

# INTRODUCCIÓN AL ANÁLISIS ORIENTADO A OBJETOS

## INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2024/2025

Francisco José García-Peñalvo / [fgarcia@usal.es](mailto:fgarcia@usal.es)

Alicia García-Holgado / [aliciagh@usal.es](mailto:aliciagh@usal.es)

Andrea Vázquez-Ingelmo / [andreavazquez@usal.es](mailto:andreavazquez@usal.es)

Departamento de Informática y Automática  
Universidad de Salamanca



A person in a white shirt and blue jeans is holding a large, silver megaphone to their mouth, as if shouting or announcing something. The background is a light, neutral color.

## MÁS INFORMACIÓN

**Tema 6 – Flujos de trabajo del Proceso Unificado [1]**

**Tema 7 – Análisis orientado a objetos [2]**

## PÍLDORAS DE VÍDEO RELACIONADAS

**Análisis Orientado a Objetos [3]**

**Análisis Orientado a Objetos en el Proceso Unificado –  
Consejos prácticos [4]**

**Máquina de reciclado [5]**

# ÍNDICE

- Introducción
- Análisis Orientado a Objetos
- Análisis en el Proceso Unificado



# 1. INTRODUCCIÓN

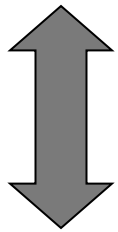
# ¿DE DÓNDE SE PARTE?



# ¿DÓNDE SE QUIERE LLEGAR?



Usuarios, clientes...



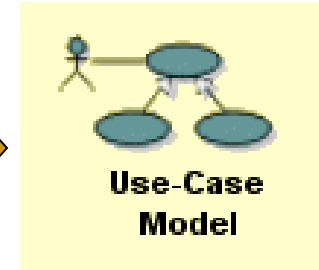
Desarrolladores



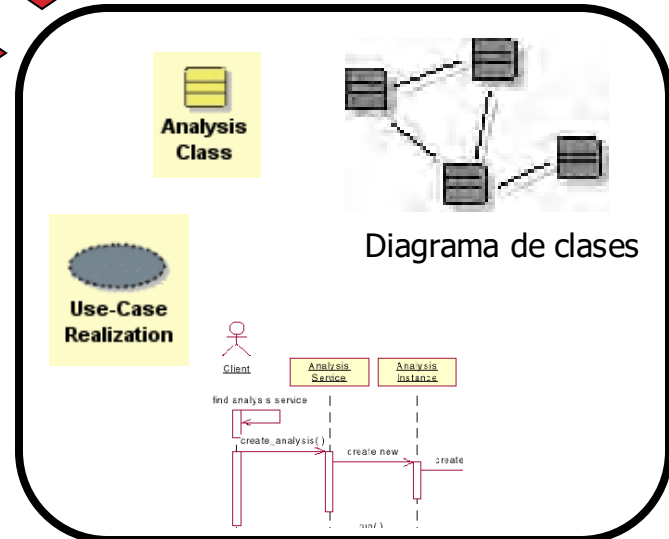
Requisitos



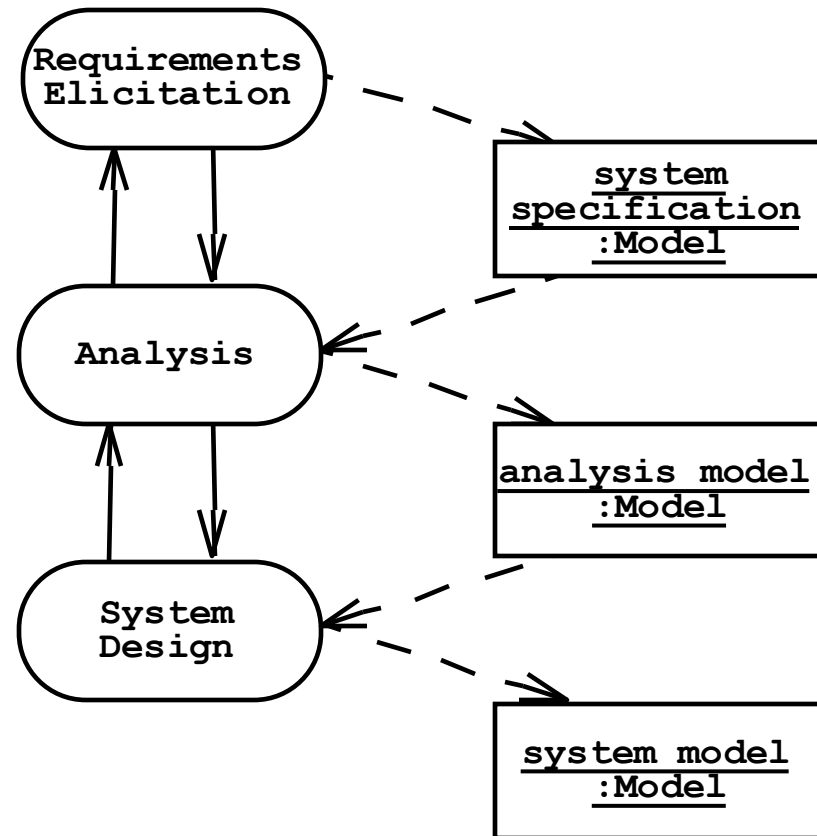
Escenarios /  
casos de uso



Use-Case  
Model



# PRODUCTOS DE LA RECOGIDA Y ANÁLISIS DE REQUISITOS



Bruegge y Dutoit [6]

# DEFINICIÓN DE ANÁLISIS

- En términos generales se define análisis como “la distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos” (RAE) [7]
- Desde un punto de vista informático se define análisis como “el estudio, mediante técnicas informáticas, de los límites, características y posibles soluciones de un problema al que se aplica un tratamiento por ordenador” (RAE) [7]
- Para la Ingeniería del *Software* el análisis es la parte del proceso de desarrollo de *software* cuyo propósito principal es realizar un modelo del dominio del problema
  - Se puede definir más precisamente como “el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de *hardware* o de *software*, así como el proceso de estudio y refinamiento de dichos requisitos” (IEEE) [8]

# EL OBJETO DEL ANÁLISIS ORIENTADO A OBJETOS

Analizar los requisitos en la forma de un modelo de análisis es importante por varios motivos [9]

- Un modelo de análisis ofrece una especificación más precisa de los requisitos que la que se tiene como resultado de la captura de requisitos, incluyendo al modelo de casos de uso
- Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores y puede, por tanto, introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema
- Un modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y, en general, su mantenimiento
- Un modelo de análisis puede considerarse como una primera aproximación al modelo de diseño y es, por tanto, una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación



<https://pixabay.com/es/mindmap-lluvia-de-ideas-idea-2123973/>

## 2. ANÁLISIS ORIENTADO A OBJETOS

# DEFINICIÓN

**El análisis orientado a objetos es el proceso que modela el dominio del problema mediante la identificación y la especificación de un conjunto de objetos semánticos que interaccionan y se comportan de acuerdo con los requisitos del sistema**

**(Monarchi y Puhr) [10]**

- Permite describir el sistema en los mismos términos que el mundo real
- Se centra en la comprensión del espacio (dominio) del problema
- Contiene elementos de síntesis
- La abstracción de requisitos de usuario y la identificación de los objetos clave del dominio es seguida del ensamblaje de estos objetos en estructuras de forma que soporten el diseño en fases posteriores

# GENERALIDADES

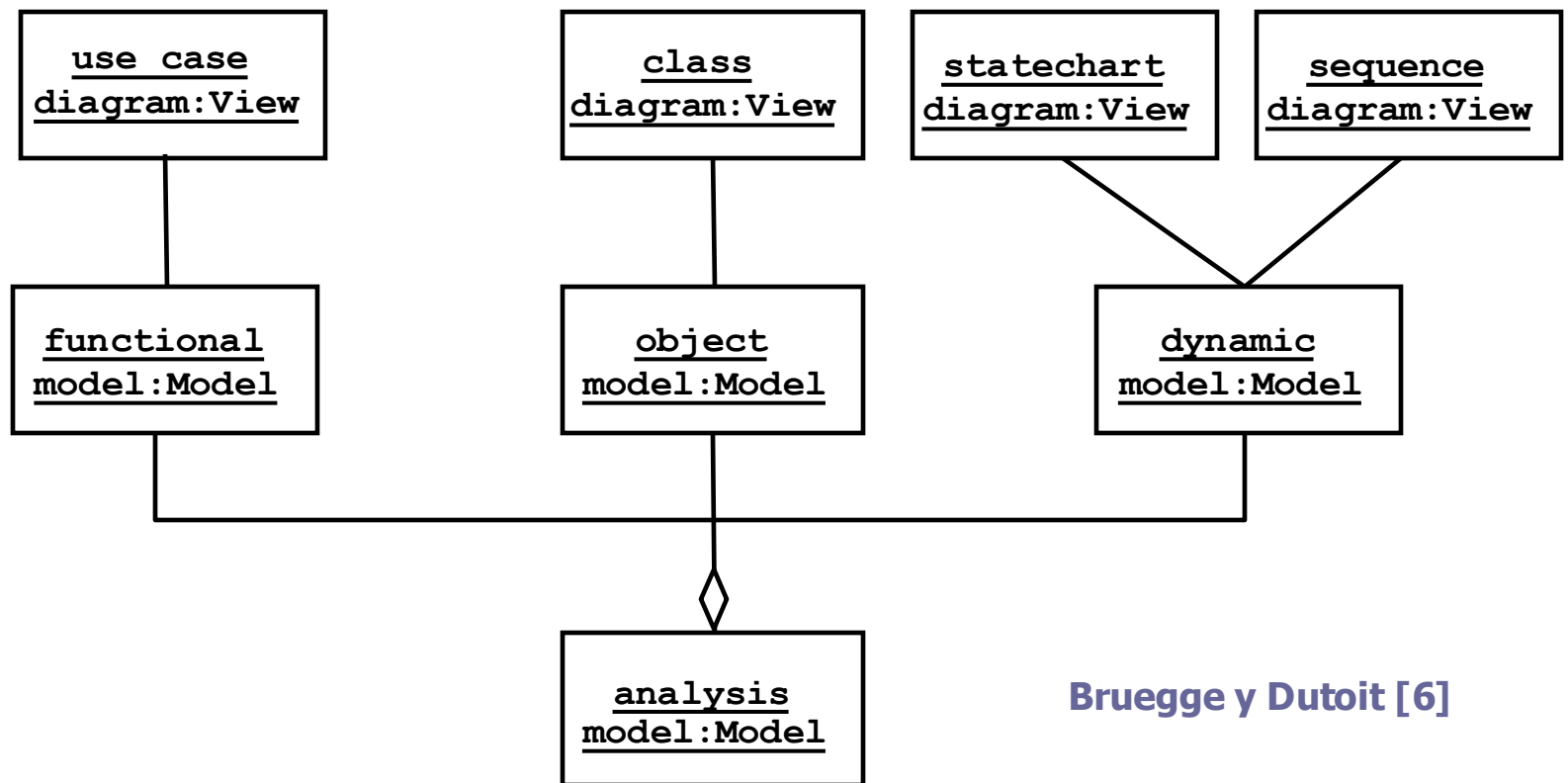
- Difícil determinar dónde acaba el análisis orientado a objetos y dónde comienza el diseño orientado a objetos
- El objetivo es modelar la semántica del problema en términos de objetos distintos pero relacionados
- El análisis casa con el dominio del problema
- Los objetos del dominio del problema representan cosas o conceptos utilizados para describir el problema (**objetos semánticos**) y se representan en el modelo de dominio [11]
- Los objetos del dominio del problema tienen una equivalencia directa en el entorno de la aplicación
- Se centra en la representación del problema
  - Identificar abstracciones que contengan el significado de las especificaciones y de los requisitos

# GENERALIDADES

- El modelo de casos de uso identifica secuencias de eventos e interacciones entre actores y el sistema
- El **modelo de análisis** especifica las clases de objetos que se encuentran o existen en el sistema
- No existen reglas fijas para esta transformación
- Se centra en la elaboración de un modelo del sistema, el modelo de análisis
  - Modelo funcional
    - Representado por los casos de uso
  - Modelo objeto análisis
    - Representado por los diagramas de clase y objetos
  - Modelo dinámico
    - Representado por los diagramas de secuencia y los diagramas de transición de estados

# ESTRUCTURA DEL MODELO DE ANÁLISIS

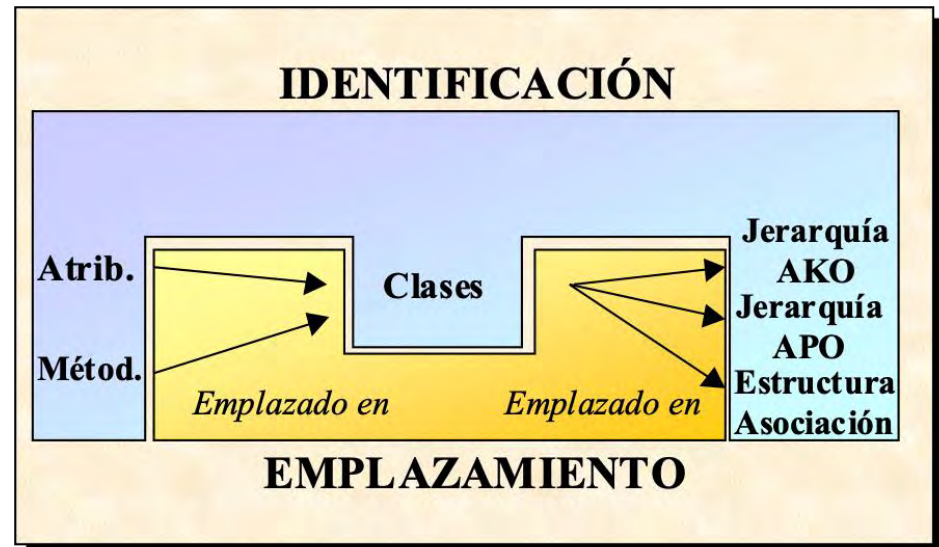
El **Modelo de Análisis** estructura el sistema independientemente del entorno actual de implementación



Bruegge y Dutoit [6]

# ACTIVIDADES DEL ANÁLISIS ORIENTADO A OBJETOS

- La identificación de las clases semánticas, los atributos, el comportamiento y las relaciones (generalizaciones, agregaciones y asociaciones)
- El emplazamiento de las clases, atributos y comportamiento
- La especificación del comportamiento dinámico mediante paso de mensajes



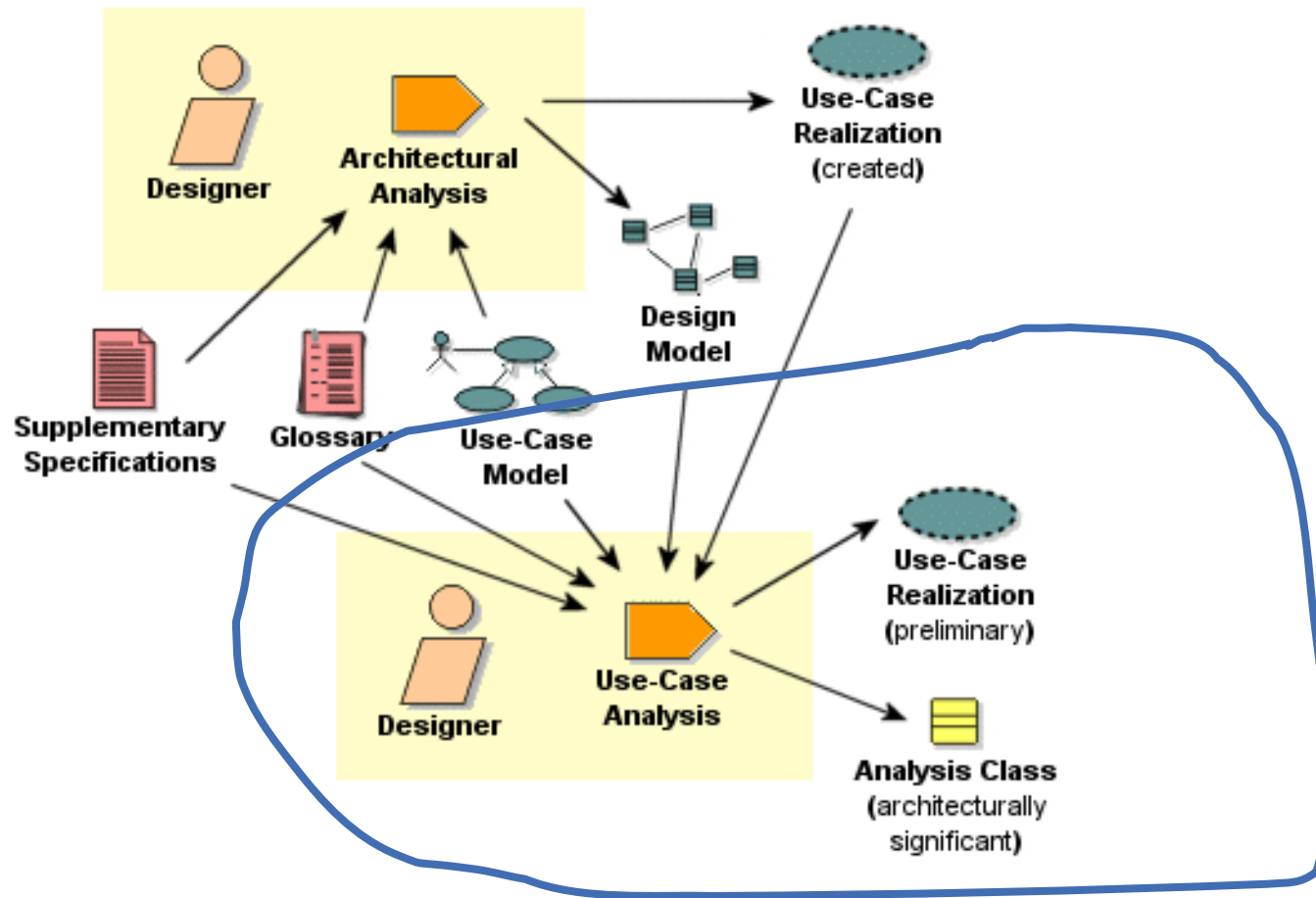
# TIPOS DE PROCESO EN ANÁLISIS

- Existen diferentes enfoques de proceso en el análisis
  - Centran en la información (datos) del sistema
  - Centran en la funcionalidad (comportamiento) del sistema
  - Síntesis de los dos procesos anteriores
- El Proceso Unificado [9] sigue el enfoque de síntesis
  - Inicio por la funcionalidad (**casos de uso**)
  - Refinamiento por la información (**diagramas de clases**)
  - Consolidación por la funcionalidad (**diagramas de secuencia /colaboración**)



### **3. ANÁLISIS EN EL PROCESO UNIFICADO**

# ANÁLISIS EN EL PROCESO UNIFICADO



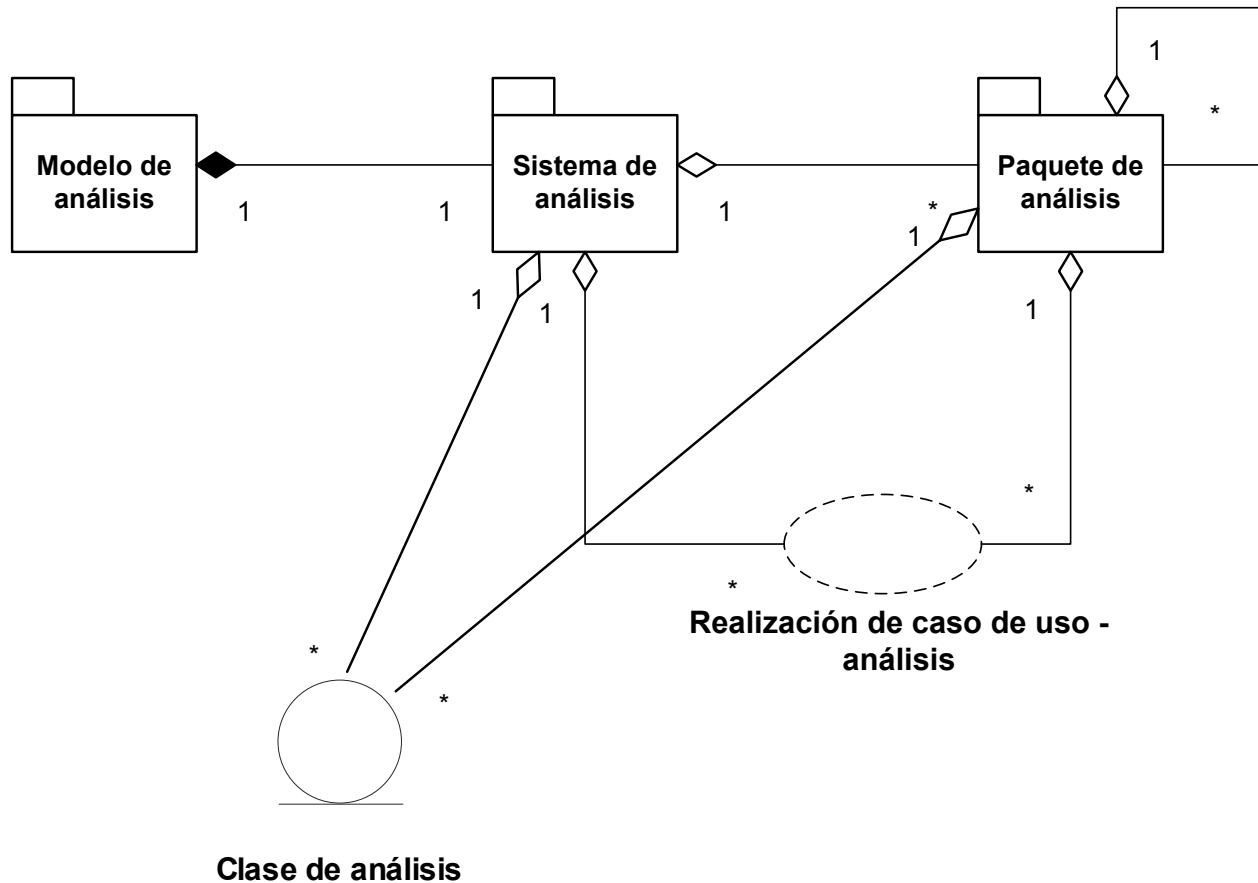
# ARTEFACTOS PROPIOS DEL ANÁLISIS EN EL PROCESO UNIFICADO

- Modelo de análisis
- Clase de análisis
- Realización de casos de uso – análisis
- Paquete de análisis
- Descripción de la arquitectura (vista del modelo de análisis)

# MODELO DE ANÁLISIS

- Se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo
- Se utilizan otros paquetes de análisis para organizar el modelo de análisis en partes más manejables
  - Representan abstracciones de subsistemas y posiblemente capas completas del diseño del sistema
- Las clases de análisis representan abstracciones de clases y posiblemente de subsistemas de diseño del sistema
- Los casos de uso se describen mediante clases de análisis y sus objetos
  - Esto se representa mediante colaboraciones denominadas **realizaciones de caso de uso - análisis**

# MODELO DE ANÁLISIS [9]



# CLASE DE ANÁLISIS

- Representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema
- Características de las clases de análisis
  - Se centran en el tratamiento de los requisitos funcionales
  - Son más evidentes en el contexto del dominio del problema
  - Raramente definen u ofrecen una interfaz en términos de operaciones
  - Pueden definir atributos, pero a un nivel bastante alto
  - Participan en relaciones de un claro carácter conceptual
  - Siempre encajan en uno de tres estereotipos básicos
    - **Entidad**
    - **Interfaz**
    - **Control**

# CLASE DE ANÁLISIS

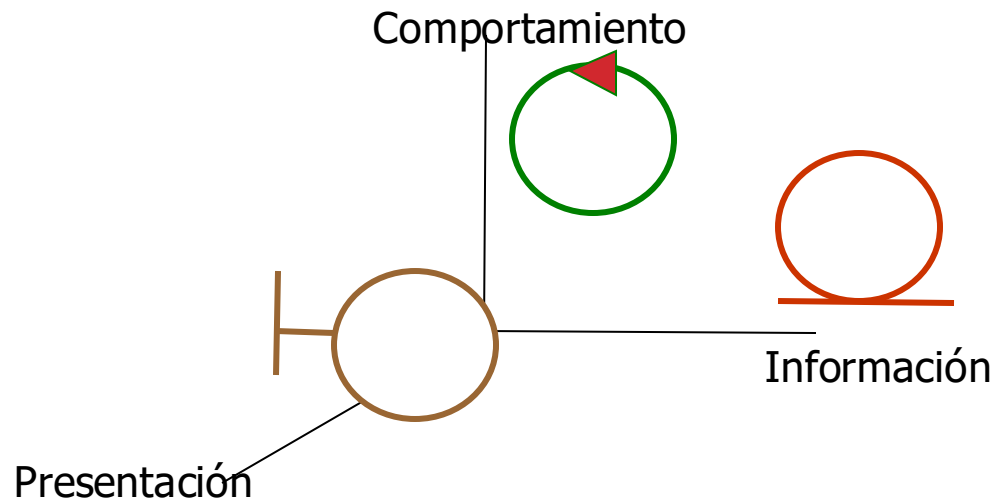
Especificación de objetos en el espacio de información definido por tres ejes

Se eligen tres tipos de objetos a fin de proporcionar una estructura más adaptable

**Objetos Entidad:** Información persistente sobre la que el sistema realiza un seguimiento

**Objetos Interfaz:** Representan las interacciones entre el actor y el sistema

**Objetos Control:** Representan las tareas realizadas por el usuario y soportadas por el sistema

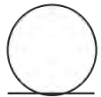


# CLASE DE ANÁLISIS

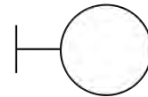
- ¿Por qué estos tres tipos de objetos (entidad, interfaz y control)?
  - Aparecen de forma natural en el texto del caso de uso
  - Se obtienen objetos especializados más pequeños
  - Dan lugar a modelos más resistentes a cambios
    - La interfaz cambia fácilmente
  - Ayudan a la construcción de diagramas de secuencia
  - Los objetos de control sirven de conexión entre los usuarios y los datos almacenados
    - Capturan las reglas de negocio (siempre sujetas a cambios)
    - Sirven de espacio de reserva para garantizar que no se olvida la funcionalidad

# CLASE DE ANÁLISIS

## Notación



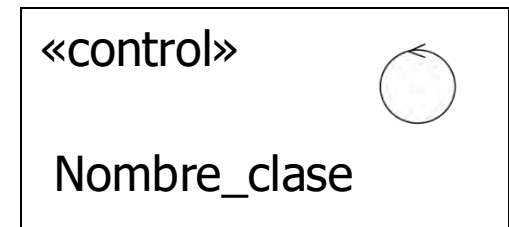
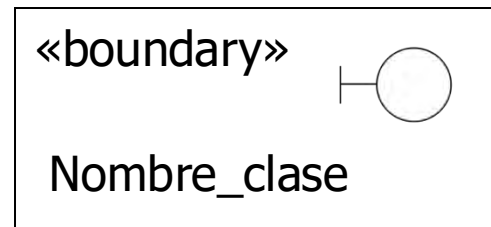
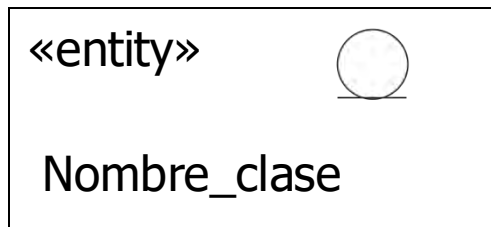
Clase de entidad



Clase de interfaz



Clase de control



# CLASE DE ENTIDAD

- Para modelar la información que el sistema tiene que gestionar se utilizan los objetos entidad
- Esta información permanece en el sistema incluso cuando el caso de uso finaliza
- En el objeto entidad se incorpora, además de la información, el comportamiento asociado a esa información
- Estos objetos se identifican a partir de la descripción de los casos de uso
- Normalmente suelen corresponder a alguno de los conceptos que se manejan en el sistema. **Clases conceptuales**
- Las operaciones identificadas en el objeto tienen que ser suficientes para todas las posibles utilidades del objeto
- Hay que evitar en la medida de lo posible que estos objetos solo sean portadores de información asignando todo el comportamiento dinámico a los objetos control

# CLASE DE INTERFAZ

- Toda la funcionalidad que depende directamente del entorno se asigna a las clases de interfaz
- Modelan las interacciones entre el sistema y sus actores
- Interaccionan con los actores externos al sistema y con las clases del sistema
- Los objetos interfaz se encargan de trasladar las acciones de los actores a eventos en el sistema y estos en “información” que se presenta al actor
- Representan una abstracción de elementos de la interfaz de usuario (ventanas, formularios, paneles...) o dispositivos (interfaz de impresora, sensores, terminales...)
- Cada actor necesita su/s propia/s interfaz/ces para llevar a cabo sus acciones
- Mantener la descripción a nivel conceptual, es decir, no describir cada botón, ítem de menú... de la interfaz de usuario
- Encapsula y aísla los cambios en la interfaz de usuario

# CLASE DE INTERFAZ

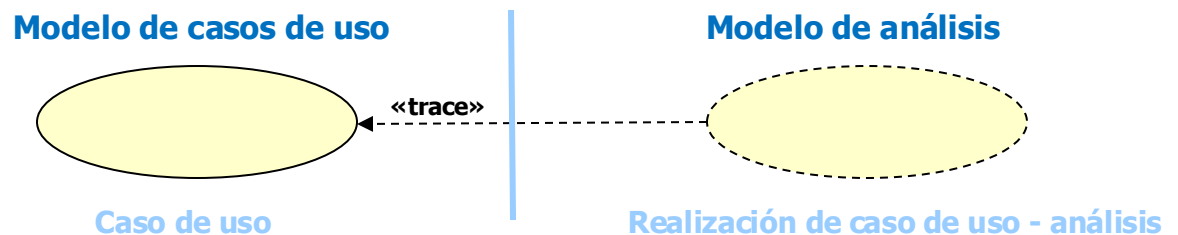
- Estrategias de identificación de clases de interfaz
  - A partir de las descripciones de los casos de uso
  - A partir de las descripciones de las interfaces de usuario
  - A partir de los actores
- Recomendaciones
  - Identificar formularios y ventanas necesarias para la introducción de datos en el sistema
  - Identificar avisos y mensajes a los que tiene que responder el usuario
  - No modelar los aspectos visuales de la interfaz
  - Utilizar siempre términos de usuario para describir las interfaces
- Identificar una **clase de interfaz central** para cada actor humano
  - Representa la “ventana” primaria con la que el actor interactúa
- Identificar una **clase de interfaz central** para cada actor sistema externo
  - Representa la interfaz de comunicación con el sistema externo
- Las clases de interfaz centrales pueden ser **agregaciones** de otras clases de interfaz

# CLASE DE CONTROL

- Los objetos de control actúan como unión entre los objetos interfaz y entidad
  - Coordinación entre objetos entidad e interfaz
- No siempre aparecen
  - Normalmente toda la funcionalidad expresada en un caso de uso está asignada a los objetos entidad e interfaz
- Se identifican a partir de los casos de uso
  - Cada caso de uso tiene inicialmente solamente un objeto de control
- El tipo de funcionalidad asignada a estos objetos suele ser el comportamiento relacionado con transacciones o secuencias específicas de control
- El comportamiento de “conexión” entre objetos interfaz y entidad suele asignarse a este tipo de objetos
- Normalmente no se corresponden con entidades reales

# REALIZACIÓN DE CASOS DE USO – ANÁLISIS

- Es una colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y se ejecuta un caso de uso en término de las clases de análisis y de sus objetos
- Ofrece una traza directa hacia un caso de uso concreto del modelo de casos de uso
- Una realización de un caso de uso posee
  - Una descripción del flujo de sucesos
  - Diagramas de clase de análisis
  - Diagramas de interacción

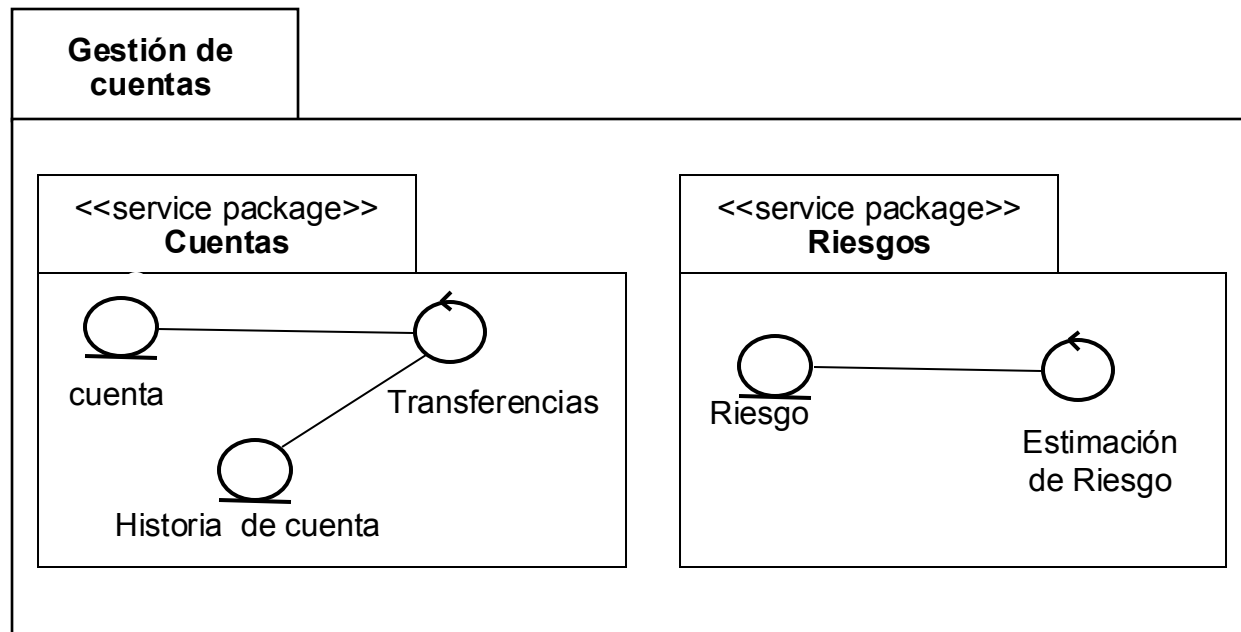


# PAQUETE DE ANÁLISIS

- Es el medio para organizar los artefactos del modelo de análisis en piezas manejables
- Puede constar de clases de análisis, de realización de casos de uso y de otros paquetes del análisis (recursivamente)
- Estos paquetes deben ser cohesivos y débilmente acoplados
- Pueden representar una separación de intereses de análisis
- Deben crearse basándose en los requisitos funcionales en el dominio del problema y deben ser reconocibles por las personas con conocimiento del dominio
- Se suelen convertir en subsistemas en las (dos) capas superiores del modelo de diseño

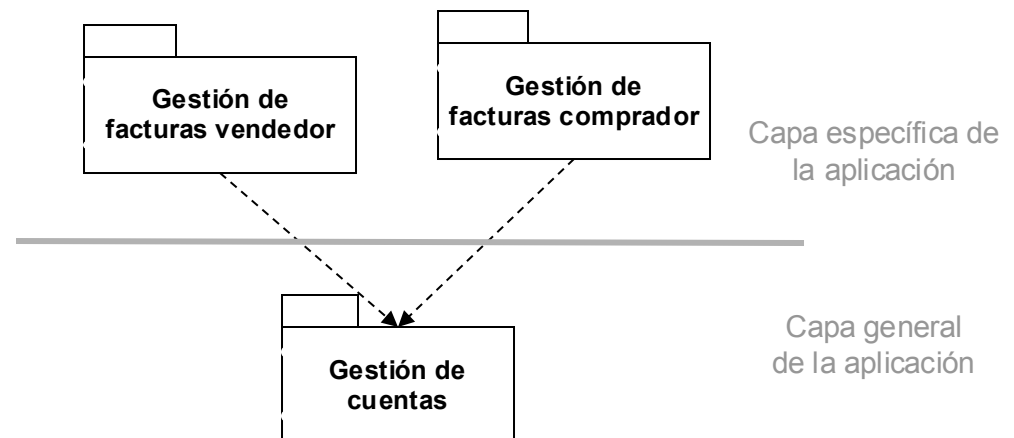
# PAQUETE DE ANÁLISIS

- Los paquetes de análisis pueden contener paquetes de servicio
- Los paquetes de servicio estructuran el sistema de acuerdo con los servicios que proporciona



# DESCRIPCIÓN DE LA ARQUITECTURA

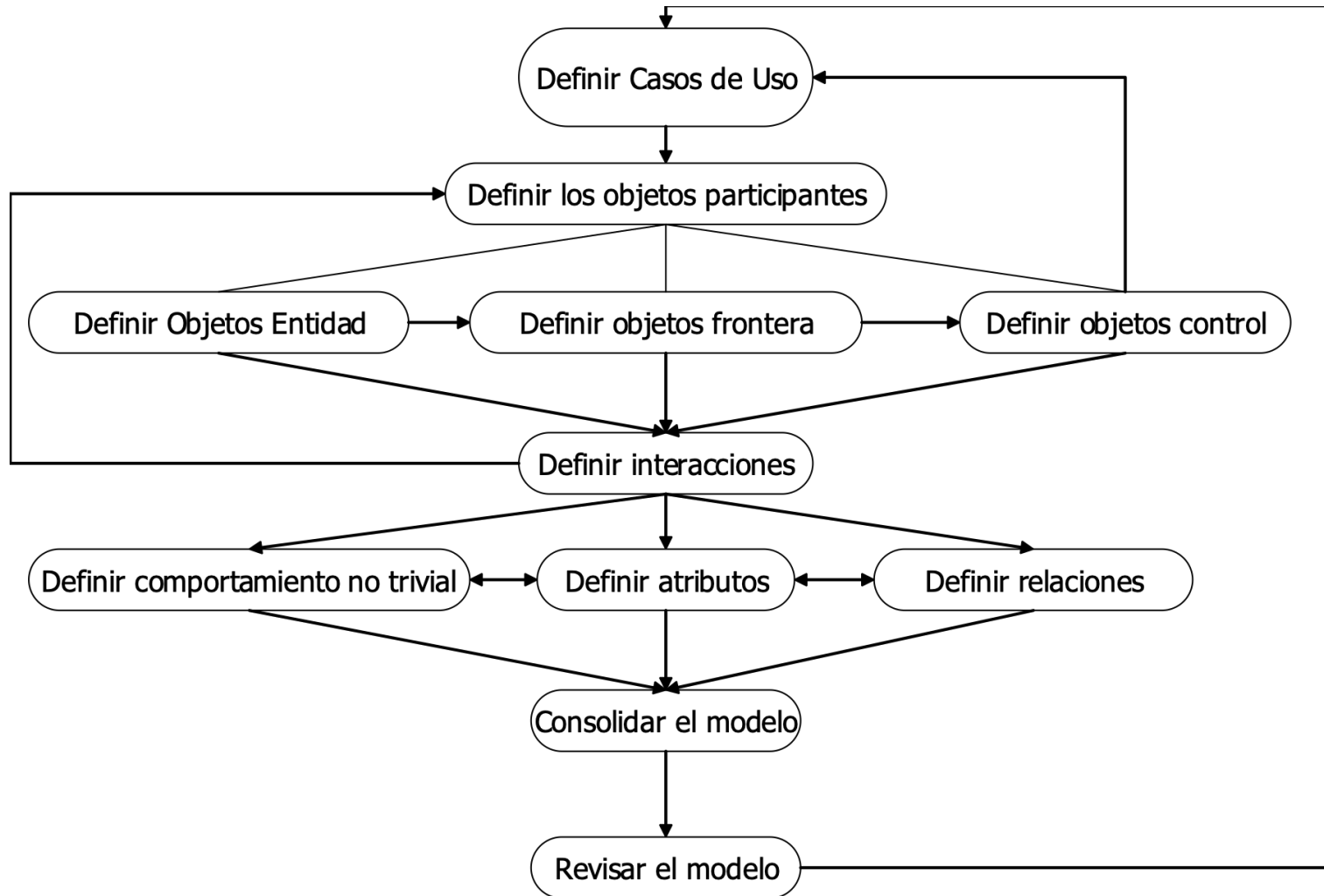
- Contiene una vista de la arquitectura del modelo de análisis, que muestra sus artefactos significativos para la arquitectura
- Descomposición del modelo de análisis en paquetes de análisis y sus dependencias
- Las clases fundamentales del análisis
- Realizaciones de casos de uso que describen cierta funcionalidad importante y crítica



# PROCESO UNIFICADO: ACTIVIDADES DEL ANÁLISIS

- Completar las descripciones de los casos de uso
  - Incluir los detalles internos de la actividad del sistema en respuesta a las acciones de los actores
- Para cada **realización de caso de uso**
  - Encontrar las **clases de análisis** a partir del comportamiento del caso de uso
  - Distribuir el comportamiento entre las clases de análisis
- Para cada clase de análisis resultante
  - Describir las responsabilidades
  - Describir atributos y asociaciones
  - Definir atributos
  - Establecer las asociaciones entre las clases de análisis
  - Describir Dependencias entre clases de análisis
- Evaluar el resultado del análisis de casos de uso

# PROCESO UNIFICADO: ACTIVIDADES DEL ANÁLISIS



# CONSEJOS PRÁCTICOS

Definir los objetos participantes

- **Para cada caso de uso** identificar los objetos que participan en cada caso de uso
- Conjunto posible de elementos del modelo (clases de análisis) capaz de llevar a cabo el comportamiento descrito en el caso de uso
- **Utilizar siempre términos del dominio**

# CONSEJOS PRÁCTICOS

## Definir interacciones

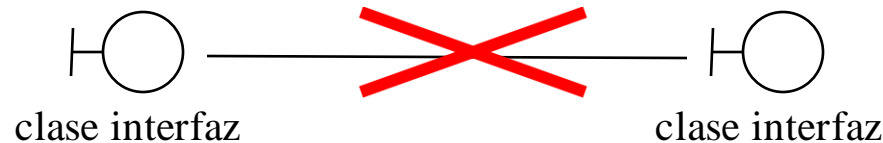
- **Distribuir el comportamiento entre las clases de análisis**
  - Incluir únicamente los objetos necesarios para realizar el caso de uso
  - Se identifican nuevos objetos participantes en el caso de uso
  - Se identifica comportamiento no considerado anteriormente
  - Se centra en el comportamiento de alto nivel
- **Recomendaciones**
  - Primera columna el actor que inicia el caso de uso
  - Segunda columna el objeto interfaz con el que interactúa el actor para iniciar el caso de uso
  - Tercera columna el objeto control que gestiona el resto del caso de uso
  - Los objetos de control son creados por objetos interfaz que inician el caso de uso
  - Los objetos interfaz son creados por objetos control
  - Los objetos entidad son accedidos por objetos control

# CONSEJOS PRÁCTICOS

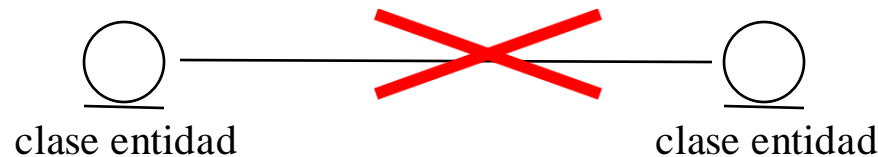
Definir interacciones

## Heurísticas

- **Actores** solamente pueden interaccionar con clases interfaz
- **Clases Interfaz** interaccionan, en principio, con clases control y actores



- **Clases Entidad** interaccionan, en principio, con clases control



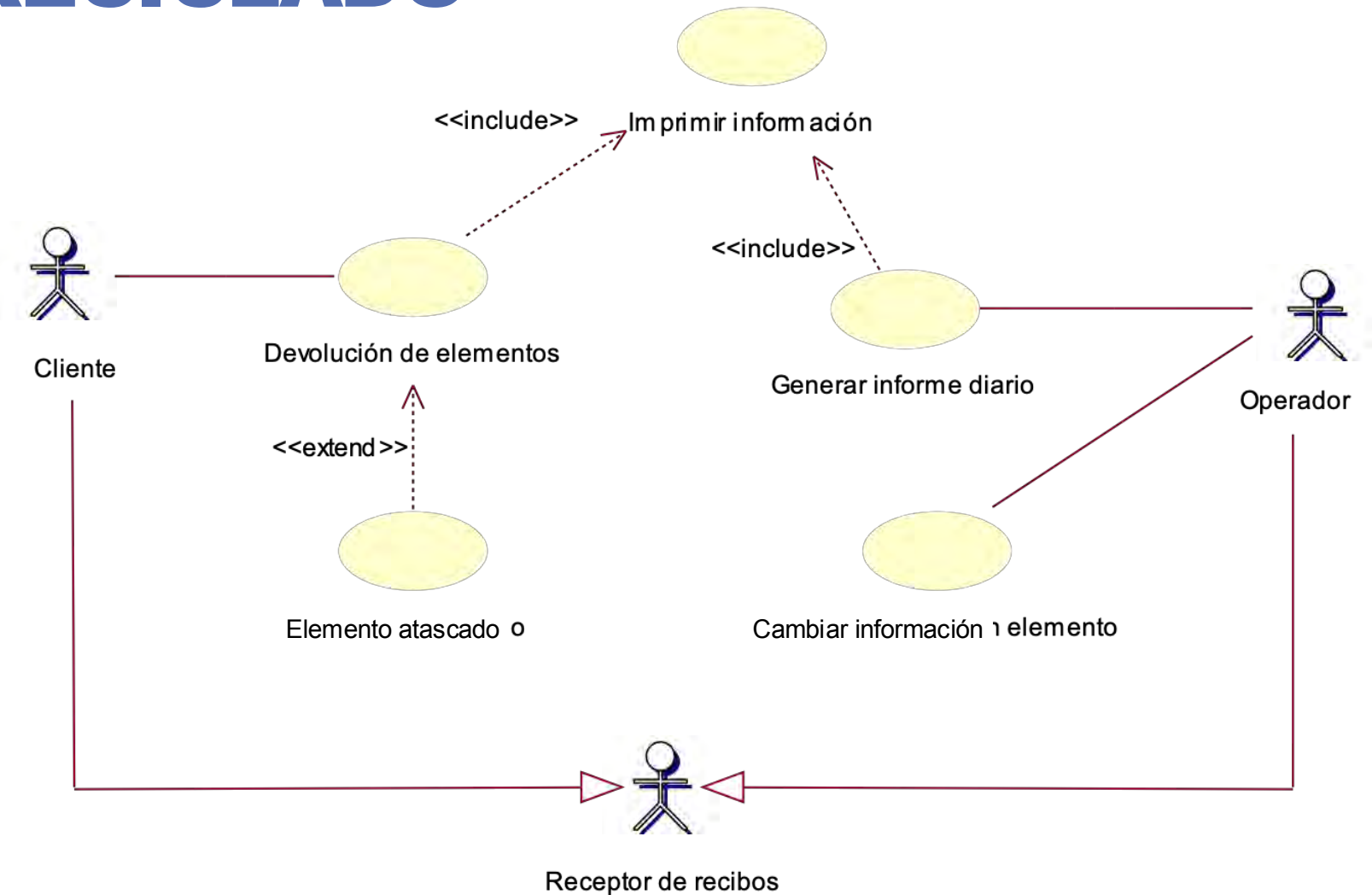
- **Clases Control** pueden interaccionar con clases interfaz, entidad y control

# CONSEJOS PRÁCTICOS

## Definir interacciones

- **Describir responsabilidades**
  - Una responsabilidad es cualquier “servicio” que puede ser solicitado a un objeto
  - Una responsabilidad puede implicar una o varias operaciones
  - Una responsabilidad se caracteriza por
    - Las acciones que el objeto puede realizar
    - El conocimiento que el objeto mantiene y proporciona a otros objetos
- **Las responsabilidades se derivan de los mensajes en los diagramas de interacción**
  - Pueden aparecer responsabilidades ligadas a requisitos no funcionales
- **Asignar responsabilidades a cada objeto en forma de conjunto de operaciones**
  - Si una operación aparece en más de un diagrama de secuencia, su comportamiento ha de ser el mismo

# EJEMPLO – MÁQUINA DE RECICLADO



# EJEMPLO – MÁQUINA DE RECICLADO

## Devolución de elementos. Escenario básico

- Inicio: El cliente desea devolver latas, botellas y envases
- El caso de uso comienza cuando el cliente presiona el “botón inicio” en el panel del cliente. Los sensores incorporados en el panel se activan
- El cliente puede devolver elementos (bote, botella, envase) a través del panel de cliente
- Los sensores informan al sistema que un objeto ha sido insertado, calibran el elemento depositado y devuelven el resultado al sistema
- El sistema utiliza el resultado medido para determinar el tipo de elemento devuelto
- El total diario de elementos depositados se incrementa, así como el total de elementos que el cliente ha depositado
- Cuando el cliente ha depositado todos los elementos a devolver solicita el recibo presionando el “botón recibo”
- El sistema genera un recibo con la información recogida por cada tipo de elemento depositado
- Se imprime el recibo y el caso de uso finaliza

# EJEMPLO – MÁQUINA DE RECICLADO

## Devolución de elementos. Objetos de dominio

- Inicio: El **cliente** desea devolver latas, botellas y envases
- El caso de uso comienza cuando el cliente presiona el “botón inicio” en el panel del cliente. Los sensores incorporados en el panel se activan
- El cliente puede devolver **elementos** (**bote**, **botella**, **envase**) a través del panel de cliente
- Los sensores informan al sistema que un objeto ha sido insertado, calibran el elemento depositado y devuelven el resultado al sistema
- El sistema utiliza el resultado medido para determinar el tipo de elemento devuelto
- El total diario de elementos depositados se incrementa, así como el total de elementos que el cliente ha depositado
- Cuando el cliente ha depositado todos los elementos a devolver solicita el recibo presionando el “botón recibo”
- El sistema genera un recibo con la información recogida por cada tipo de elemento depositado
- Se imprime el **recibo** y el caso de uso finaliza

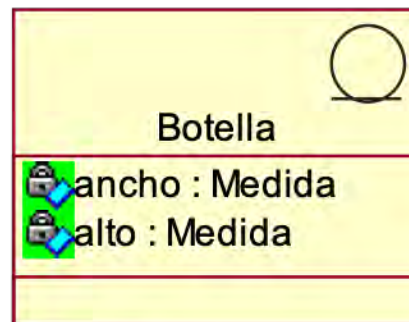
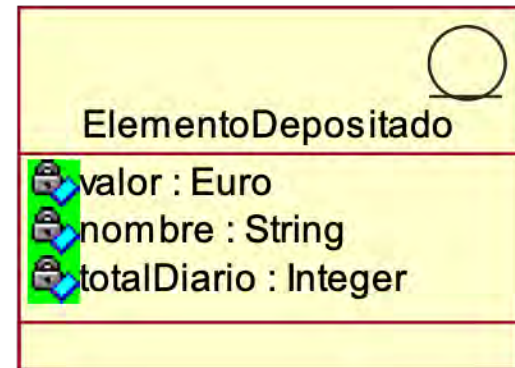
# EJEMPLO – MÁQUINA DE RECICLADO

## Devolución de elementos. Clases de entidad

- Inicio: El **cliente** desea devolver latas, botellas y envases
- El caso de uso comienza cuando el cliente presiona el “botón inicio” en el panel del cliente. Los sensores incorporados en el panel se activan
- El cliente puede devolver **elementos** (**bote**, **botella**, **envase**) a través del panel de cliente
- Los sensores informan al sistema que un objeto ha sido insertado, calibran el elemento depositado y devuelven el resultado al sistema
- El sistema utiliza el *resultado medido para determinar el tipo de elemento devuelto*
- El *total diario de elementos depositados se incrementa*, así como el *total de elementos que el cliente ha depositado*
- Cuando el cliente ha depositado todos los elementos a devolver solicita el recibo presionando el “botón recibo”
- El sistema compila la información que ha de imprimirse en el recibo. Por cada tipo de elemento depositado extrae *su valor de devolución y el número de elementos depositados* por el cliente actual
- Se imprime el **recibo** con el detalle y el total de los elementos devueltos y el caso de uso finaliza

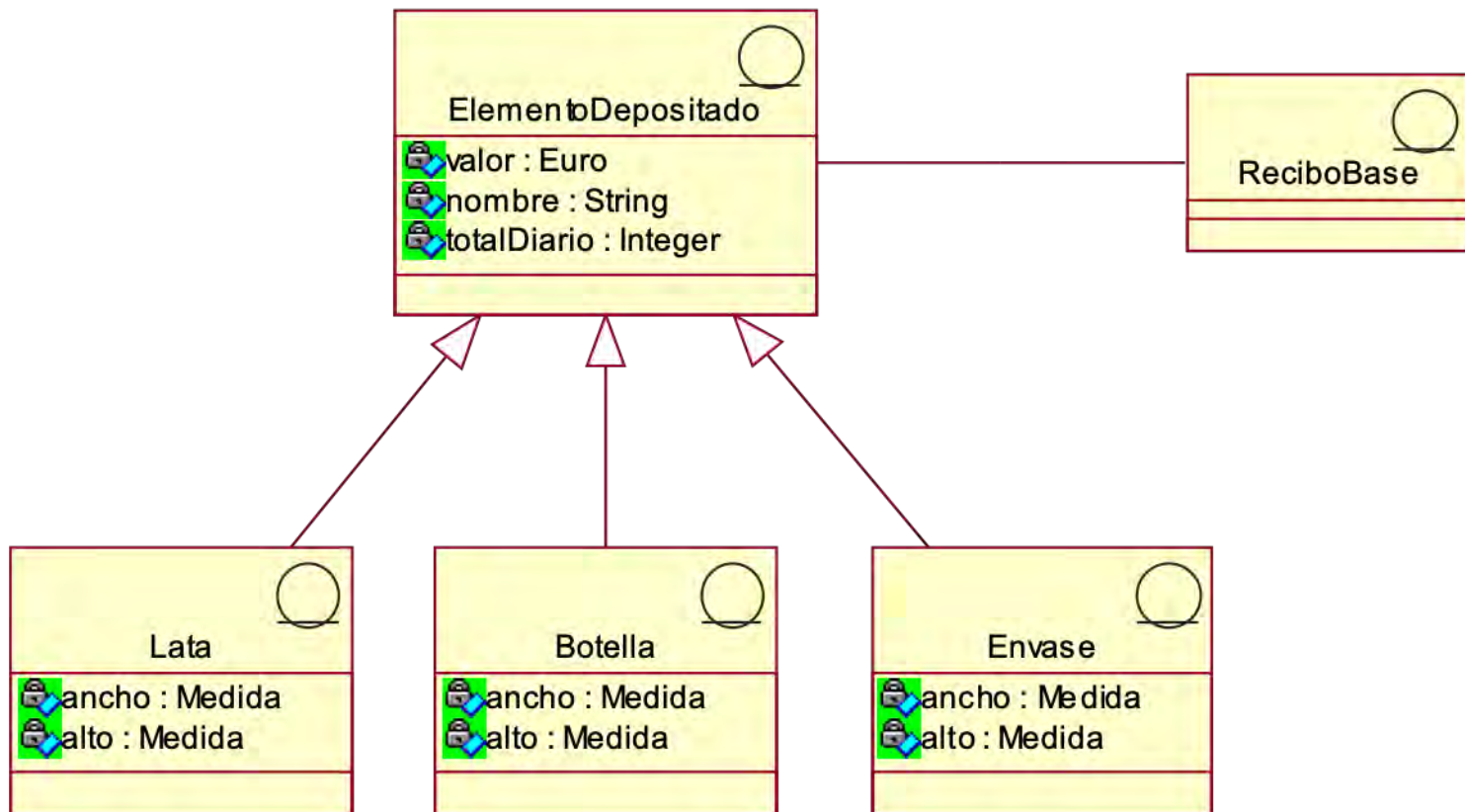
# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de entidad



# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de entidad



# EJEMPLO – MÁQUINA DE RECICLADO

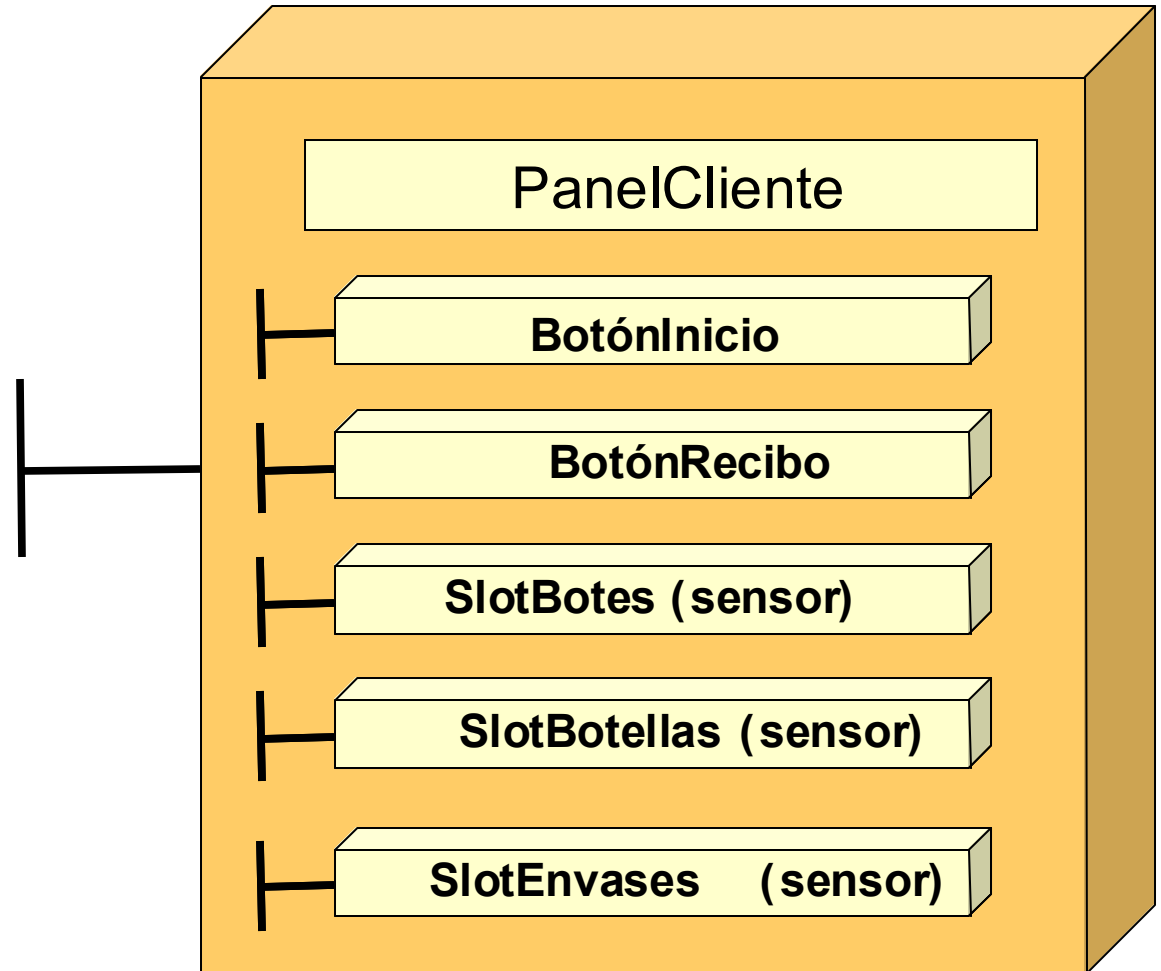
## Devolución de elementos. Clases de interfaz

- Inicio: El cliente desea devolver latas, botellas y envases
- El caso de uso comienza cuando el cliente presiona el “**botón inicio**” en el **panel del cliente**. Los **sensores** incorporados en el panel se activan
- El cliente puede devolver elementos (**bote, botella, envase**) a través del panel de cliente
- Los sensores informan al sistema que un objeto ha sido insertado, calibran el elemento depositado y devuelven el resultado al sistema
- El sistema utiliza el resultado medido para determinar el tipo de elemento devuelto
- El total diario de elementos depositados se incrementa, así como el total de elementos que el cliente ha depositado
- Cuando el cliente ha depositado todos los elementos a devolver solicita el recibo presionando el “**botón recibo**”
- El sistema genera un recibo con la información recogida por cada tipo de elemento depositado
- Se imprime el recibo y el caso de uso finaliza

# EJEMPLO – MÁQUINA DE RECICLADO

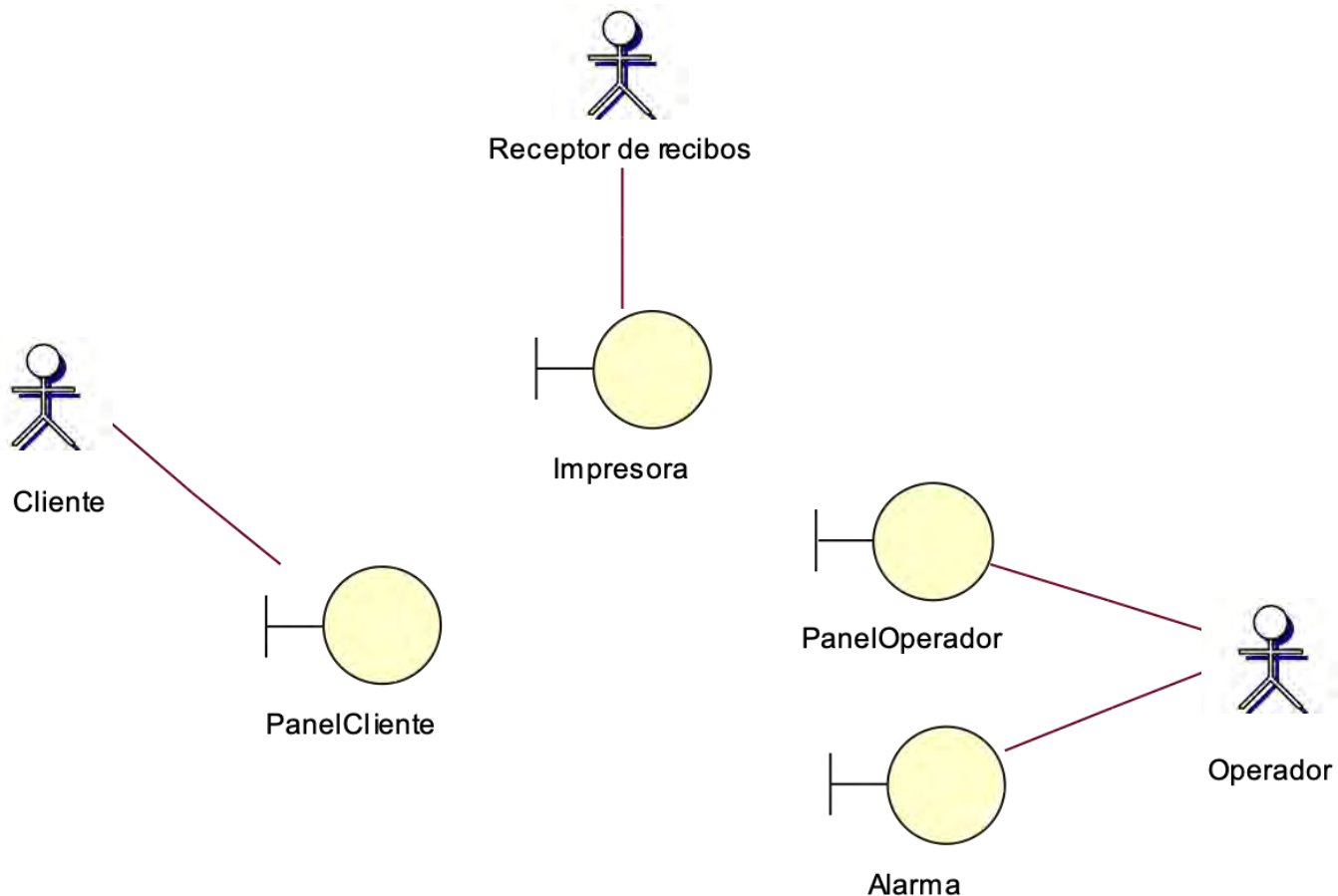
Devolución de elementos. Clases de interfaz

## Clase interfaz principal



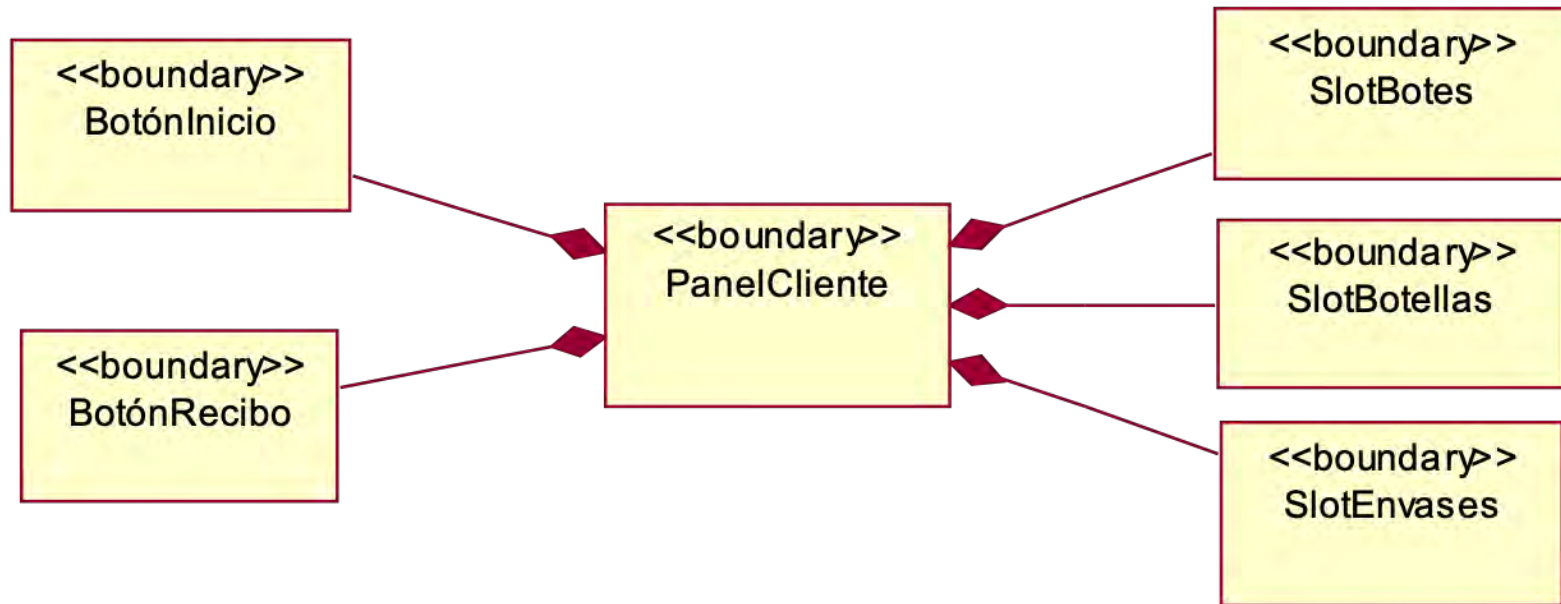
# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de interfaz



# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de interfaz



# EJEMPLO – MÁQUINA DE RECICLADO

## Devolución de elementos. Clases de control

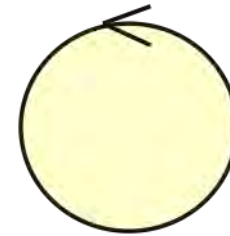
- Inicio: El cliente desea devolver latas, botellas y envases
- El caso de uso comienza cuando el cliente presiona el “botón inicio” en el panel del cliente. Los sensores incorporados en el panel se activan
- El cliente puede devolver elementos (bote, botella, envase) a través del panel de cliente
- Los sensores informan al sistema que un objeto ha sido insertado, *calibran el elemento depositado* y *devuelven el resultado al sistema*
- El sistema utiliza el *resultado medido para determinar* el tipo de elemento devuelto
- El *total diario de elementos depositados* se incrementa, así como el total de elementos que el cliente ha depositado
- Cuando el cliente ha depositado todos los elementos a devolver solicita el recibo presionando el “botón recibo”
- El *sistema compila la información que ha de imprimirse en el recibo*. Por cada tipo de elemento depositado extrae su valor de devolución y el número de elementos depositados por el cliente actual
- *Se imprime el recibo con el detalle y el total de los elementos devueltos* y el caso de uso finaliza

# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de control



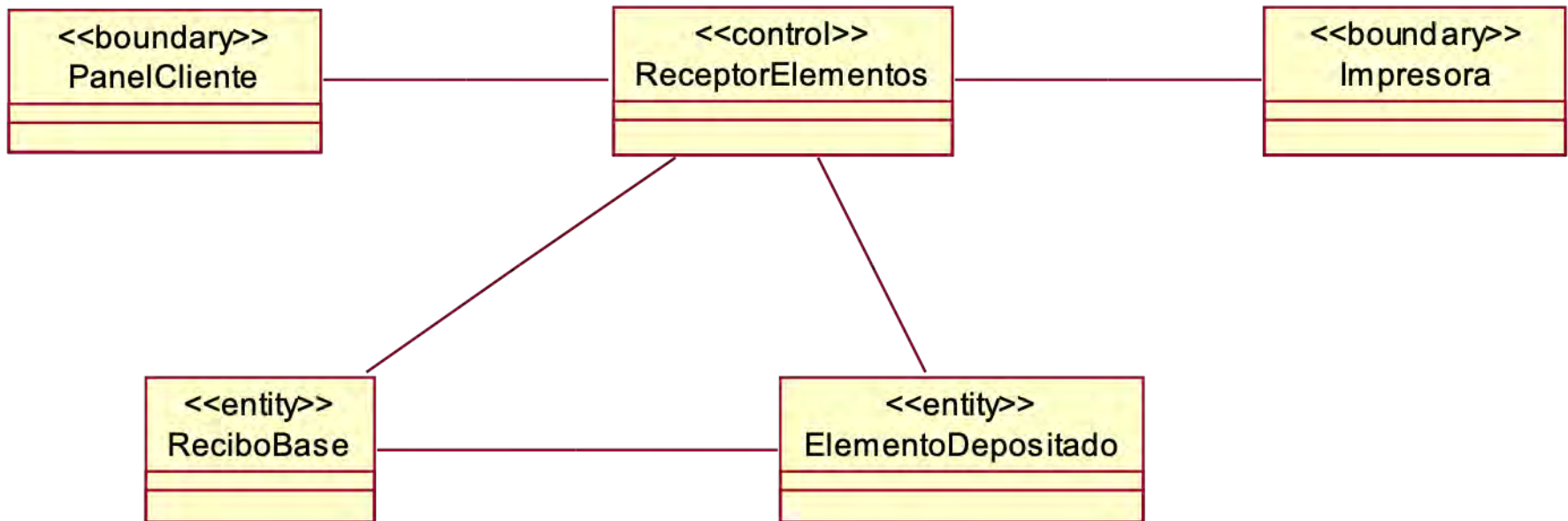
ReceptorElementos



GeneradorInformes

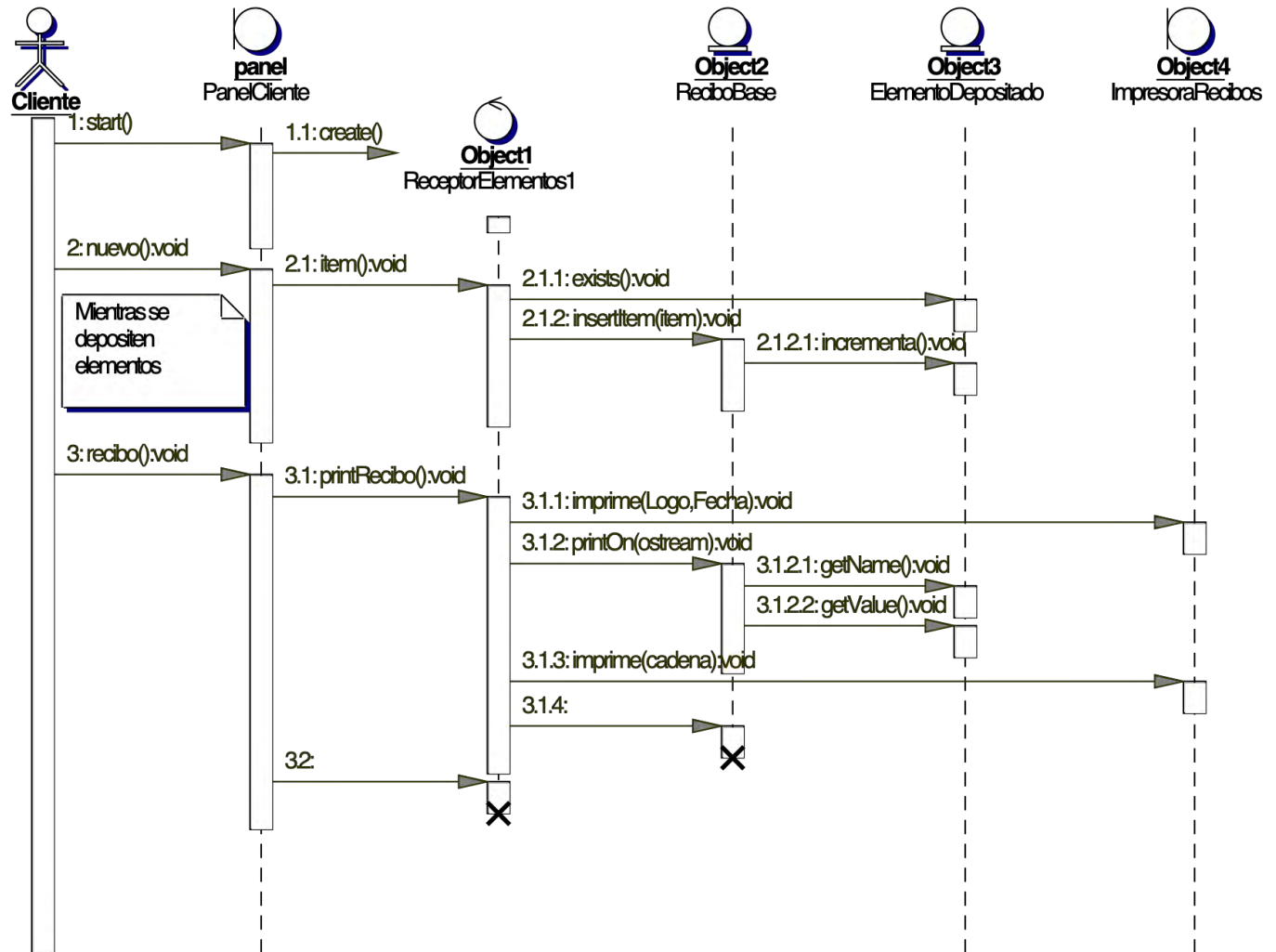
# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Clases de control



# EJEMPLO – MÁQUINA DE RECICLADO

Devolución de elementos. Diagrama de Secuencia



# REFERENCIAS

1. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Flujos de trabajo del Proceso Unificado," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2024-2025, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2025. [Online]. Disponible en. doi: 10.5281/zenodo.14841399.
2. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Análisis orientado a objetos," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2024-2025, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2025. [Online]. Disponible en. doi: 10.5281/zenodo.14841441.
3. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Análisis Orientado a Objetos – Píldora de vídeo," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2020-2021, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2021. [Online]. Disponible en: <https://bit.ly/3sbW2cn>. doi: 10.5281/zenodo.5786754.
4. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Análisis Orientado a Objetos en el Proceso Unificado – Consejos prácticos," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2020-2021, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2021. [Online]. Disponible en: <https://bit.ly/3DWc1xz>. doi: 10.5281/zenodo.5786835.
5. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Máquina de reciclado," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2020-2021, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca, 2021. [Online]. Disponible en: <https://bit.ly/3sb6ssB>. doi: 10.5281/zenodo.5786856.

# REFERENCIAS

6. B. Bruegge y A. H. Dutoit, *Object-oriented software engineering. Using UML, patterns, and Java*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
7. Real Academia Española, *Diccionario de la lengua española. Versión electrónica 23.8*, Madrid, España: Real Academia Española, 2024. [Online]. Disponible en: <http://www.rae.es>.
8. IEEE. *IEEE Software Engineering Standards Collection 1999 Edition. Volume 1: Customer and Terminology Standards*. USA: IEEE Computer Society Press, 1999.
9. I. Jacobson, G. Booch y J. Rumbaugh, *The Unified Software Development Process* (Object Technology Series). Reading, Massachusetts, USA: Addison Wesley, 1999.
10. D. E. Monarchi y G. I. Pühr, "A Research Typology for Object-Oriented Analysis and Design," *Communications of the ACM*, vol. 35, no. 9, pp. 35-47, 1992. doi: 10.1145/130994.130995.
11. F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, "Modelo de Dominio," Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2024-2025, F. J. García-Peñalvo, A. García-Holgado y A. Vázquez-Ingelmo, Eds., Salamanca, España: Grupo GRIAL, 2025. [Online]. Disponible en. doi: 10.5281/zenodo.14984086.

# INTRODUCCIÓN AL ANÁLISIS ORIENTADO A OBJETOS

## INGENIERÍA DE SOFTWARE I

2º DE GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2024/2025

Francisco José García-Peñalvo / [fgarcia@usal.es](mailto:fgarcia@usal.es)

Alicia García-Holgado / [aliciagh@usal.es](mailto:aliciagh@usal.es)

Andrea Vázquez-Ingelmo / [andreavazquez@usal.es](mailto:andreavazquez@usal.es)

Departamento de Informática y Automática  
Universidad de Salamanca

