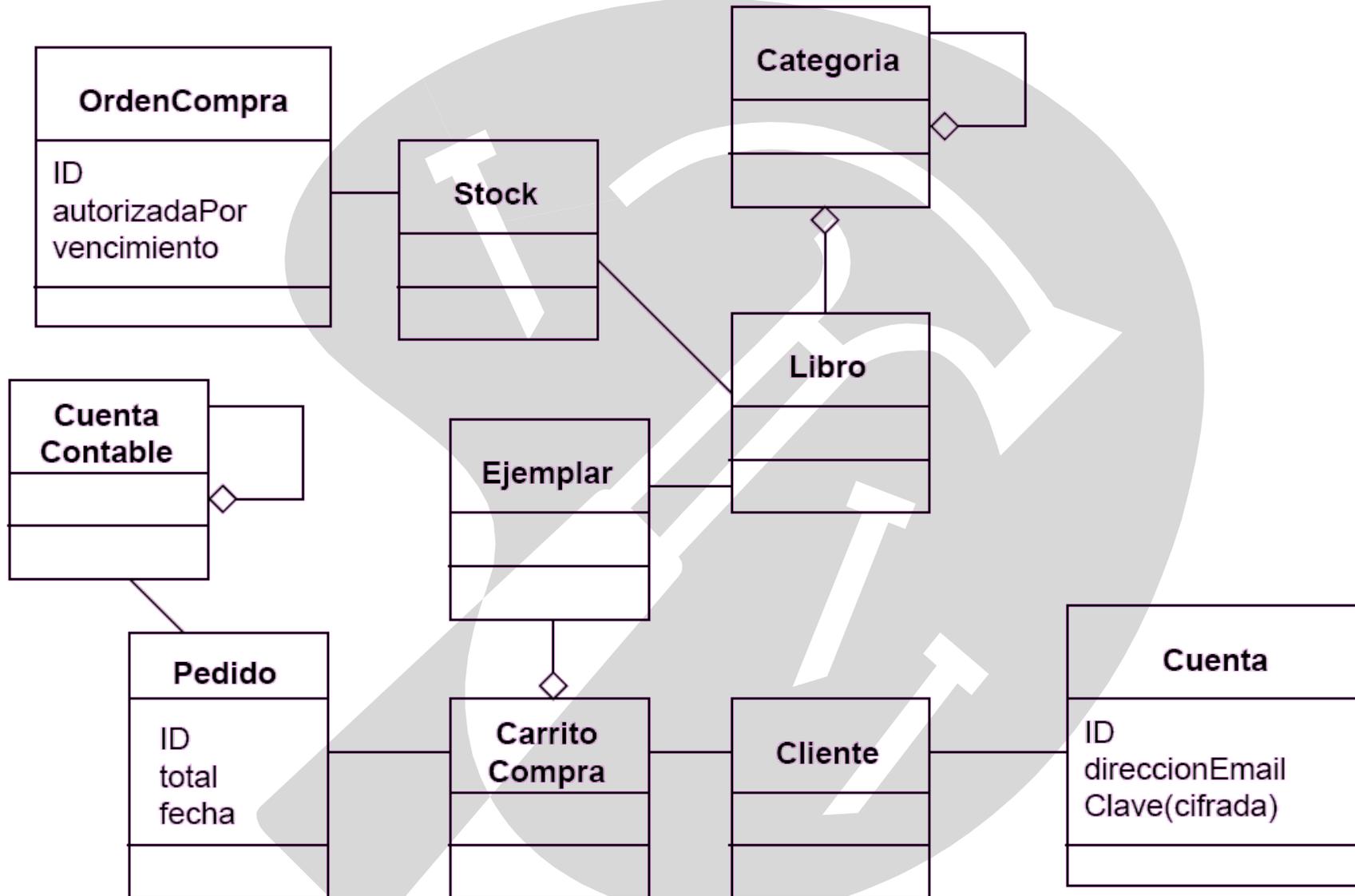


- Los diagramas de clases se crean a diferentes niveles

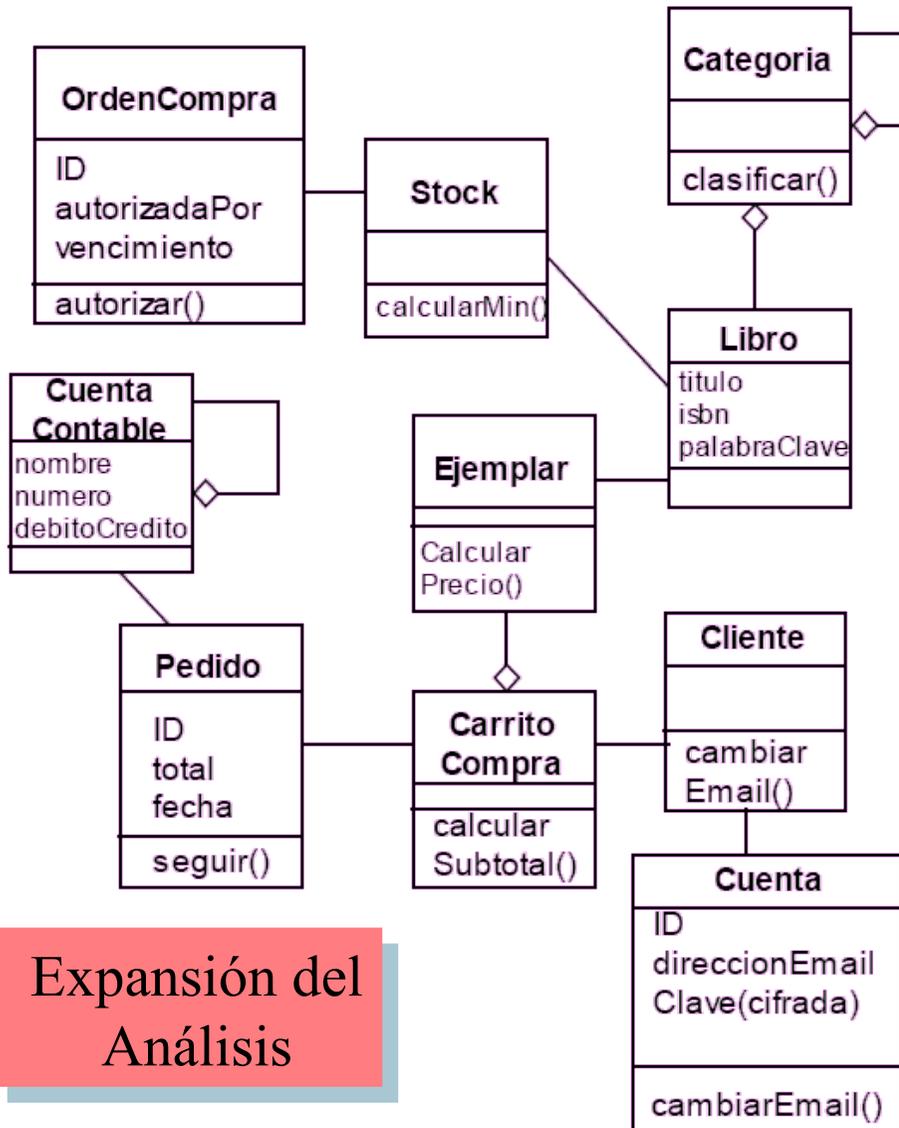
- Dominio
- **Análisis**

- Interfaces entre clases

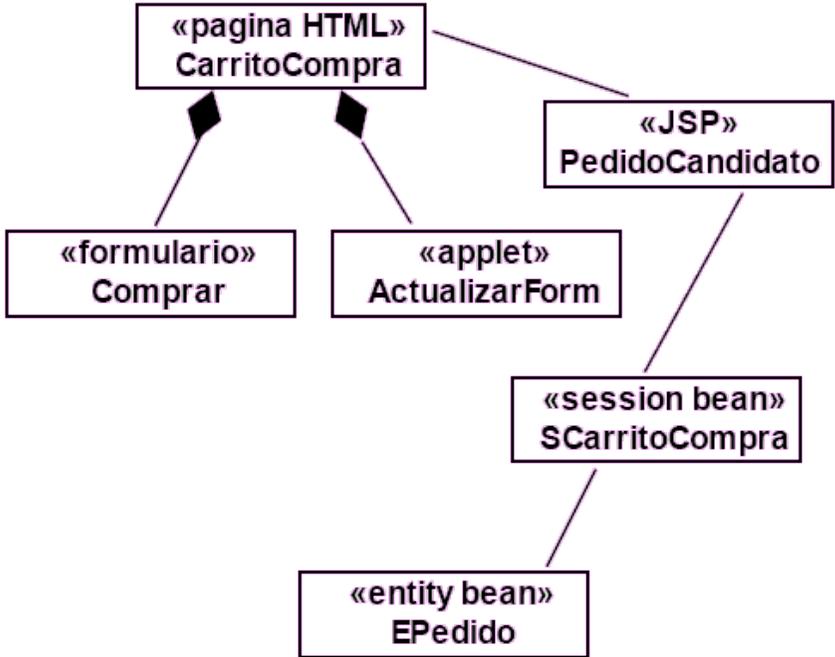
# Diagramas a Nivel de Análisis (1)



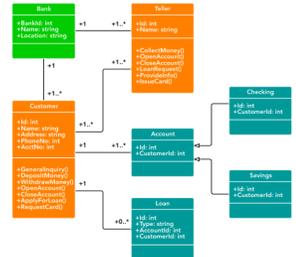
# Diagramas a Nivel de Análisis (2)



Expansión del Análisis



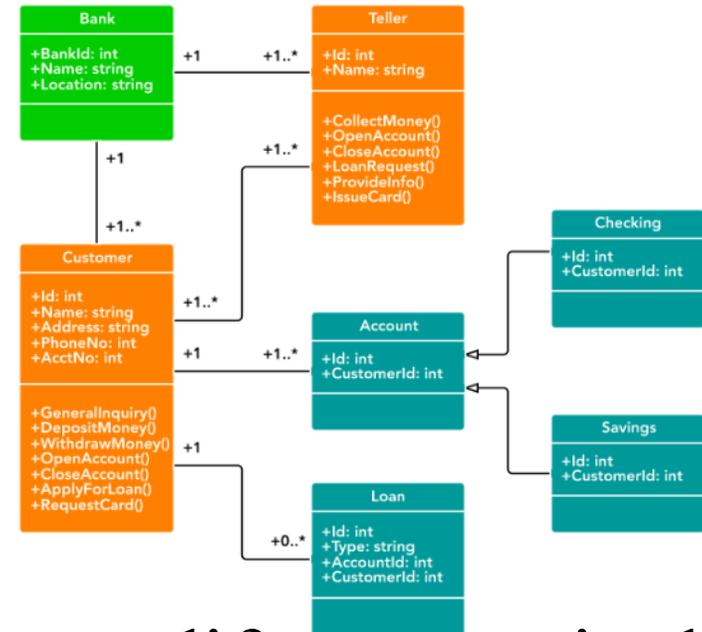
Clases de Diseño



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias



- Los diagramas de clases se crean a diferentes niveles

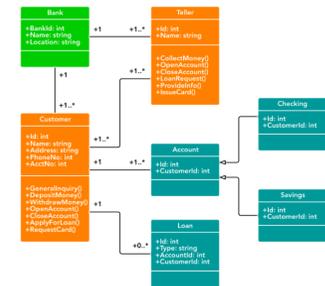
- Dominio
- Análisis

- Interfaces entre clases



# Interfaces (2)

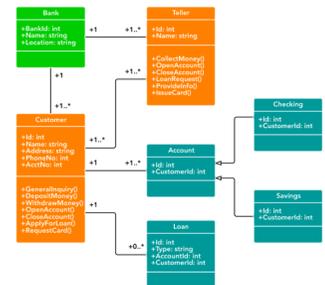
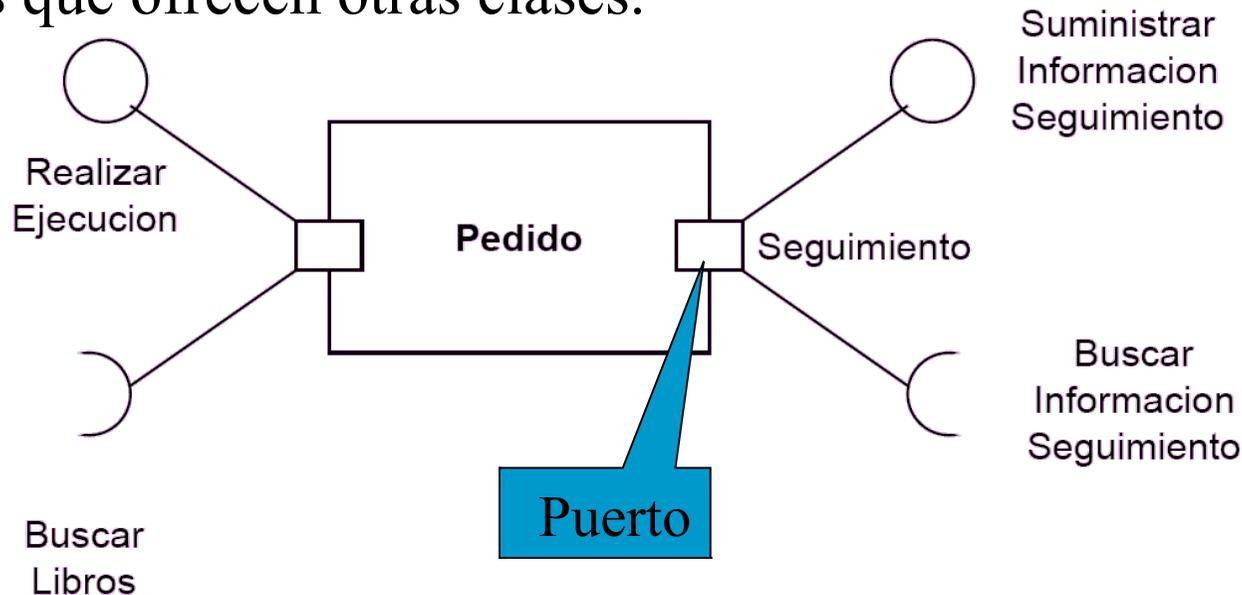
- Las interfaces requeridas son aquellas que necesita una clase para realizar su cometido. El símbolo usado para representarla es el semicírculo.
- Las instancias de la clase *Pedido* utilizan *BuscarLibros* para cumplimentar el pedido especificado, y *Buscar Información Seguimiento* si el cliente que realizó el pedido quiere seguir la historia del envío del pedido.



# Interfaces (3)

## Puertos y Conectores

- Un puerto especifica un punto claro de interacción entre una clase y su entorno. Los puertos agrupan interfaces suministradas y/o interfaces requeridas.
- Por un lado sirven como puntos focales a través de los cuales se pueden hacer peticiones para invocar las operaciones que ofrece la clase. Por otro, sirven como puente para las llamadas que hace la clase a operaciones que ofrecen otras clases.



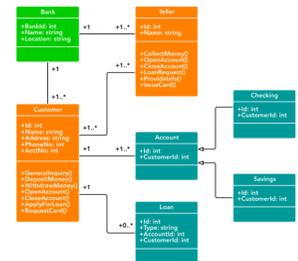
# Interfaces (4)

## Paquetes

- Un paquete es un agrupamiento de pieza del modelo y son muy útiles para gestionar modelos. No necesariamente implican algo concreto en la implementación.
- Se representan en UML como una carpeta con lengüeta y hay tres variantes:

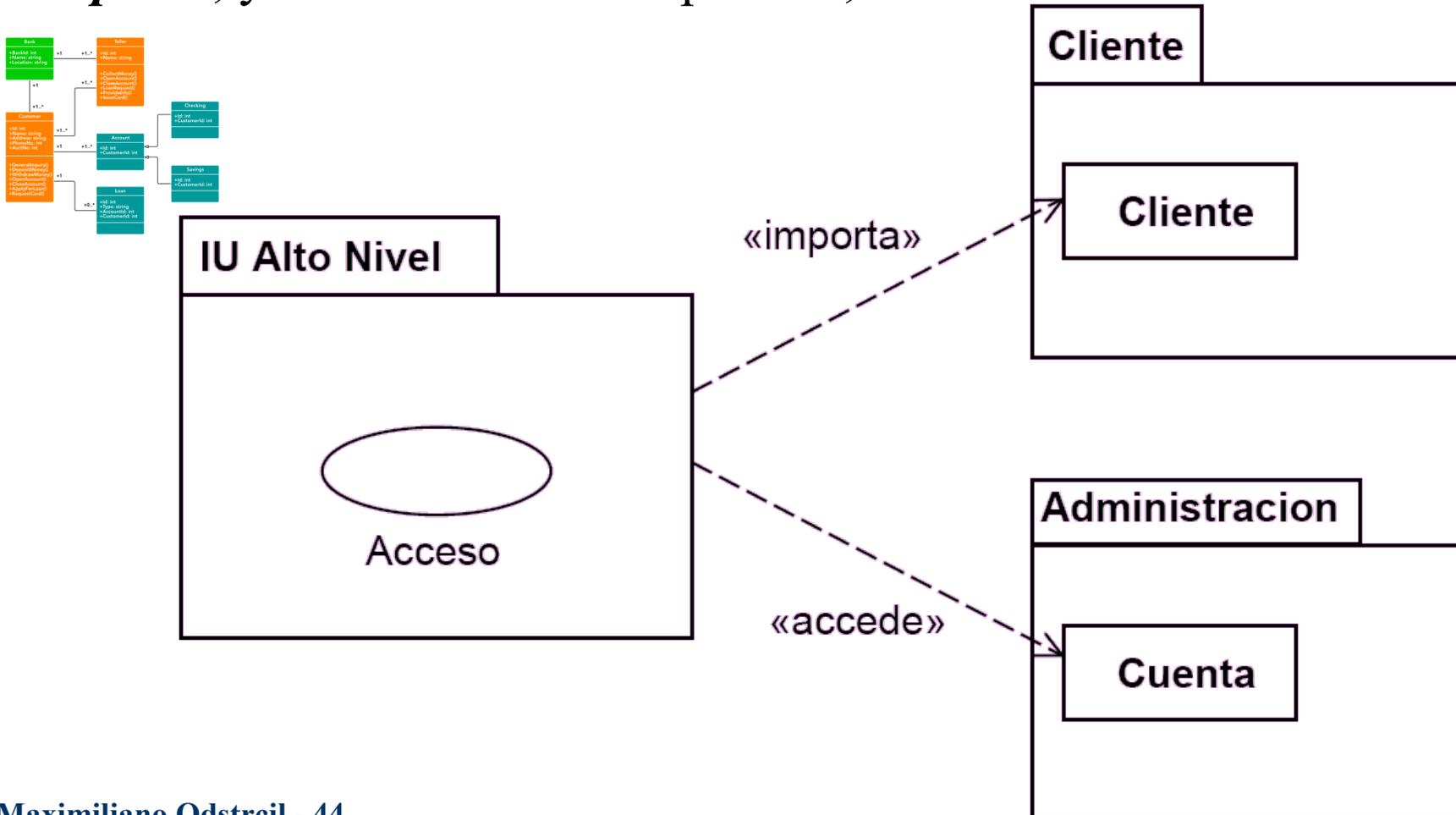


El nombre del paquete aparece en el cuerpo de la carpeta y el contenido está oculto



# Interfaces (5)

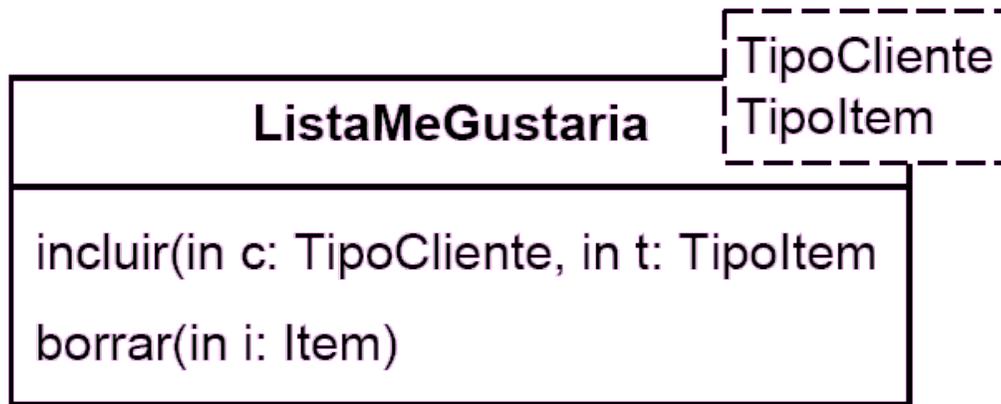
- Un paquete puede añadir elementos externos del modelo por referencia. Si la visibilidad de un elemento dado es pública, el paquete realiza un *importa*, y si la visibilidad es privada, realiza un *accede*.



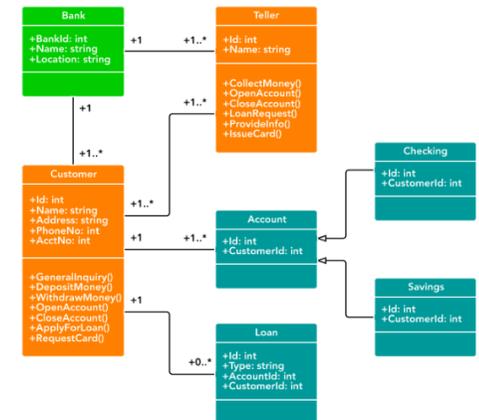
# Interfaces (6)

## Plantillas (Template, Instanciación)

- Una plantilla es un descriptor para un elemento de modelado que tiene uno o más parámetros formales que pueden usarse para crear una familia de elementos de modelos relacionados



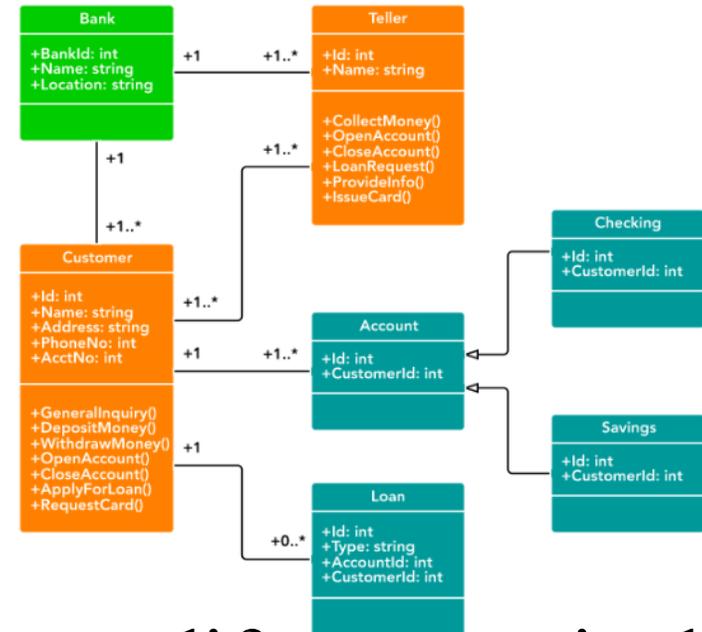
«ligar» <TipoCliente -> Vacaciones, Tipoltem -> CD>



# Diagramas de Clases (11)

- Relaciones entre clases:

- Asociaciones
- Agregación
- Composición
- Generalización
- Dependencias



- Los diagramas de clases se crean a diferentes niveles

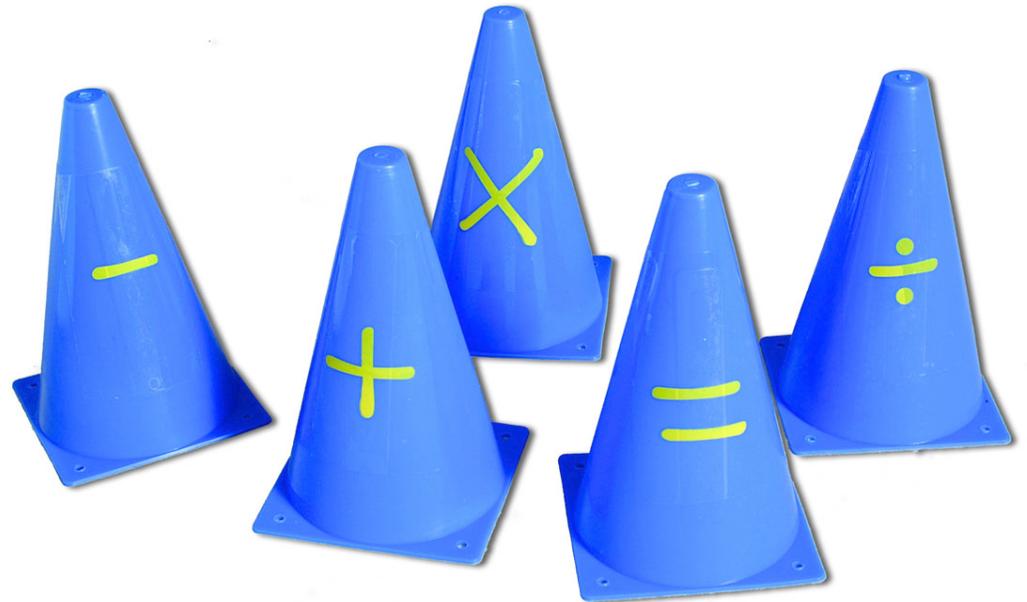
- Dominio
- Análisis

- Interfaces entre clases

# La Notación

---

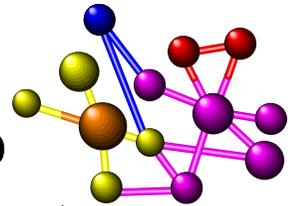
- Contenido
  - Elementos de la Notación
  - Diagramas de Clases
  - **Diagramas de Objetos**
  - Diagramas de Interacción o Secuencia
  - Diagrama de Estados



# Diagramas de Objetos (1)

---

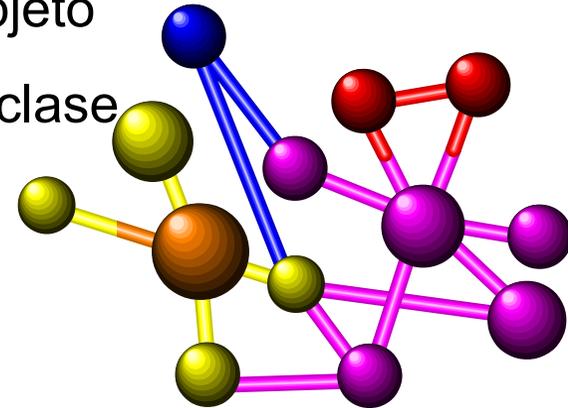
- Un diagrama de objetos se utiliza para mostrar la existencia de objetos y sus relaciones en el diseño lógico del sistema.
- Dicho de otro modo, un diagrama de objetos representa una instantánea en el tiempo de una corriente de eventos sobre una cierta configuración de objetos.
- Cada diagrama de objetos representa las interacciones o relaciones estructurales que pueden darse entre un conjunto determinado de instancias de clases, con diferencia de qué objetos de nombre concreto participan en la colaboración.
- Los diagramas de objetos se usan para indicar la semántica de escenarios primarios y secundarios que proporcionan una traza del comportamiento emergente del sistema.



# Diagramas de Objetos (2)

---

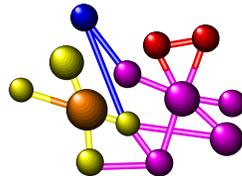
- La forma del icono del objeto es también un rectángulo cuyo contorno es una línea llena y siempre tiene el carácter “:”, que indica instanciación.
- Dentro de la misma aparece el nombre del objeto y sus atributos con los valores de los mismos.
- El nombre de un objeto puede escribirse en cualquiera de las tres formas siguientes:
  - **Objeto** → Solamente el nombre del objeto
  - **: Clase** → Solamente el nombre de la clase
  - **Objeto: Clase** → Nombre y clase del objeto



# Diagramas de Objetos (3)

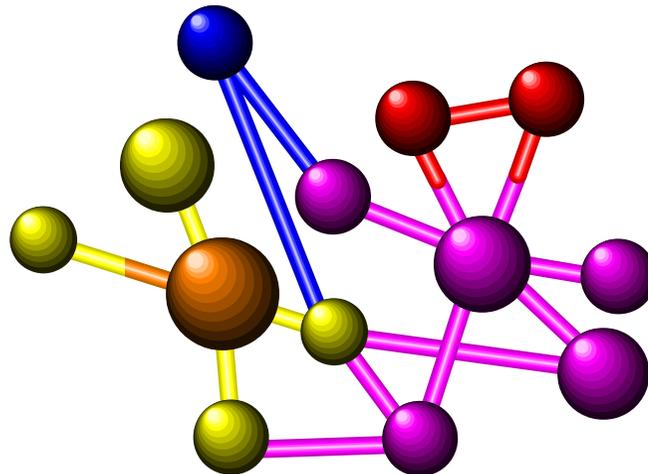
---

- Algunas particularidades se detallan a continuación:
  - Si el texto es largo, puede abreviarse o puede agrandarse el icono.
  - Si hay varios iconos en el mismo diagrama con el mismo nombre de objeto, entonces todos ellos denotan el mismo objeto.
  - Si varios iconos de objeto usan el mismo nombre en diferentes diagramas, entonces denotan a objetos diferentes (a menos que se califique explícitamente).
  - Si no se especifica nunca la clase del objeto entonces la misma se considera anónima.
  - El nombre dado para la clase del objeto debe ser el de la clase verdadera.
  - Los objetos deben exponer los atributos, pero no todos, sólo los que se considere útiles; pueden no escribirse los mismos.
  - La sintaxis para los atributos es la misma que para los atributos de clases.



# Diagramas de Objetos (4)

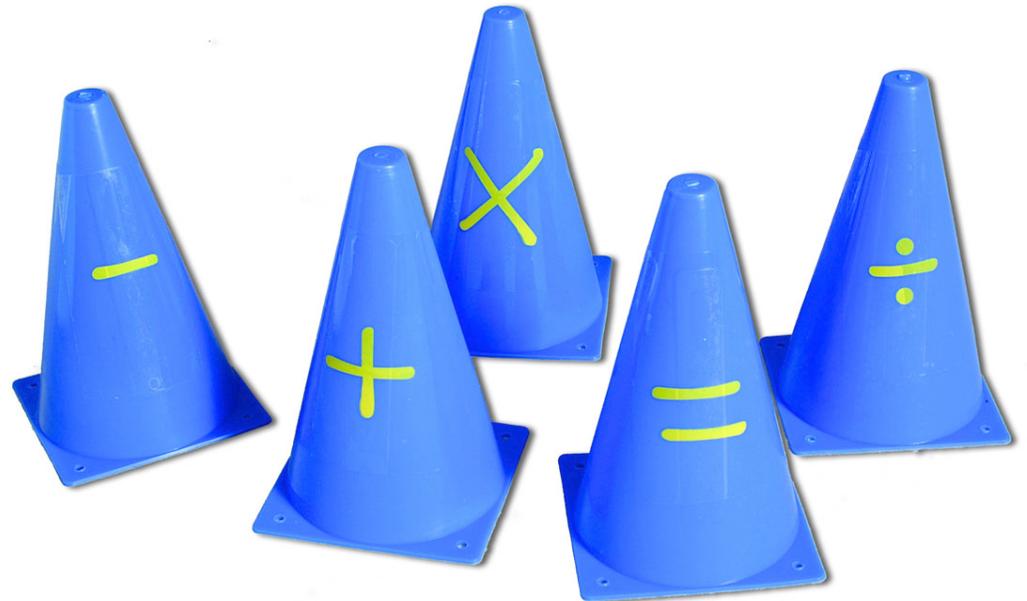
---



# La Notación

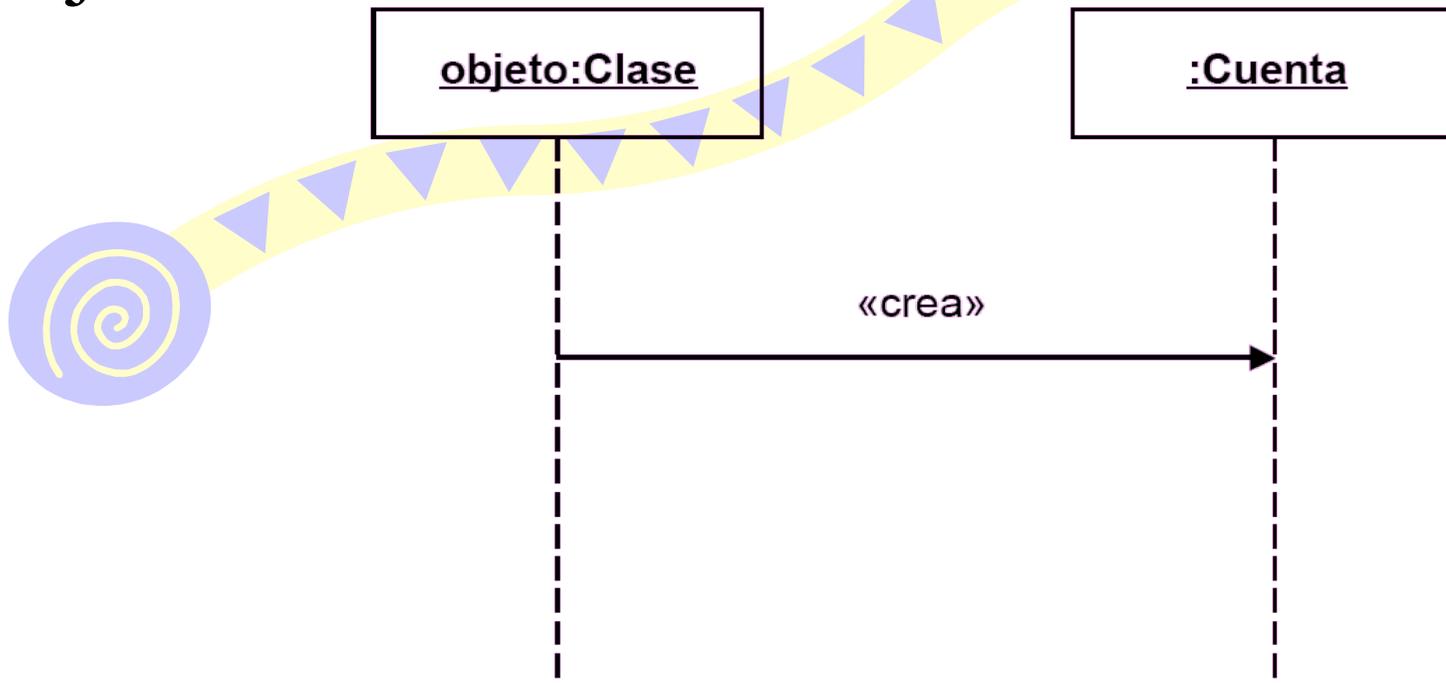
---

- Contenido
  - Elementos de la Notación
  - Diagramas de Clases
  - Diagramas de Objetos
  - **Diagramas de Interacción o Secuencia**
  - Diagrama de Estados



# Diagrama de Secuencia (1)

- Una *interacción* es un comportamiento que se centra en los intercambios observables de información entre los objetos.
- Una *línea de vida* representa la participación de un objeto dado en una determinada interacción.

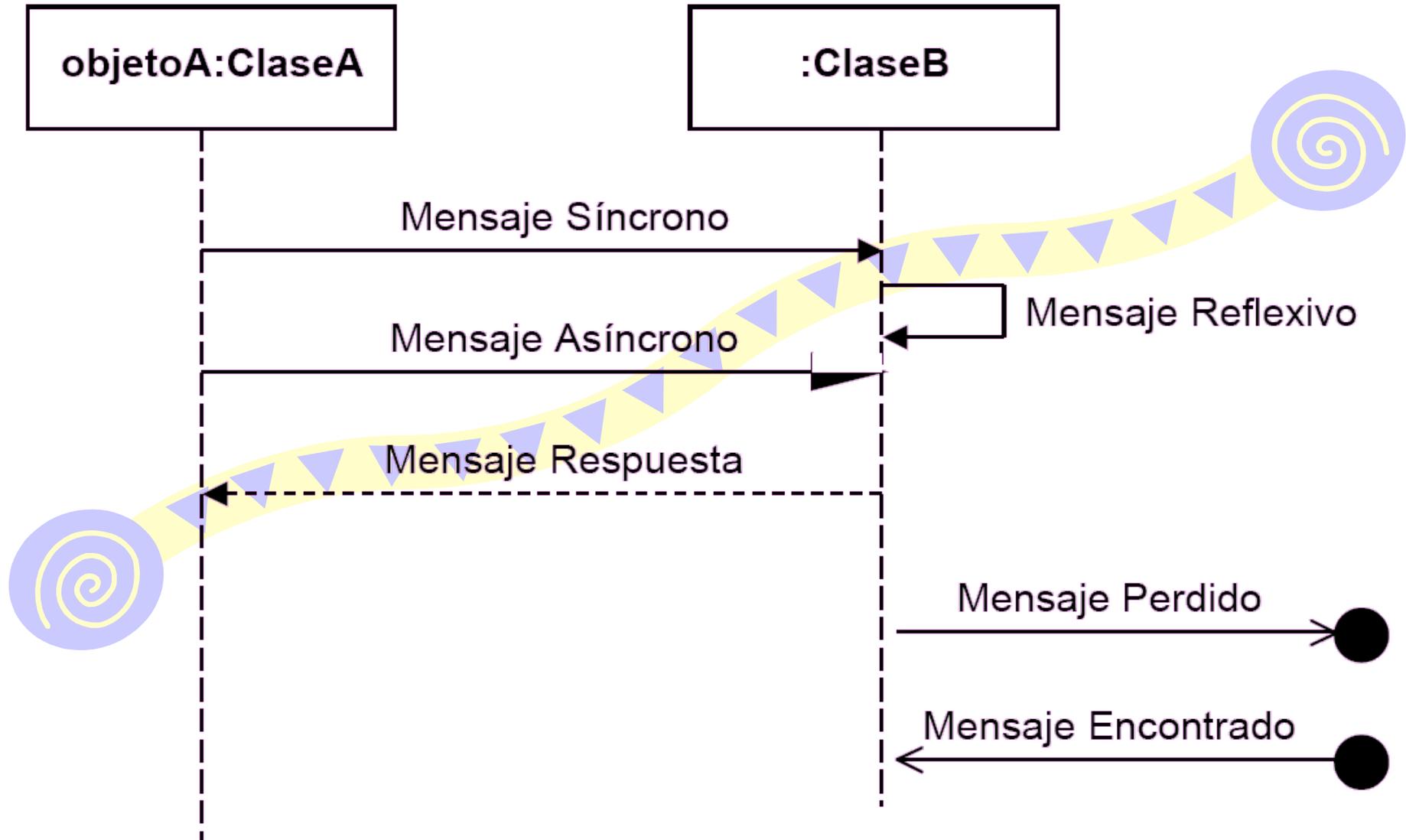


# Diagrama de Secuencia (2)

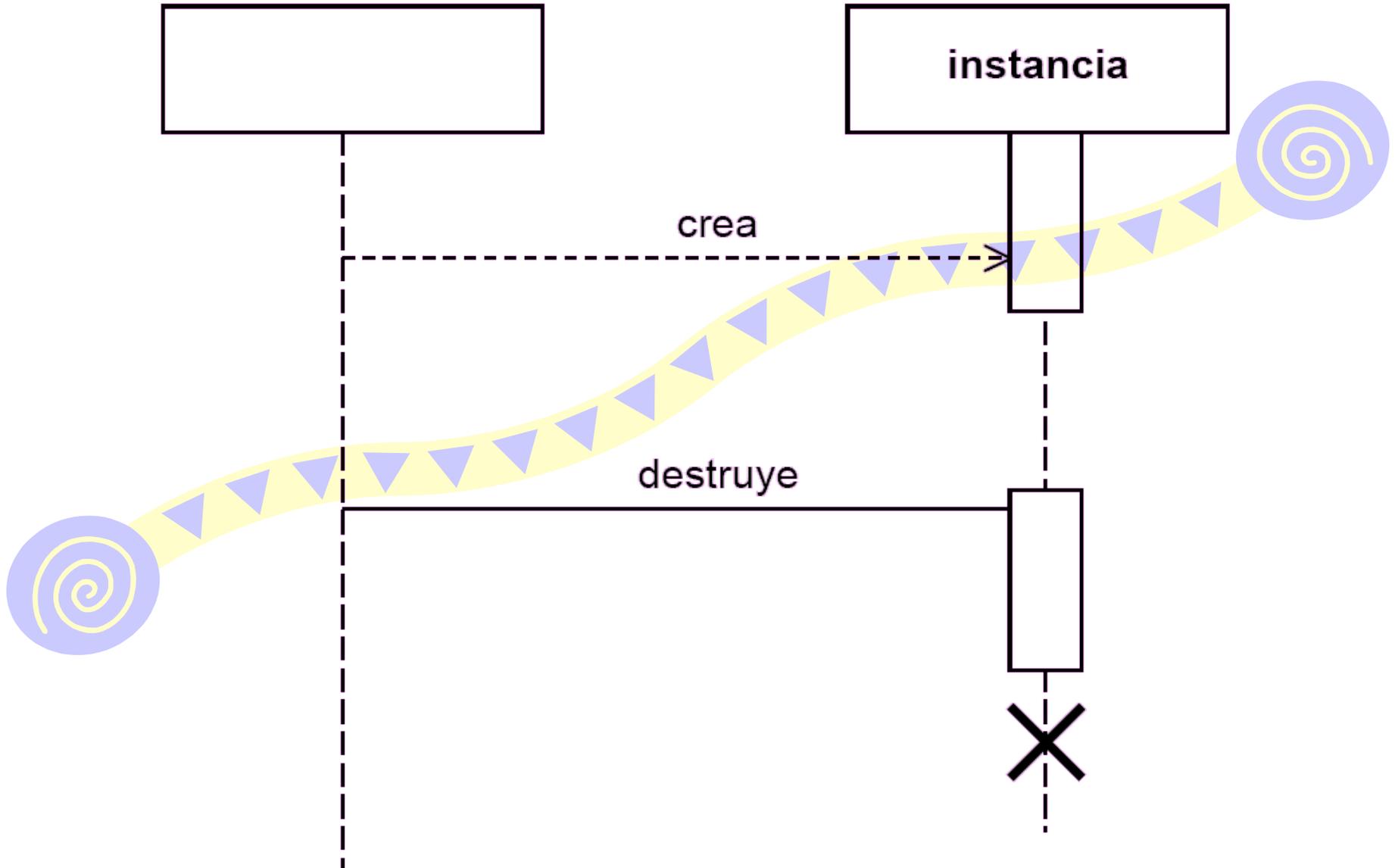
---

- Los objetos se comunican a través de mensajes entre líneas de vida. La notación para el mensaje siempre es una flecha, pero el tipo de flecha y la punta varía en función del tipo de mensaje:
  - Llamada síncrona: aparece como una línea sólida con una punta también sólida.
  - Llamada asíncrona: aparece como una línea sólida con una mitad de punta de flecha.
  - Mensaje respuesta: aparece como una línea punteada con una punta sólida.
  - Mensaje de creación de objeto: aparece como una línea punteada con una punta emplumada.
  - Mensaje “perdido”: (uno que se conoce el remitente pero no el receptor) tiene un pequeño círculo negro siguiendo a la punta.
  - Mensaje “encontrado”: (uno en que se conoce el receptor pero no al remitente) tiene un pequeño círculo negro justo antes de la punta.

# Diagrama de Secuencia (3)

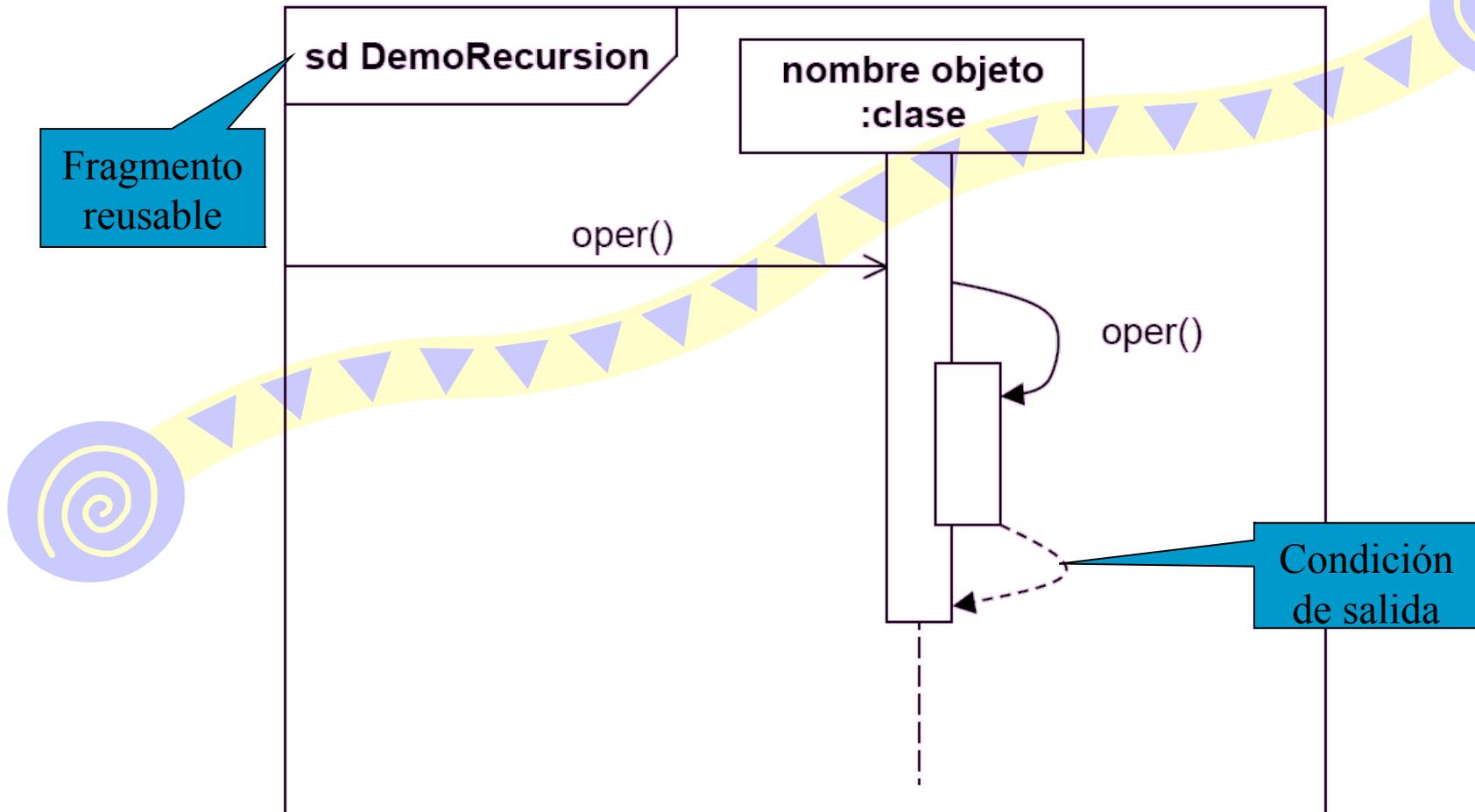


# Diagrama de Secuencia (4)



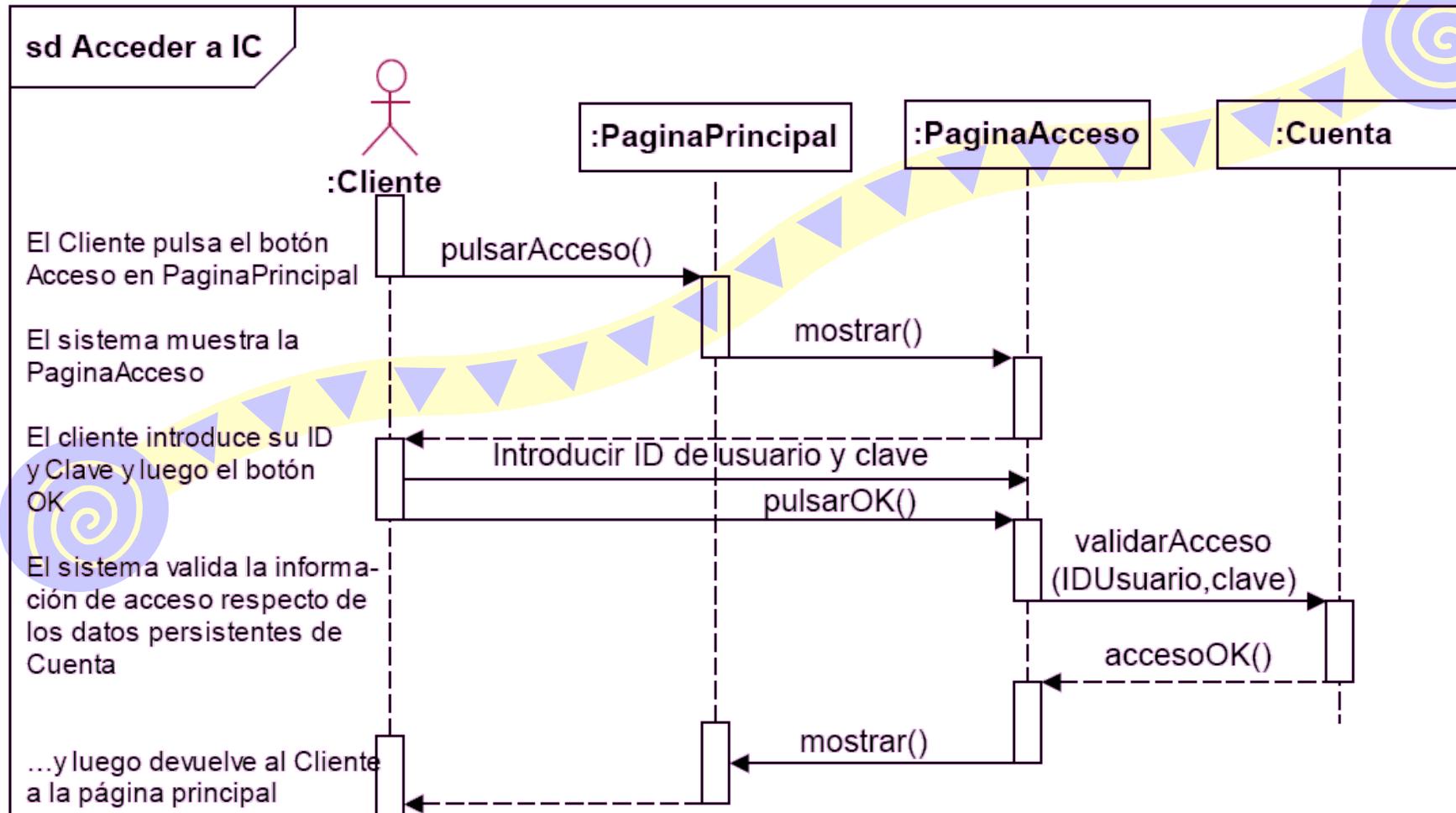
# Diagrama de Secuencia (5)

- La operación *oper()* se llama a sí misma. Habrá una condición en la operación que parará la recursividad.



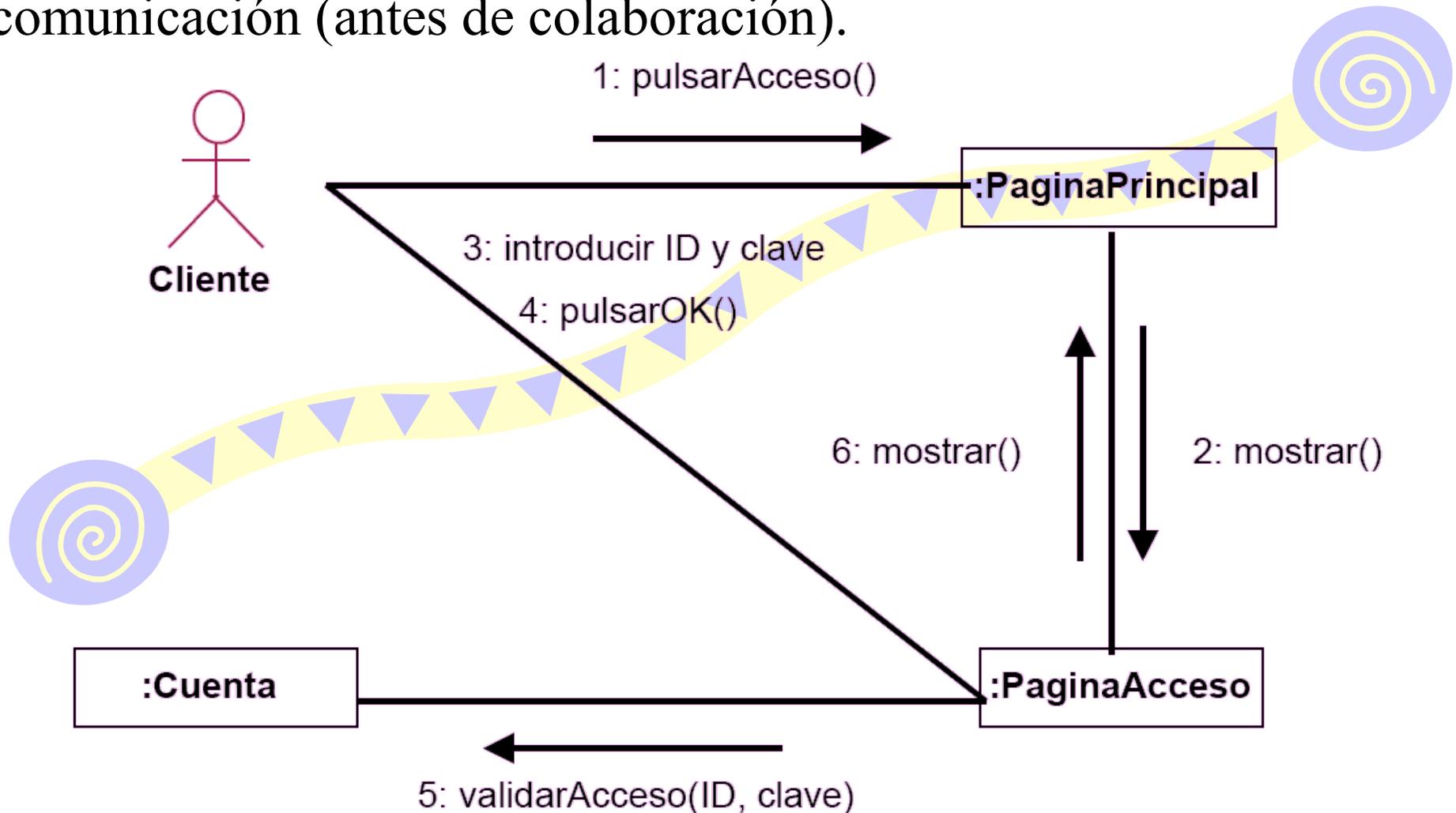
# Diagrama de Secuencia (6)

- Un diagrama de secuencia muestra el orden temporal de los mensajes.



# Diagrama de Secuencia (7)

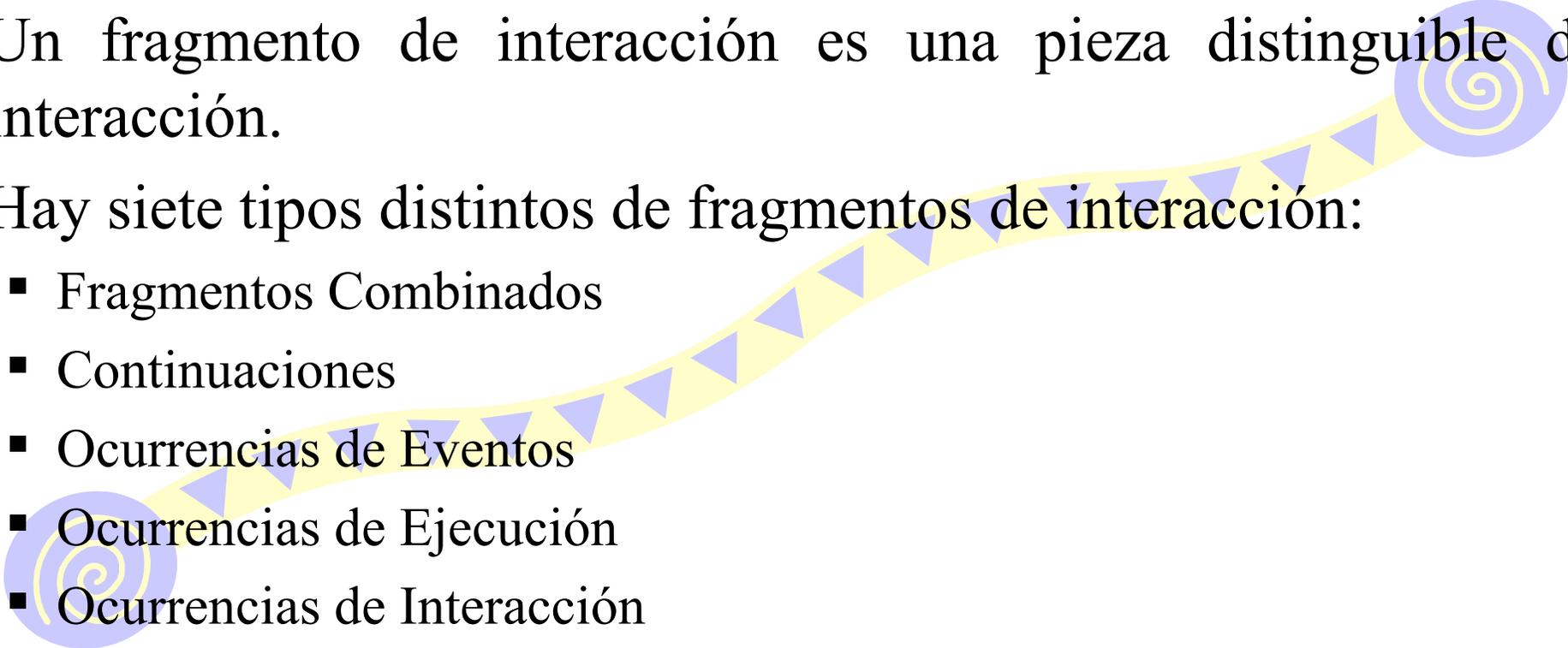
- Un diagrama de secuencia puede transformarse en un diagrama de comunicación (antes de colaboración).



# Diagrama de Secuencia (8)

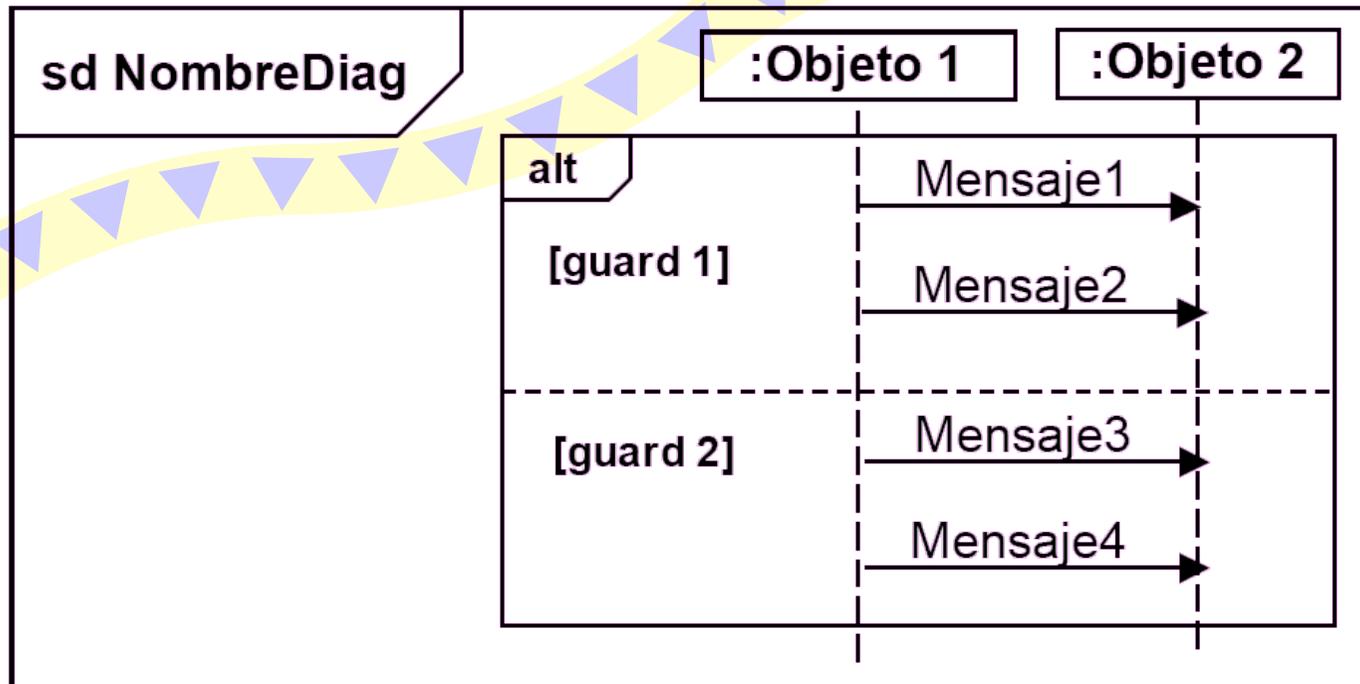
---

## Fragmentos de Interacción

- Un fragmento de interacción es una pieza distinguible de interacción.
  - Hay siete tipos distintos de fragmentos de interacción:
    - Fragmentos Combinados
    - Continuaciones
    - Ocurrencias de Eventos
    - Ocurrencias de Ejecución
    - Ocurrencias de Interacción
    - Descomposición en Partes
    - Invariantes de Estado
- 

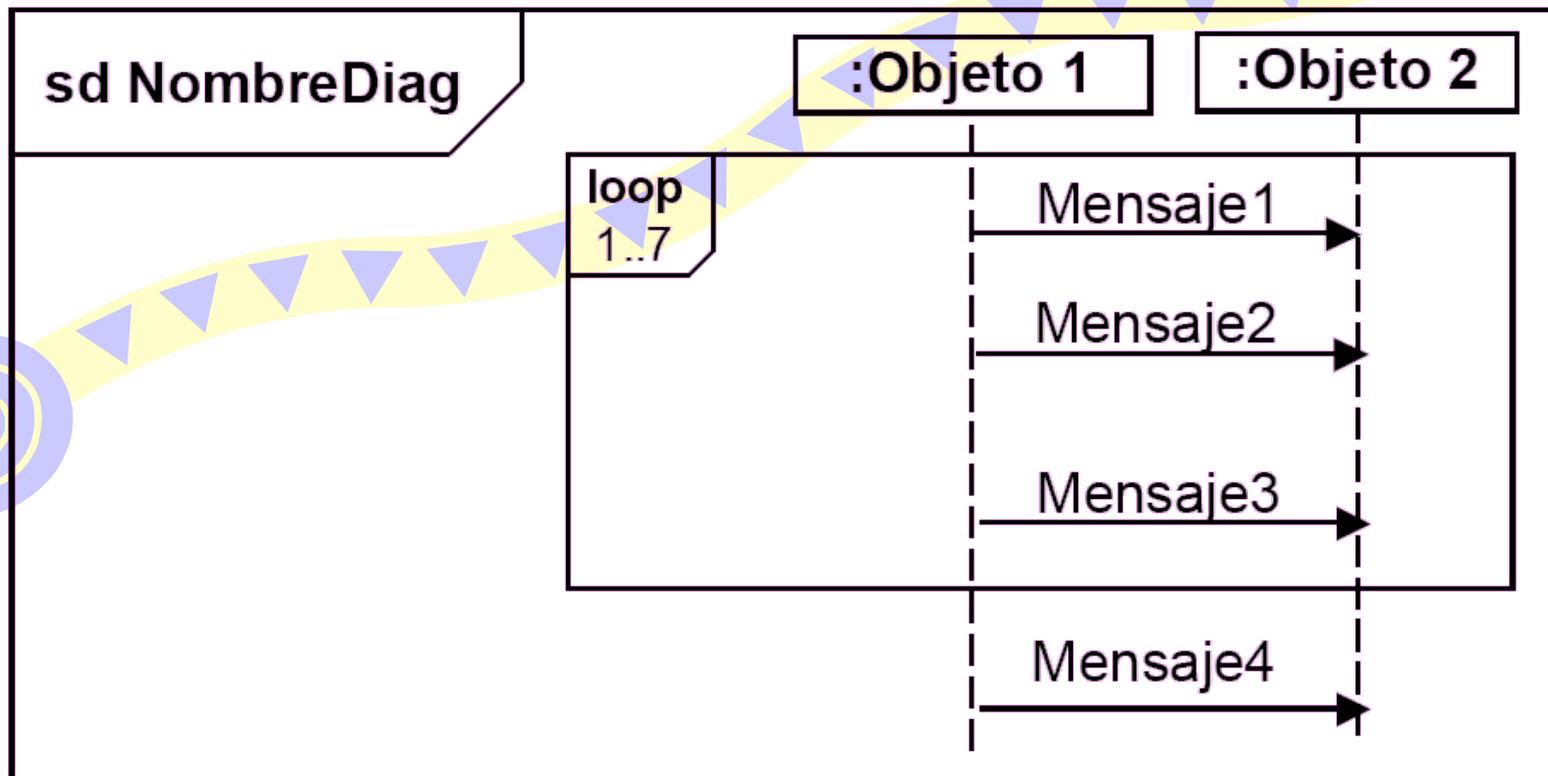
# Diagrama de Secuencia (9)

- Un fragmento combinado es una combinación de uno o más operandos de interacción, cada uno de los cuales contiene uno o más fragmentos de interacción y un operador de interacción:
  - alt:** en el que el fragmento combinado representa una elección de comportamiento: cada fragmento tiene un resguardo, que significa que sólo se ejecutará uno de los operandos.



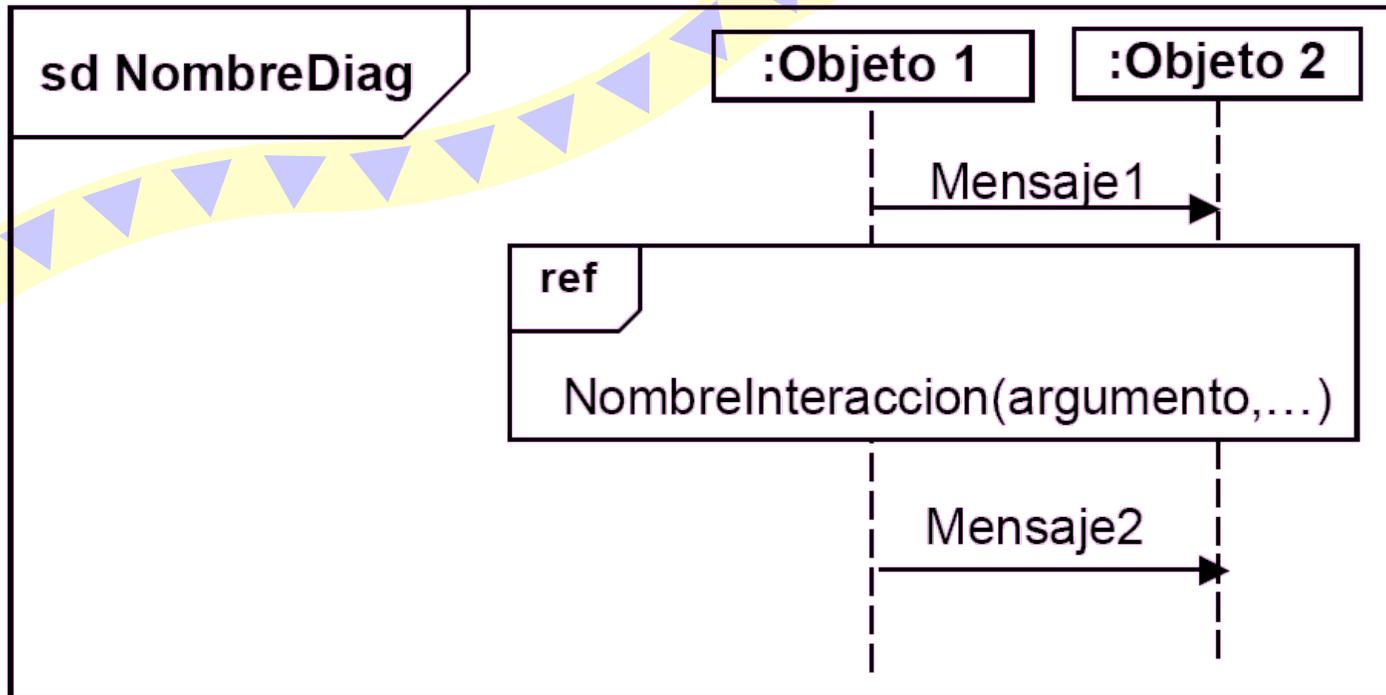
# Diagrama de Secuencia (10)

- **assert:** en el que el fragmento combinado representa una aserción (acción y efecto de afirmar o dar por cierto algo): el fragmento debe ocurrir tal como se especifica.
- **loop:** en el que el fragmento combinado representa un bucle: el operando bucle se repite el número especificado de veces.



# Diagrama de Secuencia (11)

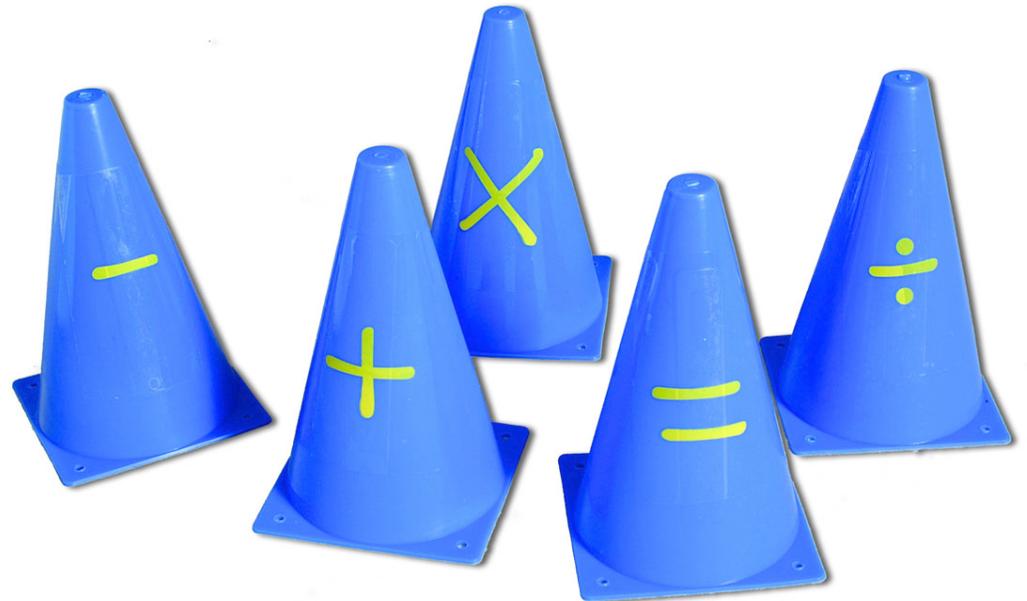
- Una ocurrencia de interacción representa la ocurrencia de una pieza de una interacción particular con valores específicos y que reemplaza al ocupante definido para la interacción. Se usan generalmente para factorizar el comportamiento común que existe dentro de un número de interacciones.



# La Notación

---

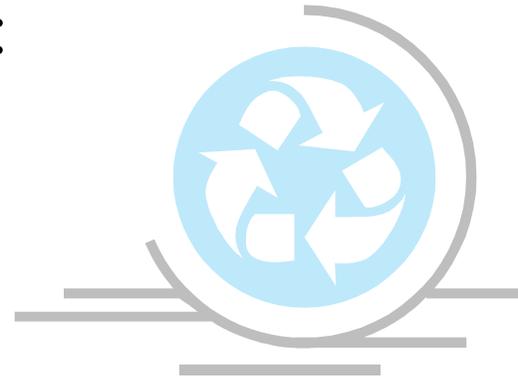
- Contenido
  - Elementos de la Notación
  - Diagramas de Clases
  - Diagramas de Objetos
  - Diagramas de Interacción o Secuencia
  - **Diagrama de Estados**



# Diagrama de Estado (1)

---

- Un **estado** es una condición en la que puede estar un objeto en algún momento de su ciclo de vida, durante un cierto tiempo.
- Mientras está en un determinado estado, el objeto puede realizar algunas (o todas) las siguientes acciones:
  - Realizar una actividad
  - Esperar un evento
  - Satisfacer una o más condiciones



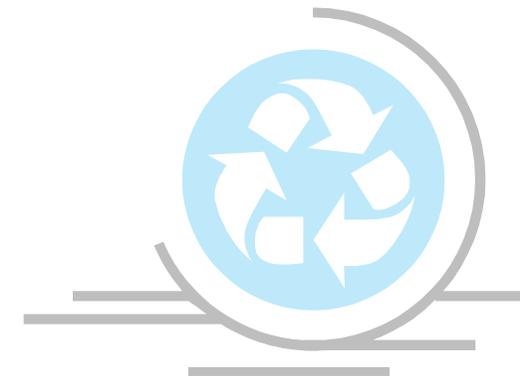
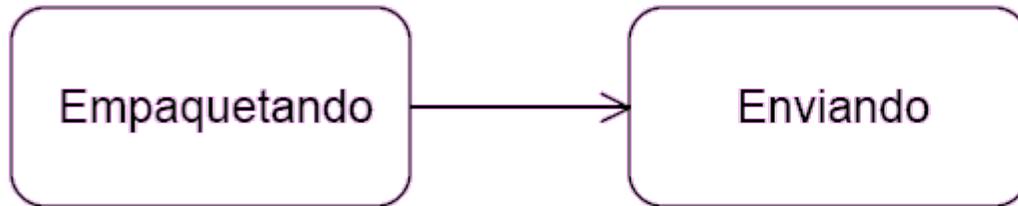
Contabilizando

Buscando  
Libros

Enviado

# Diagrama de Estado (2)

- Una transición es un cambio del objeto desde un estado (estado fuente) a otro (estado destino). También puede haber una auto-transición cuando ambos estados coinciden.

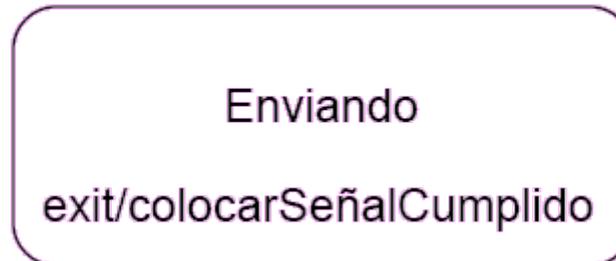


# Diagrama de Estado (3)

- El símbolo de estado puede contener además la siguiente información:
  - Una acción de entrada:** es aquella que el objeto realiza siempre inmediatamente después de entrar al estado. Aparece como entrada/NombreAcción.

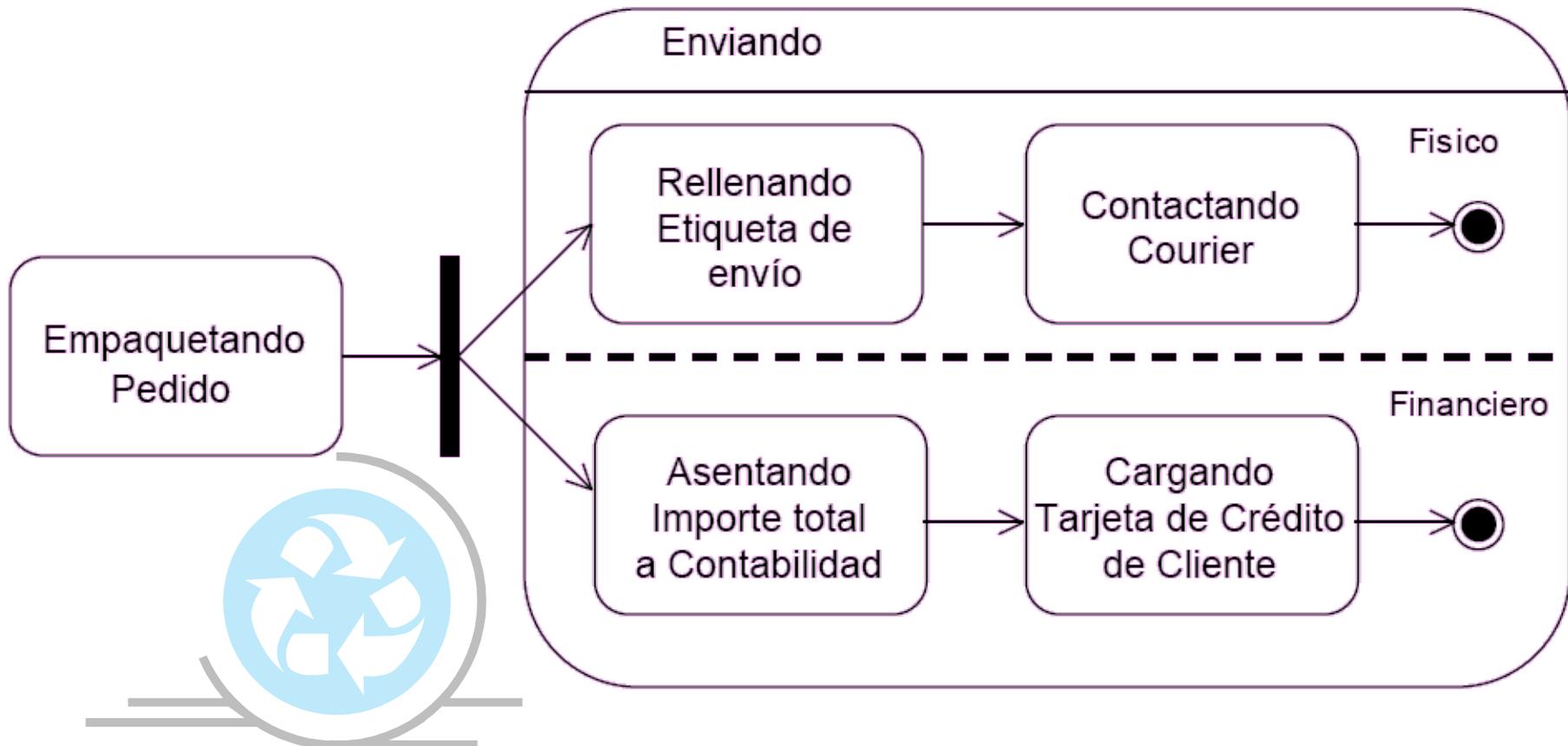


- Una acción de salida:** es aquella que siempre realiza el objeto inmediatamente antes de abandonar el estado en que se encuentra.



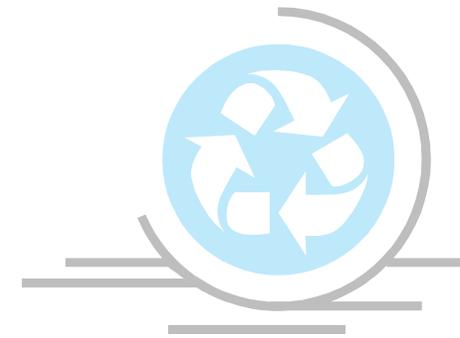
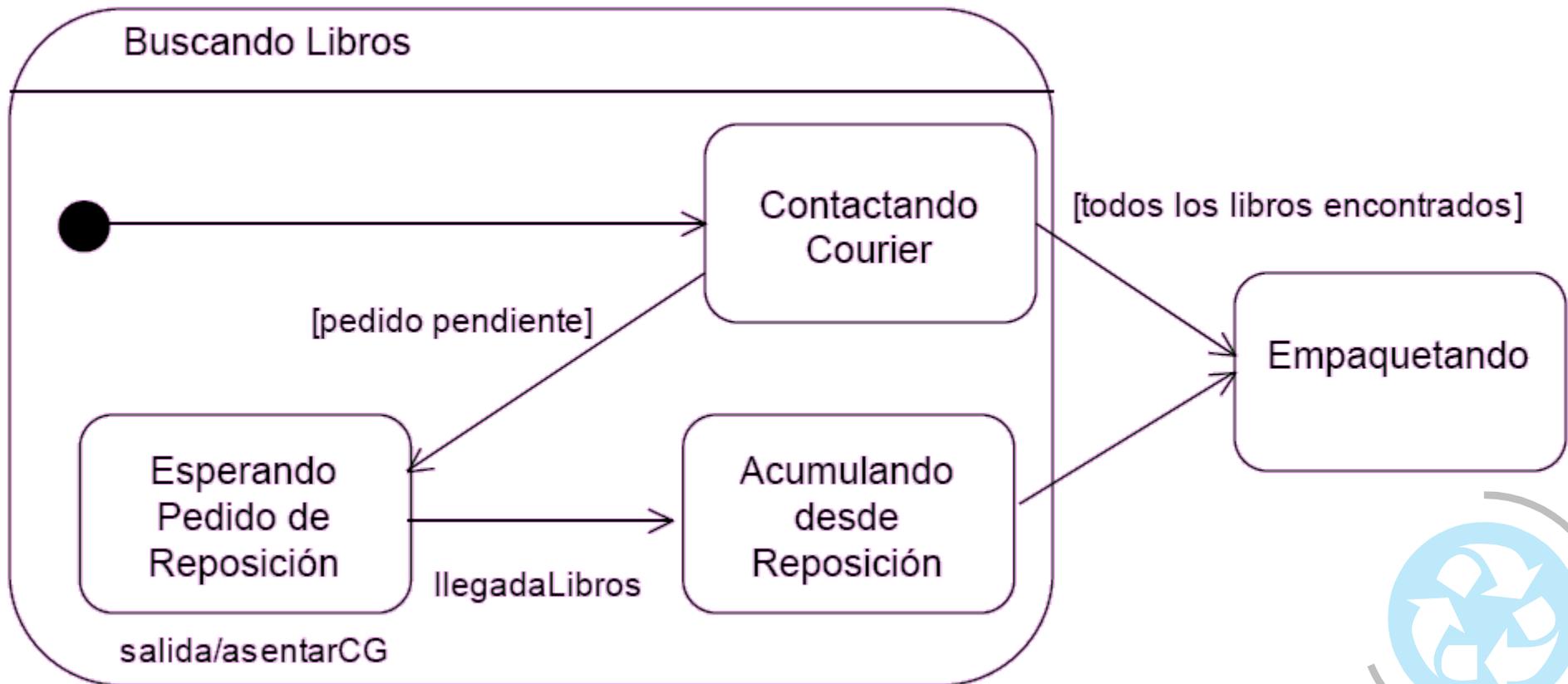
# Diagrama de Estado (4)

- Una bifurcación (fork) divide la transición en dos o más. En la figura vemos la notación para una bifurcación que opera en la transición proveniente del estado *Empaquetando el Pedido*.



# Diagrama de Estado (5)

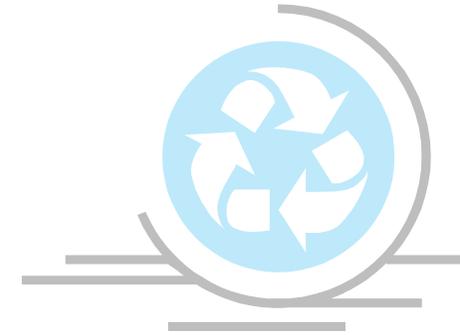
- Un **Subestado Secuencial** es aquel en el que se encuentra un objeto con exclusión de cualquier otro en el mismo nivel dentro del estado compuesto.



# Diagrama de Estado (6)

---

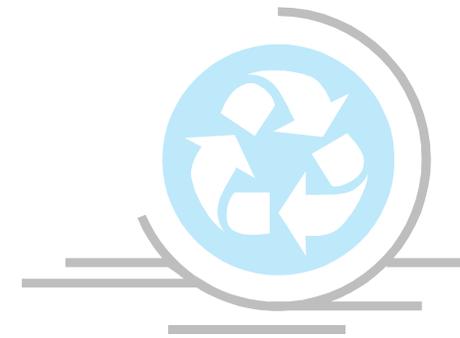
- Un diagrama de transición de estados se utiliza para mostrar:
  - El espacio de estados de una clase determinada.
  - Los eventos que provocan una transición de un estado a otro.
  - Las acciones que resultan de ese cambio de estado.
- Un solo diagrama de transición de estados representa una vista del modelo dinámico de una sola clase o del sistema completo.
- Los dos elementos esenciales de un diagrama de transición de estados son los estados y las transiciones entre estados.
- El **estado** de un objeto representa los resultados acumulados de su comportamiento.



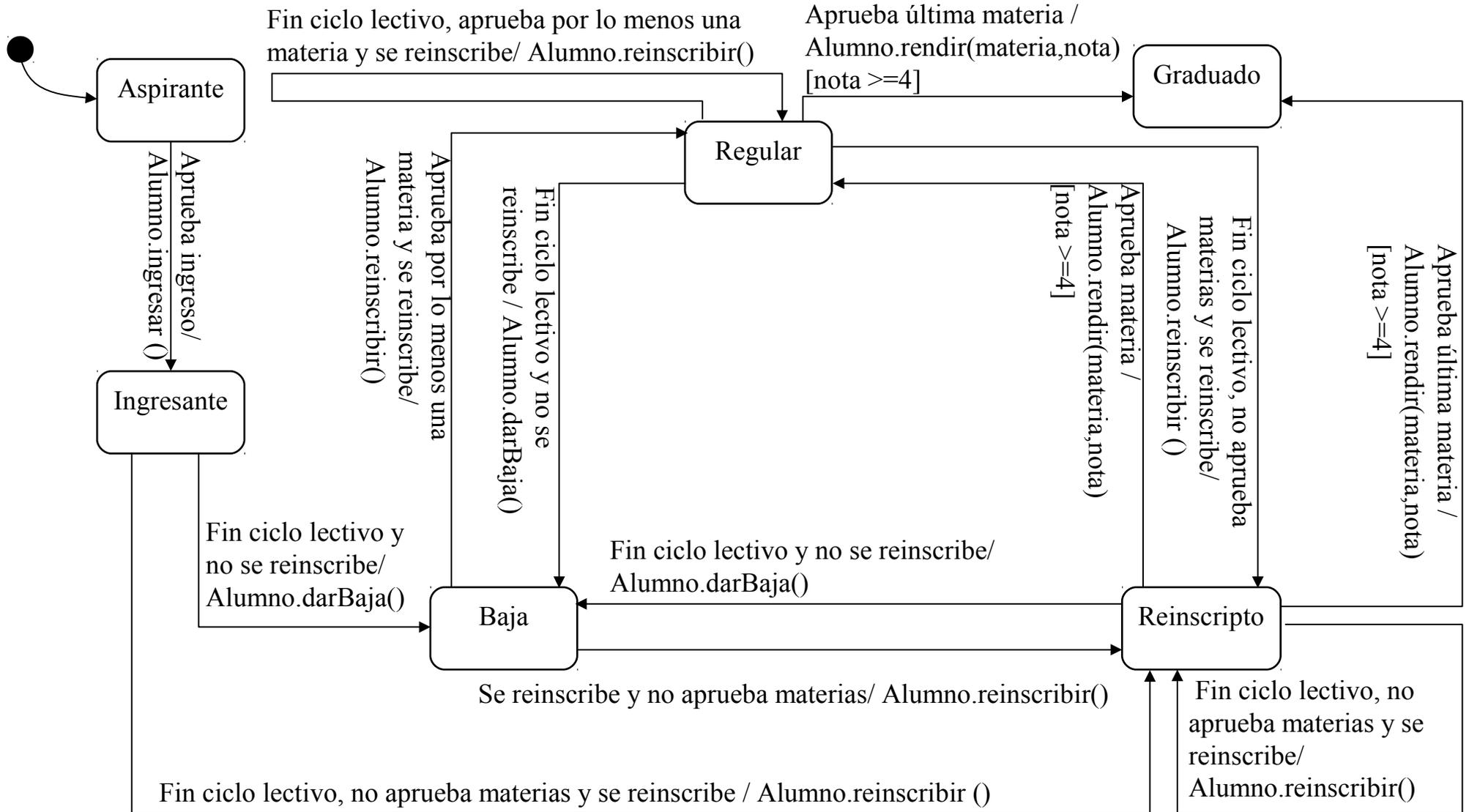
# Diagrama de Estado (7)

---

- Los iconos descritos hasta aquí constituyen los elementos esenciales de todos los diagramas de transición de estados.
- En conjunto, proveen al desarrollador de suficiente notación para describir máquinas de estados finitos lisas y llanas, adecuadas para aplicaciones con un número limitado de estados.
- Un ejemplo de la notación esencial puede verse en la diapositiva siguiente, donde se modelan los estados de la clase alumnos de un determinado establecimiento educativo.



# Diagrama de Estado (8)



# La Notación

---

- Contenido
  - Elementos de la Notación
  - Diagramas de Clases
  - Diagramas de Objetos
  - Diagramas de Interacción o Secuencia
  - Diagrama de Estados

