

### Sumario

Análisis y diseño orientado a objetos.....	2
Definición.....	2
Ejemplo de Análisis Orientado a Objetos.....	2
Requisitos.....	2
Expresar los requisitos como Casos de Uso.....	3
Modelo de objetos.....	4
Identificar objetos y clases.....	5
Identificar y depurar relaciones.....	6
Identificar atributos de objetos y relaciones.....	8
Añadir herencia.....	9
Comprobar los casos de uso (iterar).....	9
Modularizar.....	10
Añadir y simplificar métodos.....	10
Modelo dinámico.....	10
Identificar sucesos.....	11
Construir diagramas de estados.....	11
Añadir métodos.....	12
Modelo funcional.....	12
Identificar valores de entrada/salida.....	12
Construir diagramas de flujo de actividad.....	12
Describir funciones.....	13
Identificar restricciones y dependencias funcionales entre objetos.....	13
Definir criterios de optimización (iterar).....	13
Añadir métodos.....	14

# Análisis y diseño orientado a objetos

## Definición

El análisis y diseño orientado a objetos (ADOO) es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio absoluto en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional. Todo sistema de información requiere de artefactos o componentes (clases) para llevar a cabo tareas, es de gran importancia dentro de la ingeniería de software tener un buen "análisis y diseño" para un mejor desarrollo, que conlleva a que tan "escalable" sea un sistema de información.

En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML).

**ADOO** aplica técnicas de modelado de objetos para analizar los requerimientos para un contexto (por ejemplo, un sistema de negocio, un conjunto de módulos de software) y para diseñar una solución para mejorar los procesos involucrados.

No está restringido al diseño de programas de computadora, sino que cubre sistemas enteros de distinto tipo. Las metodologías de análisis y diseño más modernas son "casos de uso" guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue.

El lenguaje unificado de modelado se ha vuelto el lenguaje de modelado estándar usado en análisis y diseño orientado a objetos.

## Ejemplo de Análisis Orientado a Objetos

### Requisitos

Explican lo que se desea que haga el sistema, ya sea en lenguaje natural, o en forma de casos de uso a la Jacobson.

### Ejemplo:

Se desea diseñar el software necesario para una red bancaria provista de cajeros automáticos (ATM, *automatic teller machines*), que serán compartidos por un consorcio de bancos. Cada banco dispone de su propio ordenador, provisto de software propio, que lleva la información sobre sus cuentas y procesa las transacciones que actúan sobre dichas cuentas. A este ordenador están conectadas 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 correctamente el registro de las transacciones efectuadas,

cumplirá características aceptables de seguridad y manejará correctamente 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.

### **Expresar los requisitos como Casos de Uso**

Preparar escenarios detallados. Primero los normales. Después se añaden los problemas que pueden surgir. En el ejemplo de los cajeros automáticos:

- **Escenario normal:**

- El cajero automático pide al cliente que inserte la tarjeta de crédito.
- El cliente inserta la tarjeta de crédito.
- El cajero automático acepta la tarjeta de crédito y lee el número de tarjeta y el código del banco.
- El cajero automático pide la contraseña al cliente.
- El cliente teclea "1234".
- El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
- El consorcio envía el número de tarjeta y la contraseña al banco.
- El banco notifica la aceptación al consorcio.
- El consorcio notifica la aceptación al cajero automático.
- El cajero automático pide al cliente que elija el tipo de transacción: retirada de fondos, depósito, transferencia, información.
- El cliente selecciona retirada de fondos.
- El cajero automático pide al cliente que teclee la cantidad.
- El cliente teclea 25000.
- El cajero automático comprueba que la cantidad está dentro de los límites generales.
- El cajero automático genera una transacción y la envía al consorcio.
- El consorcio pasa la transacción al banco.
- El banco aprueba la transacción.
- El banco actualiza la cuenta.
- El banco envía al consorcio la notificación de aceptación y el nuevo saldo de la cuenta.
- El consorcio envía al cajero automático la notificación de aceptación y el nuevo saldo de la cuenta.
- El cajero automático entrega el dinero al cliente.
- El cliente toma el dinero.
- El cajero automático pregunta al cliente si quiere un recibo.
- El cliente contesta SI.
- El cajero automático imprime un recibo y pide al cliente que lo tome.
- El cliente toma el recibo.
- El cajero automático pregunta al cliente si quiere hacer otra operación.

- El cliente contesta NO.
- El cajero automático expulsa la tarjeta de crédito e indica al cliente que la tome.
- El cliente toma la tarjeta de crédito.
- El cajero automático vuelve a la situación inicial.
- **Escenario con problemas:**
  - El cajero automático pide al cliente que inserte la tarjeta de crédito.
  - El cliente inserta la tarjeta de crédito.
  - El cajero automático acepta la tarjeta de crédito y lee el número de tarjeta y el código del banco.
  - El cajero automático pide la contraseña al cliente.
  - El cliente teclea "9999".
  - El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
  - El consorcio envía el número de tarjeta y la contraseña al banco.
  - El banco notifica el rechazo al consorcio.
  - El consorcio notifica el rechazo al cajero automático.
  - El cajero automático notifica el rechazo al cliente y pide que teclee de nuevo la contraseña.
  - El cliente teclea "1234".
  - El cajero automático envía el número de tarjeta, el código del banco y la contraseña al consorcio.
  - El consorcio envía el número de tarjeta y la contraseña al banco.
  - El banco notifica la aceptación al consorcio.
  - El consorcio notifica la aceptación al cajero automático.
  - El cajero automático pide al cliente que elija el tipo de transacción: retirada de fondos, depósito, transferencia, información.
  - El cliente selecciona retirada de fondos.
  - El cajero automático pide al cliente que teclee la cantidad.
  - El cliente teclea CANCELAR.
  - El cajero automático expulsa la tarjeta de crédito e indica al cliente que la tome.
  - El cliente toma la tarjeta de crédito.
  - El cajero automático vuelve a la situación inicial.

### **Modelo de objetos**

Consta de los siguientes pasos:

- 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

### Identificar objetos y clases

Consta de los siguientes pasos:

- Seleccionar nombres en los requisitos
- Añadir clases adicionales procedentes de nuestro conocimiento del tema
- Eliminar redundancias
- Eliminar clases irrelevantes
- Eliminar clases vagas
- Separar atributos
- Separar métodos
- Eliminar objetos de diseño

Resultado: Preparar diccionario de clases

En el ejemplo de los cajeros automáticos:

- Seleccionar nombres en los requisitos

Los nombres extraídos de los requisitos en lenguaje natural del sistema de cajeros automáticos son los siguientes (23):

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

- Añadir clases adicionales procedentes de nuestro conocimiento del tema

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.

- Separar atributos

Los atributos definen datos asociados a un objeto, en lugar de objetos. Aunque la separación no es clara (los atributos pueden ser objetos embebidos) en algunos casos se pueden

distinguir. En el ejemplo, pueden considerarse atributos *Información sobre la cuenta*, (atributo de *Cuenta bancaria*), *Dinero en efectivo* y *Recibo* (atributos de *Cajero automático*).

- Separar métodos

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*.

Resultado. Del análisis anterior, resultan seleccionadas las siguientes clases (11):

Cajero automático, Consorcio de bancos, Banco, Ordenador del banco, Cuenta bancaria, Transacción, Estaciones de cajero, Cajero humano, Tarjeta de crédito, Ordenador central, Cliente.

El diccionario de clases contiene la definición detallada de todas estas clases en lenguaje natural.

Ejemplo:

- *Cajero automático*: Terminal remoto que permite a los clientes realizar transacciones utilizando tarjetas de crédito para identificarse. El cajero automático 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 cajero automático no opera cuando está desconectado de la red.
- *Consorcio 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.

### Identificar y depurar relaciones

Consta de los siguientes pasos:

- Seleccionar verbos relacionales en los requisitos
- Añadir relaciones adicionales procedentes de nuestro conocimiento del tema
- 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

En el ejemplo de los cajeros automáticos:

- Seleccionar verbos relacionales en los requisitos
  1. Una *Red bancaria* está provista de *Cajeros automáticos*.
  2. El *Consortio de bancos* comparte los *Cajeros automáticos*.
  3. Cada *Banco* dispone de un *Ordenador del banco*.
  4. El *Ordenador del banco* dispone de *Software*.
  5. El *Ordenador del banco* lleva la información sobre las *Cuentas bancarias*.
  6. El *Ordenador del banco* procesa *Transacciones*.
  7. Una *Transacción* actúa sobre una *Cuenta bancaria*.
  8. Las *Estaciones de cajero* están conectadas al *Ordenador del banco*.
  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*.
  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*.
- Añadir relaciones adicionales procedentes de nuestro conocimiento del tema
    1. 28. Las *Tarjetas de crédito* están asociadas a las *Cuentas bancarias*.
    2. 29. Los *Cajeros humanos* son empleados de los *Bancos*.
  - Eliminar relaciones de diseño o entre clases eliminadas

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.

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:

16 a. El *Ordenador central* se comunica con el *Banco*.

- Reducir relaciones ternarias

Son relaciones entre tres o más clases. Muchas veces es posible descomponerlas en varias relaciones binarias (entre dos clases). Por ejemplo, la relación número 12 (El *Cajero humano* introduce *Transacciones* sobre las *Cuentas bancarias*) puede descomponerse en:

1. 12a. El *Cajero humano* introduce *Transacciones*
2. 12b. Las *Transacciones* actúan sobre las *Cuentas bancarias*.

De igual modo, la número 17 puede descomponerse así:

3. 17a. Los *Cajeros automáticos* entregan *Dinero en efectivo*.
4. 17b. El *Usuario* recoge el *Dinero en efectivo*.

- 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 (por ejemplo, si la multiplicidad es distinta).

- Añadir relaciones olvidadas

Por ejemplo:

1. 30. Los *Clientes* tienen *Cuentas*.
2. 31. Las *Transacciones* son autorizadas por la *Tarjeta de crédito*.
3. 32. Las *Transacciones* pueden introducirse en una *Estación de cajero*.

- Definir la multiplicidad de cada asociación

1. Un *Banco* puede contener muchas *Cuentas*.
2. Un *Cliente* puede tener muchas *Cuentas*.
3. Un *Cliente* puede tener muchas *Tarjetas de crédito*.
4. Un *Banco* emplea muchos *Cajeros*.
5. Un *Banco* tiene un solo *Ordenador del banco*.
6. El *Ordenador central* se comunica con muchos *Ordenadores del banco*.
7. Etc.

El resultado de estas operaciones es un esqueleto del modelo de clases sin herencia.

### Identificar atributos de objetos y relaciones

Consta de los siguientes pasos:

- Distinguir los objetos de los atributos
- Distinguir entre los atributos de objetos y de relaciones
- El identificador del objeto es siempre un atributo implícito



- Eliminar atributos privados (de diseño)
- Eliminar atributos de detalle fino
- Localizar atributos discordantes (dividir la clase)

En el ejemplo de los cajeros automáticos:

- 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 del banco, Código de la tarjeta.
- Atributos de las relaciones
  - 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.

### Añadir herencia

Introducimos clases nuevas (virtuales) 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 de los cajeros automáticos:

- 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.

### Comprobar los casos de uso (iterar)

Para localizar fallos que deben corregirse fijarse en:

- Asimetrías en las relaciones: añadir clases nuevas para equilibrarlas.
- Atributos muy dispares: descomponer una clase en dos.
- Dificultades en la formación de superclases: descomponer una clase en dos. Una de sus partes puede ajustar mejor.
- Operaciones sin objetivo: añadir clase.
- Relaciones duplicadas: crear superclase.
- Conversión de relaciones en clases: por ejemplo, clase *Empleado*.

- 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.

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 *Consorcio* y *Ordenador central*, por otra. Fusionamos esas clases.

## Modularizar

Agrupar clases en módulos.

En el ejemplo de los cajeros automáticos. Posibles módulos:

- Cajeros en general: *Cajero*, *Estación de cajero*, *Cajero automático*, *Estación de entrada*.
- Cuentas en general: *Cuenta*, *Tarjeta de crédito*, *Autorización*, *Cliente*, *Transacción*, *Transacción de cajero*, *Transacción remota*.
- Bancos: *Banco*, *Consorcio*.

## Añadir y simplificar métodos

- Todos los atributos se suponen accesibles.
- Añadir métodos que permitan navegar de un objeto a otro.

## Modelo dinámico

Consta de los siguientes pasos:

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

### Identificar sucesos

Los sucesos 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 (trazas de eventos) y diagramas de colaboración (diagramas de flujo de eventos).

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 evento 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 eventos 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 evento independientemente de la cantidad entregada.

No agrupar los eventos no equivalentes: *El banco autoriza la transacción* es distinto evento que *El banco rechaza la transacción*.

### Construir diagramas de estados

Uno por clase.

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

### **Añadir métodos**

Los eventos son métodos. Es preciso decidir de qué clase de objetos.

Las acciones y actividades realizadas en los estados son métodos.

## **Modelo funcional**

Consta de los siguientes pasos:

- Identificar valores de entrada/salida
- Construir diagramas de flujo de actividad
- Describir funciones
- Identificar restricciones y dependencias funcionales entre objetos
- Definir criterios de optimización (iterar)
- Añadir métodos

### **Identificar valores de entrada/salida**

Son los que pasan información desde los objetos externos al sistema de software propiamente dicho.

En el ejemplo de los cajeros automáticos son objetos externos:

- Cliente
- Tarjeta de crédito
- Cajero humano

Los valores de entrada/salida serán:

- Del cliente al cajero automático: contraseña, tipo de transacción, tipo de cuenta, cantidad solicitada.
- De la tarjeta de crédito al cajero automático: código del banco, código de la tarjeta.
- Del cajero automático al cliente: dinero en efectivo, recibo, mensajes.

### **Construir diagramas de flujo de actividad**

Relacionan los valores de entrada con los de salida. Suele dividirse en varias capas o niveles.

En el ejemplo de los cajeros automáticos:

- Nivel superior: relaciona el cliente, la tarjeta de crédito y la cuenta.
- Nivel intermedio: expande la operación "realizar transacción", incluida en el nivel superior.

### Describir funciones

Descripción de cada una de las funciones de nivel mínimo que aparecen en los diagramas de flujo de actividad. La descripción puede ser:

- En lenguaje natural
- Un modelo matemático
- Pseudocódigo
- Tablas de decisión
- Etc.

En el ejemplo de los cajeros automáticos:

Descripción de la función "actualizar cuenta":

```
actualizar cuenta (cuenta, cantidad, tipo de transacción)
-> efectivo, recibo, mensaje
  Si es una retirada de efectivo:
    Si la cantidad a retirar excede el saldo,
      rechazar la transacción y no entregar dinero
    En caso contrario:
      Restar la cantidad del saldo y entregar dinero
  Si es un depósito:
    Aumentar el saldo de la cuenta y no entregar dinero
  Si es una petición de información:
    Escribir saldo y no entregar dinero
  En cualquier caso:
    El recibo incluye:
      - número del cajero automático
      - fecha y hora
      - número de la cuenta
      - tipo de transacción
      - cantidad movida
      - nuevo saldo
```

### Identificar restricciones y dependencias funcionales entre objetos

En el ejemplo de los cajeros automáticos:

- El saldo de una cuenta no puede ser negativo
- o bien:
- El saldo de una cuenta, si es negativo, no puede rebasar el límite de crédito.

### Definir criterios de optimización (iterar)

En el ejemplo de los cajeros automáticos:

- Minimizar el número de mensajes enviados entre localidades diferentes.
- Minimizar el tiempo de bloqueo de una cuenta.
- Extremadamente urgente: minimizar el tiempo de bloqueo de un banco entero.

### Añadir métodos

Las funciones del modelo funcional pueden ser simples transferencias de información, o corresponder a un método de algún objeto (operaciones interesantes). En este caso hay que asignarlos y añadirlos al modelo de objetos.

En el ejemplo de los cajeros automáticos, son interesantes:

- Comprobar contraseña (método de *Autorización de la tarjeta*).
- Actualizar cuenta (método de la clase *Cuenta*).

Se añadirán otros métodos, no relacionados con ninguna de las cuestiones anteriores, procedentes de nuestro conocimiento del tema:

- Cerrar una cuenta (sobre *Cuenta*).
- Autorizar una tarjeta de crédito (sobre *Cuenta*, parámetro la *Autorización de la tarjeta*).
- Crear una cuenta (sobre *Banco*, parámetro el *Cliente*).
- Crear una tarjeta de crédito (sobre *Banco*, parámetro el *Cliente*).
- Cerrar una autorización (sobre *Autorización de la tarjeta*).