

# Programación III

## Clase II

*Andrés Fortier*

# Repaso

- Abstracción.
- Modelos.
- Paradigmas.
- Paradigma Orientado a Objetos
  - Todo es un objeto.
  - Los objetos colaboran enviándose mensajes.

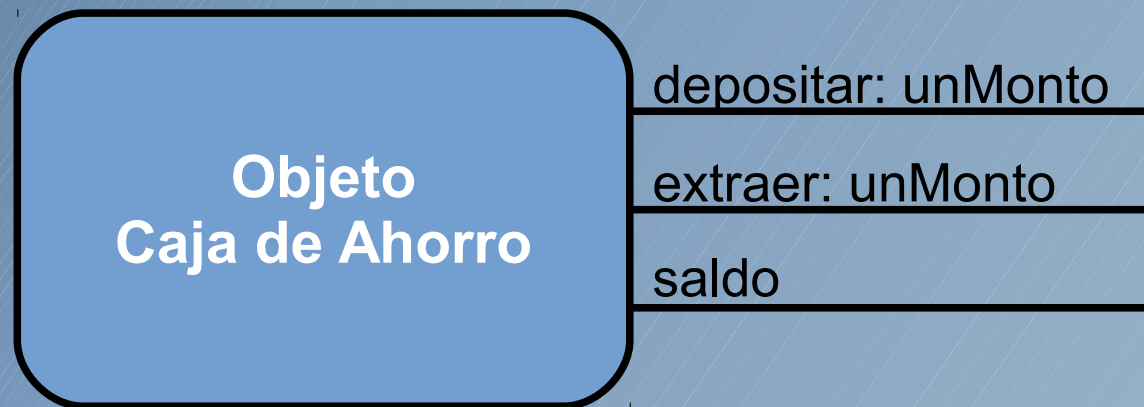
# Características de los objetos

- Comportamiento bien determinado
  - ¿Qué hace el objeto y cómo lo hace?
- Estado interno o estructura interna
  - El conjunto de variables de instancia.
- Identidad
  - ¿Cómo podemos distinguir un objeto de otro?



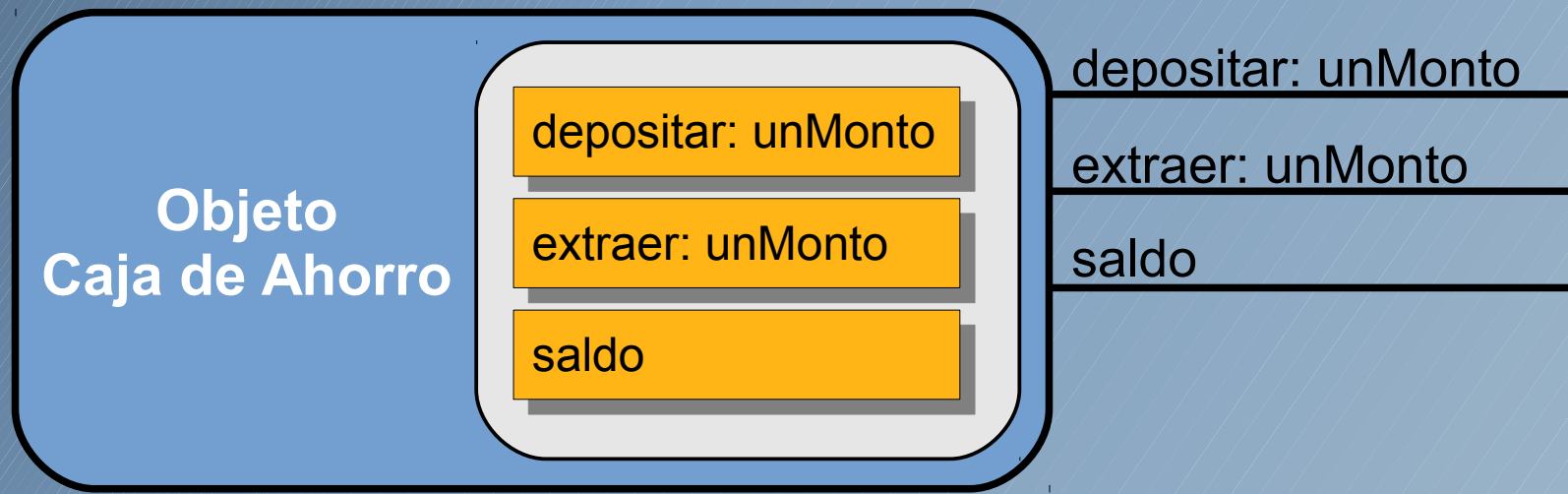
# Comportamiento - ¿Qué hace un objeto?

- Un objeto se define en términos de su comportamiento.
- El comportamiento indica qué sabe hacer el objeto. (cuáles son sus responsabilidades).
- Se especifica a través del conjunto de mensajes que el objeto sabe responder: protocolo.
- Ejemplo:



# Comportamiento - ¿Cómo lo hace?

- La implementación indica cómo hace el objeto para responder a sus mensajes.
- Se especifica a través de un conjunto de métodos.
- Es privada del objeto. Ningún otro objeto puede acceder a ella.

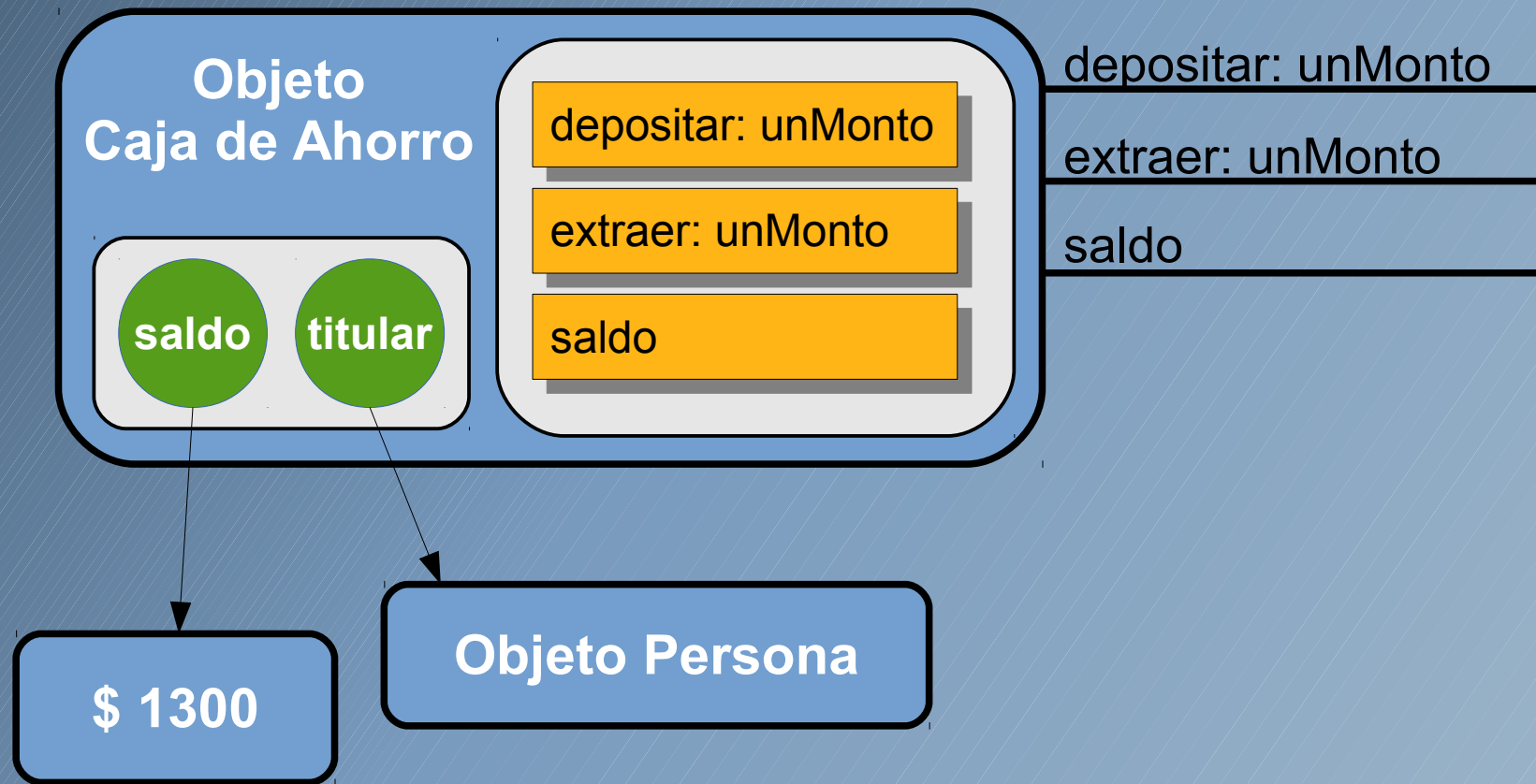


# Estado interno

- Está compuesto por las variables de instancia del objeto.
- Las v.i. hacen referencia a otros objetos con los cuales puede colaborar para llevar a cabo sus responsabilidades.
- Es privado del objeto. Ningún otro objeto puede acceder a él.



# Estado interno



# Identidad

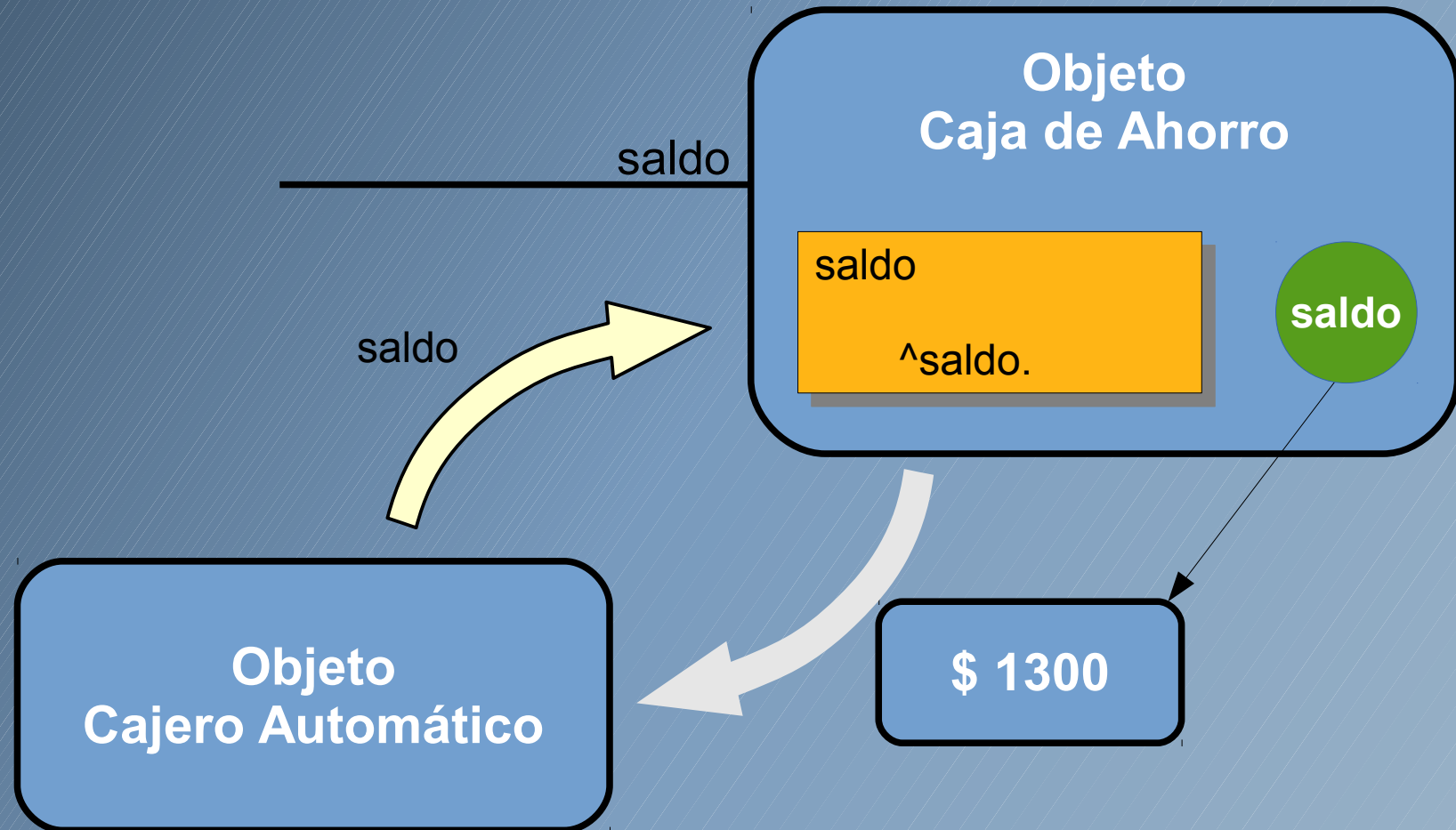
- Es una propiedad intrínseca de los objetos.
- Un objeto sólo es idéntico a sí mismo.
- No es lo mismo que la igualdad entre objetos
  - Dos objetos pueden tener las mismas propiedades pero no son el mismo objeto.



# Envío de un mensaje

- El comportamiento de un objeto está definido en términos de los mensajes que éste entiende.
- Para poder enviarle un mensaje a un objeto, hay que conocerlo.
- Al enviarle un mensaje a un objeto, éste responde activando el método asociado a ese mensaje (siempre y cuando exista).
- Como resultado del envío de un mensaje el objeto receptor puede retornar otro objeto.

# Enviar un mensaje y retornar un objeto



# Especificación de un mensaje

- ¿Cómo se especifica un mensaje?
  - Se le asigna un nombre.
  - El nombre debe tener significado en el dominio de problema.
  - Se indica cuáles son los parámetros requeridos para resolver el mensaje.
- Cada lenguaje de programación propone una sintaxis particular para indicar el envío de un mensaje. En la materia vamos a utilizar la sintaxis Smalltalk.
  - unaCajaDeAhorro saldo
  - unaCajaDeAhorro depositar: 100
  - unCajero depositar: 100 en: unaCajaDeAhorro



# Métodos

- ¿Qué es un método?
  - Es la contraparte funcional del mensaje.
  - Expresa la forma de llevar a cabo la semántica propia de un mensaje particular (el *cómo*).
- Un método puede realizar básicamente 3 cosas:
  - Modificar el estado interno del objeto.
  - Colaborar con otros objetos enviando mensajes.
  - Retornar y terminar.

# Especificación de un Método

Nombre del método

Parámetros

El comentario describe la semántica del mensaje

Variables temporales

```
nombreMetodo: parametro1 con: parametro2  
  "Comentario del método"
```

```
| variableTemporal |
```

```
variableDeInstancia := parametro1.
```

```
variableTemporal := parametro2.
```

```
^variableTemporal haceAlgoCon: self.
```

El cuerpo del método consiste en la resolución del comportamiento

# Ejemplos de métodos

depositar: unMonto

*“Agrega unMonto al saldo actual de la cuenta”*

saldo := saldo + unMonto

asignarDeposito: unMonto a: unNumeroDeCuenta

*“Deposita unMonto en la cuenta asociada  
a unNumeroDeCuenta”*

| cuenta |

cuenta := banco obtenerCuentaConNumero: unNumeroDeCuenta.  
cuenta depositar: unMonto.



# Comportamiento y estado interno

- Dijimos que el comportamiento tiene una implementación privada.
- También que el estado interno es privado.
- ¿De qué hablamos cuando hablamos de privacidad?

# Encapsulamiento

*“Es la cualidad de los objetos de ocultar los detalles de implementación y su estado interno del mundo exterior”*

- Características:
  - Esconde detalles de implementación.
  - Protege el estado interno de los objetos.
  - Un objeto solo muestra su “cara visible” por medio de su protocolo.
  - Los métodos y su estado quedan escondidos para cualquier otro objeto. Es el objeto quien decide qué se publica.
  - Facilita modularidad y reutilización.

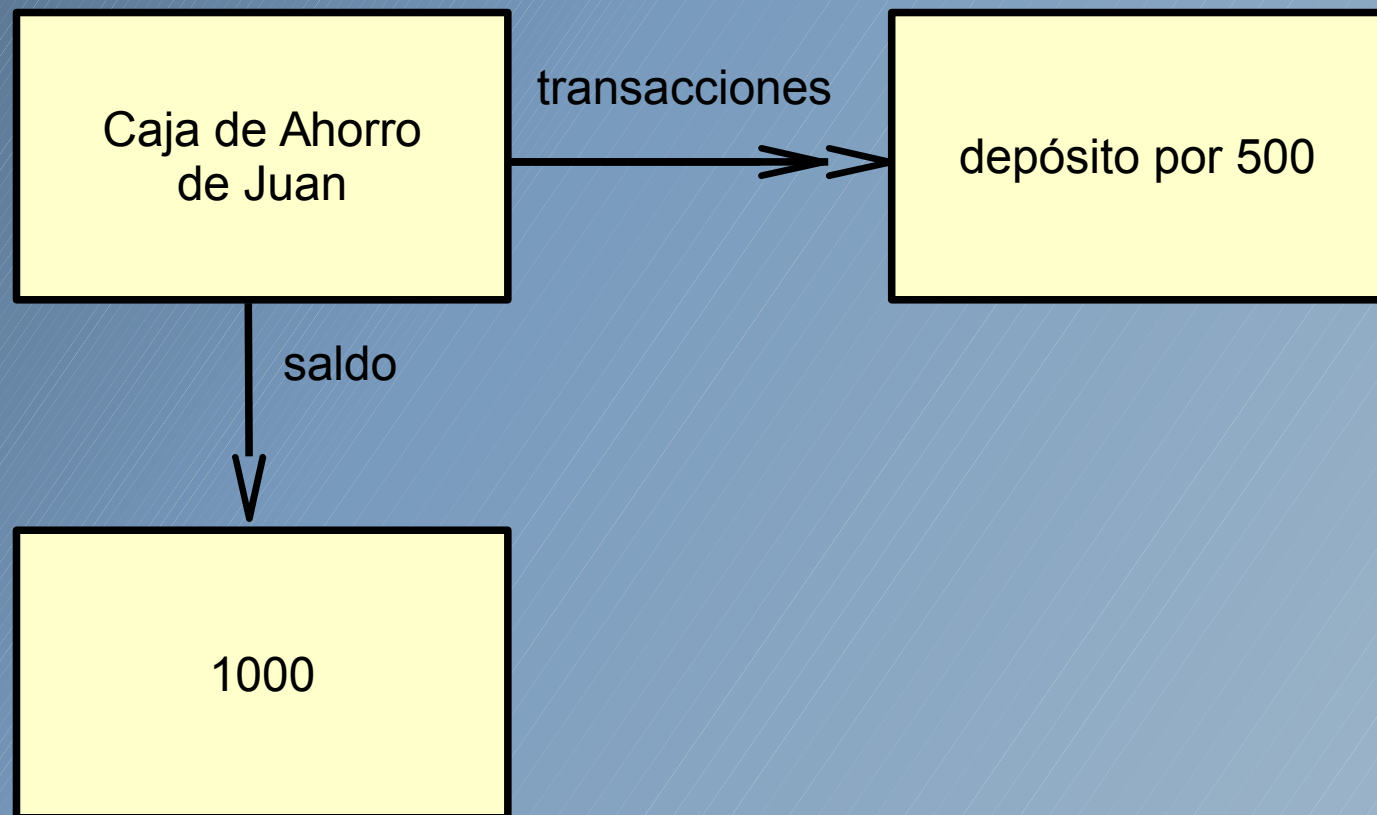
# Encapsulamiento - importancia en la POO

- Característica fundamental del paradigma OO.
  - Motiva a que el acoplamiento entre objetos sea bajo.
  - Permite que el software escale mejor frente a cambios.
- Un objeto debe saber lo mínimo indispensable sobre los objetos que conoce.
  - De esta manera, los cambios internos no impactan en los otros objetos del sistema.
- Programar en términos del protocolo:
  - Representación interna de los objetos tiende a cambiar.



# Diagramas de Instancia (u Objetos)

- Representa una vista estática del modelo.
- Se incluyen los objetos y sus relaciones de colaboración.



# Diagramas de Secuencia (o Colaboración)

- Representa una vista dinámica del modelo.
- Muestra la secuencia de envío de mensajes para llevar a cabo una tarea.

# Diagramas de Secuencia (o Colaboración)

