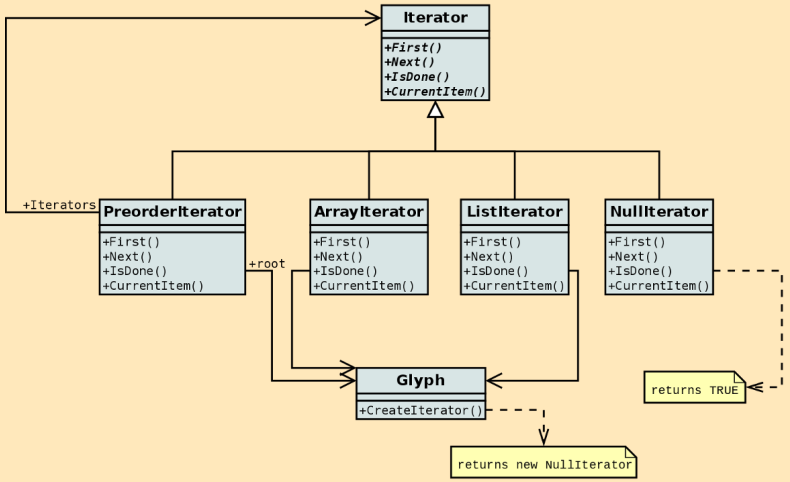
WEBMASTER II

UML
por
Andros Fenollosa

¿Qué es UML



UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

¿Por qué necesitamos UML?

- **Visualizar**: UML permite expresar de forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar**: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir**: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar**: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

¿Cómo nació?

- 1994. James Burmbaugh se une a la compañía **Rational Software** (fundada por Grady Booch), con el objetivo de unificar 2 métodos que habían desarrollado sobre la metodología del software: el **método Booch** y el **OMT** (Object Modelling Tool).
- 1995. Se publica el primer borrador. Ivar Jacobson se une a la compañía y añade ideas propias. A partir de este momento serán conocidos como los "*tres amigos*".
- 1997. Nace la primera versión (UML 1.0). Es ofrecida a un grupo de trabajo para que lo convierta en un estándar. Éste grupo, con gran experiencia en estándares relacionados con la tecnología orientada a objetos, proponen cambios. En noviembre se libera UML 1.1 .
- 2005. Se declara como estándar por la ISO.
- Actualidad. Se continúa ampliando y actualizando, a día de hoy UML 2.5 .

Tipos de diagramas

- Dinámicos
  - Diagrama de casos de uso.
  - Diagrama de colaboración.
  - Diagrama de secuencia.
  - Diagrama de estados.
  - Diagrama de actividades.
- Estáticos
  - Diagrama de clases.
  - Diagrama de objetos.
  - Diagrama de componentes.
  - Diagrama de despliegue.

Casos de uso - ¿Qué es?

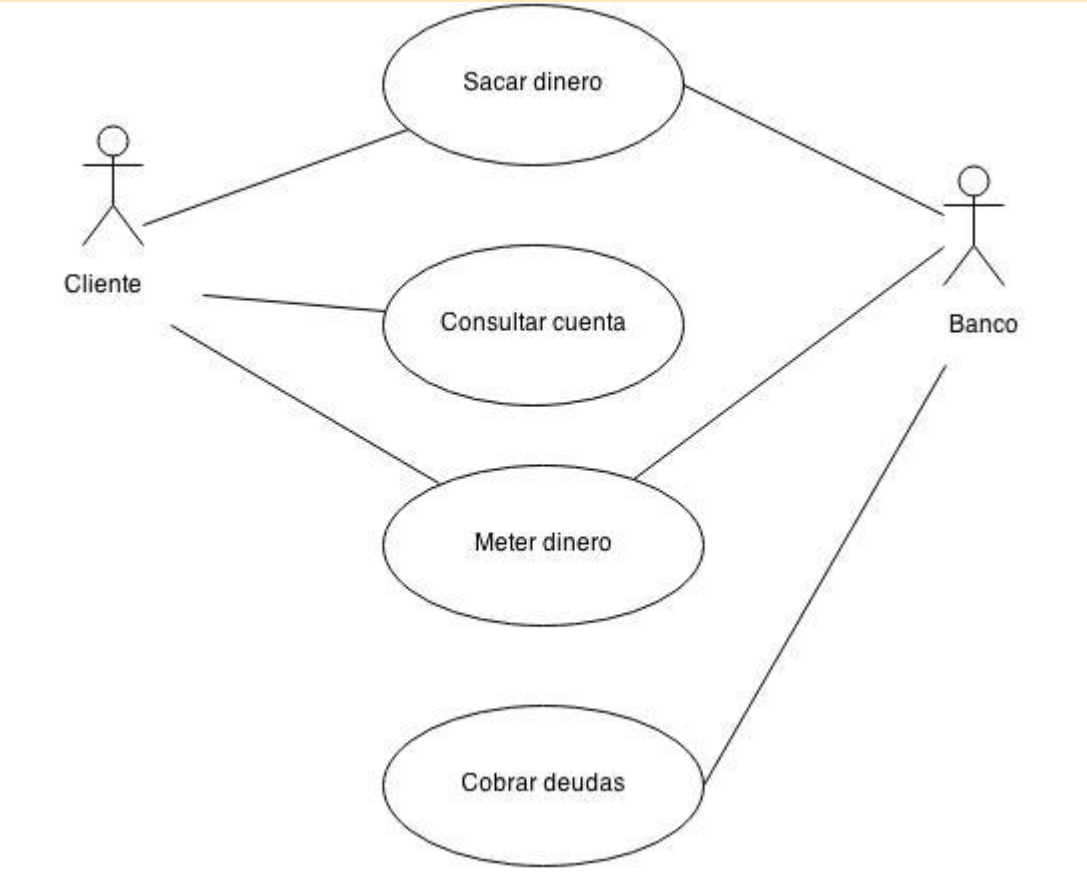
Los casos de uso es un diagrama para mostrar el comportamiento final de forma sencilla.

Esta orientado para el cliente y como herramienta para encontrar nuevas funcionalidad.

Casos de uso - ¿Por qué?

- Identificación de actores.
- Primer paso para otros diagramas.
- Centralizar diferentes visiones.

Casos de uso - Ejemplo



Captura de requisitos - ¿Qué es?

La captura de requisitos es una lista de necesidades para entender el contexto del sistema con la intención de capturar todos sus requisitos funcionales y no funcionales.

Captura de requisitos - ¿Por qué?

- Es fácil.
- Usuarios no saben qué quieren.
- Construir sistema correcto.
- Usar lenguaje sencillo para los usuarios.

Captura de requisitos - Ejemplo

Caso de uso: Sacar dinero
Actores: Cliente
Descripción: Un cliente llega al cajero de un banco con su tarjeta. La introduce, se identifica con su PIN, y saca dinero.

Secuencia de eventos típica

Acciones de actores	Acciones de sistema
1. El cliente llega al cajero.	
2. El cliente casa su tarjeta y la introduce en el cajero	
	3. Solicita el código PIN al cliente.
4. Introduce el código PIN	
	5. Valida el código PIN con el banco.
	6. Solicita la cantidad de dinero a retirar al cliente.
7. Introduce la cantidad a retirar.	
	8. Autoriza la entrega de la cantidad con el banco.
	9. Entrega la cantidad al cliente.
10. El cliente retira su tarjeta y la cantidad entregada por el cajero.	

Captura de requisitos - Ejemplo

Secuencias de eventos alternativas
Acción 3: Si el cajero no puede leer la tarjeta del cliente muestra un mensaje 'Tarjeta incorrecta' al usuario y devuelve la tarjeta al cliente. **Acción 5: Si el PIN introducido es incorrecto muestra el mensaje 'PIN incorrecto' y vuelve a la acción 4.** **Acción 8: Si la cantidad solicitada es mayor al saldo del cliente o a la máxima permitida, muestra un mensaje 'Cantidad no permitida' y vuelve a la acción 7.**

Diagrama de secuencia - ¿Qué es?

El diagrama de secuencia es un esquema de cómo se comunica los diferentes objetos entre sí y cuáles son sus llamadas para hacer una determinada labor.

Diagrama de secuencia - ¿Por qué?

- Minimizar la labor del programador.
- Y por lo tanto, desarrollar menos errores.
- Ver como se mueve la información.
- Ayudar a realizar futuros cambios.
- Optimizar.

Diagrama de secuencia - Ejemplo

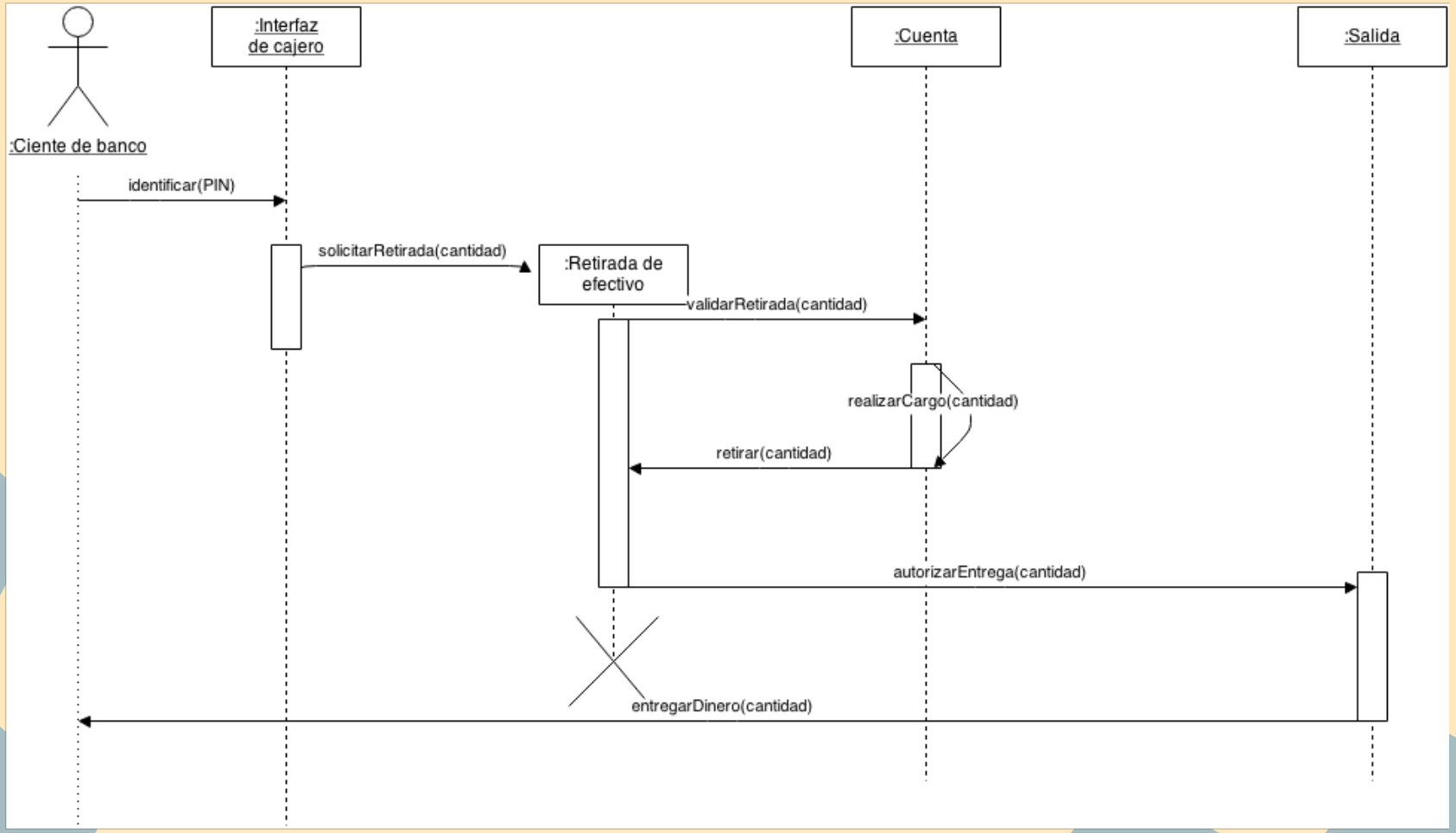


Diagrama de clases - ¿Qué es?

El diagrama de clases es una representación visual de los atributos y clases que se van a implementar.

Diagrama de clases - ¿Por qué?

- Sabes exactamente que debe tener cada clase.
- Mayor control.
- Documentación.
- Maravilloso paso para la hora de implementar.

Diagrama de clases - Ejemplo

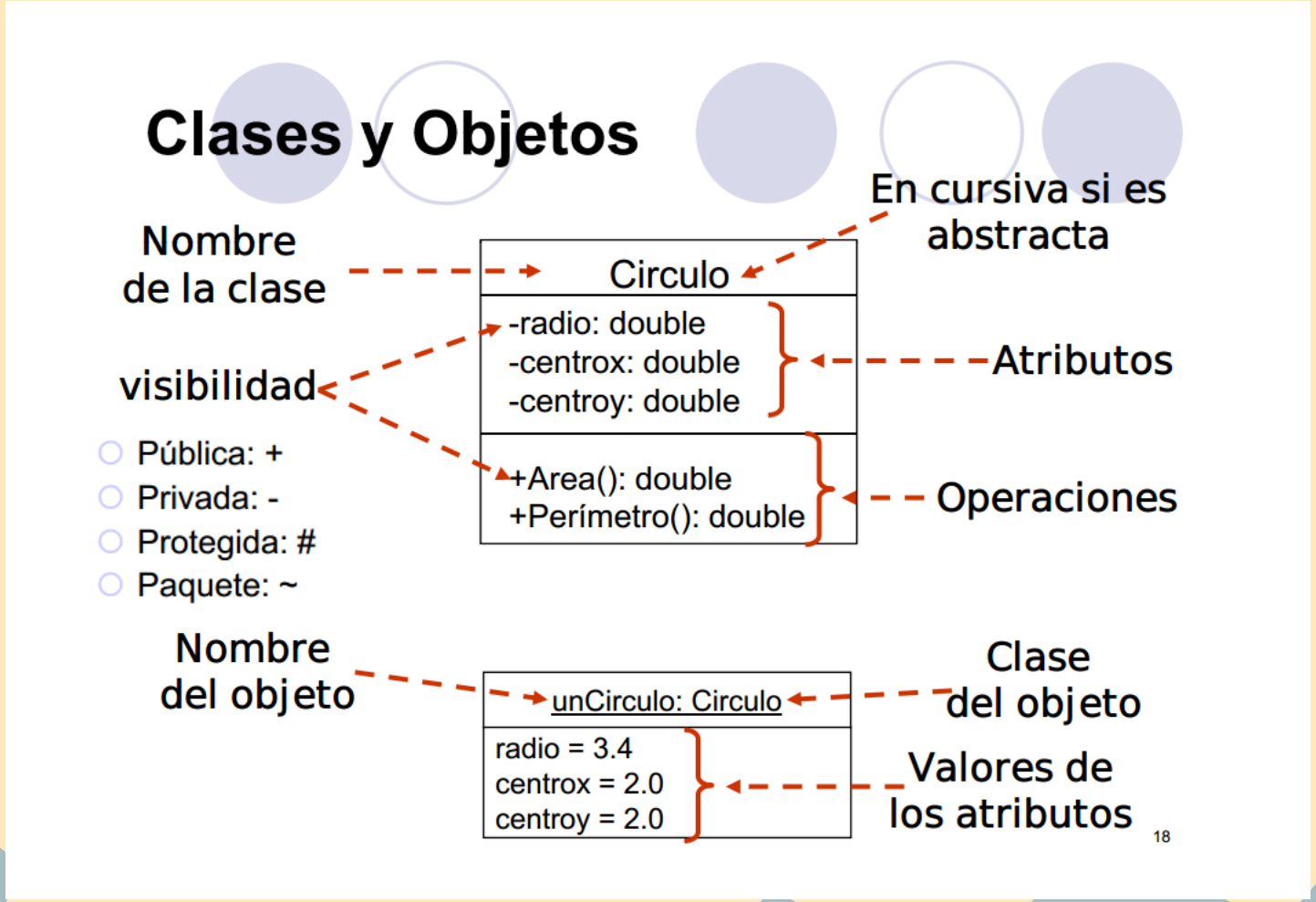


Diagrama de clases - Ejemplo