

Modelo Orientado a Objetos

1. Identificación de Objetos

- Narrativa de Procesamiento ->Análisis Gramatical
- Subrayado de sustantivos (Lista de Objetos Potenciales)
- Tipificación y filtrado (Obtener Objetos del Sistema)
- Determinación de Atributos (Adjetivos)
- Determinación de Métodos (Verbos)

2. Casos de Uso(Escenarios ó Escenas de uso)

3. Modelado CRC (Tarjetas Índice)

4. Definición de Jerarquía de Clases

Modelo Orientado a Objetos

- 5. Validación
- 6. Modelo Objeto-Relación
- 7. Modelo Objeto-Comportamiento

1. Identificación de Objetos

► Determinación de Atributos (Adjetivos)

Los atributos **describen un objeto** que ha sido seleccionado para ser incluido en el modelo de análisis.

El contexto que se está analizando define, en gran medida, los atributos del objeto.

Para desarrollar un conjunto significativo de atributos para un objeto, el analista debe seleccionar de la narrativa de procesamiento aquellos **elementos que pertenecen al objeto**.

3

► Debemos responder a la siguiente pregunta:

¿Qué elementos (compuestos y /o simples) definen completamente al objeto en el contexto del problema actual?

1. Identificación de Objetos

► Determinación de Métodos (Verbos)

Los Métodos **definen el comportamiento de un objeto y cambian**, de alguna manera, los atributos de dicho objeto.

Un Método **cambia valores de uno o más atributos del objeto**.

Deben tener “conocimiento” de la naturaleza de esos atributos y deben estar implementadas de manera tal que permita manipular las estructuras de datos que han sido derivadas de dichos atributos.

4

Los Métodos se pueden clasificar en tres grandes categorías:

- Métodos que **manipulan datos**.
- Métodos que **realizan cálculos**.
- Métodos que **monitorizan un objeto** frente a algún suceso de control

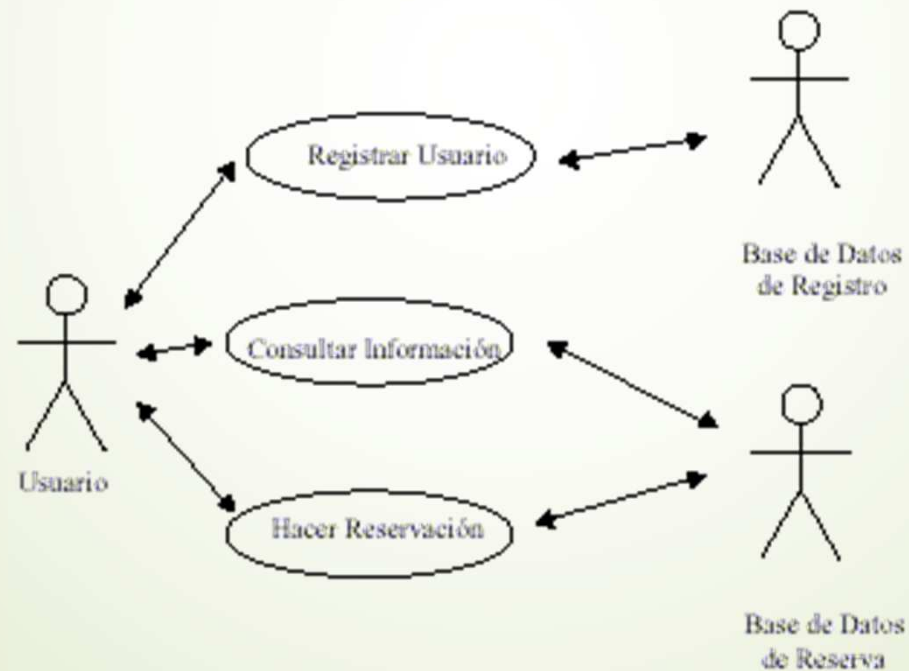
2. Casos de Uso

- Se desarrollan en base a los requisitos del sistema:
 - Son escenas que indican una parte del sistema a construir.
 - Aportan una descripción de cómo el sistema será usado.
 - **En General**, son simples narraciones escritas que describen la interacción entre el actor y el sistema.
- Primero Identificar actores:
 - Cualquier entidad que se comunique con el sistema y que sea externo a él.
 - Representar papeles ó roles ejecutados por personas o dispositivos cuando el sistema esta en operación.
 - Actor no es lo mismo que usuario.

2. Casos de Uso

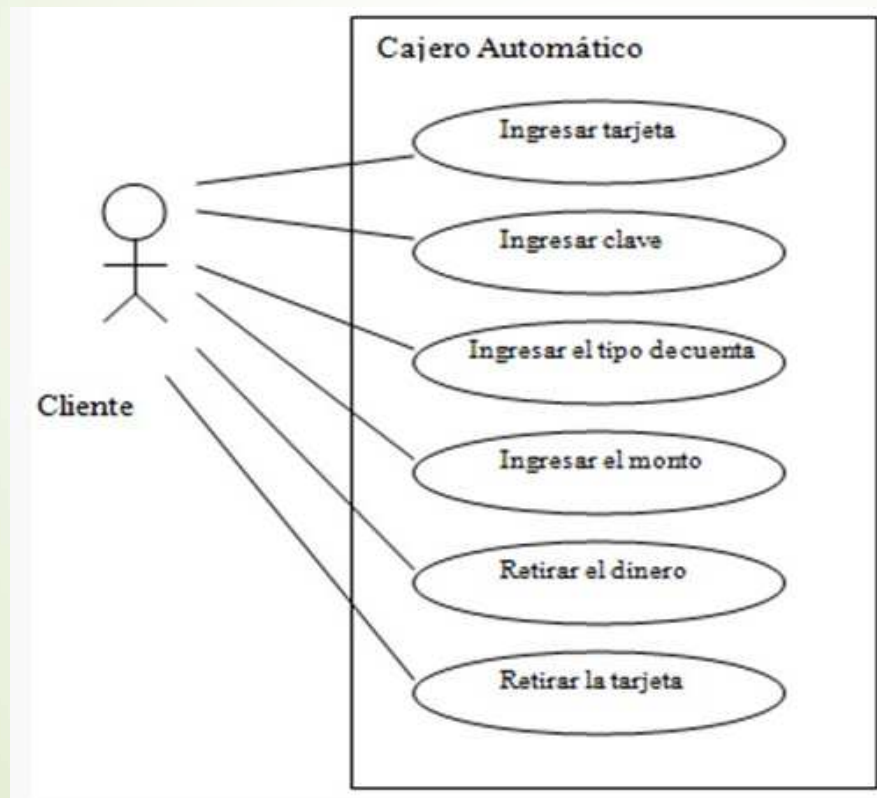
■ Actores:

- **Actores primarios** (interactúan en forma directa)
- **Actores secundarios** (interactúan para posibilitar la interacción del sistema con los actores primarios)



2. Casos de Uso

- Cada Caso de uso aporta un escenario no ambiguo de interacción entre el actor y el software.



Notación UML

2. Casos de Uso

■ Según **Jacobson** (92) el caso de uso debe responder a las siguientes cuestiones:

- ¿Cuáles son las tareas o funciones principales a realizar por el actor?.
- ¿Qué información del sistema adquirirá, producirá o cambiará el actor?.
- ¿Tendrá el actor que informar al sistema acerca de cambios en el entorno exterior?.
- ¿Qué información desea el actor sobre el sistema?.
- ¿Desea el actor ser informado sobre cambios inesperados?.

2. Casos de Uso

► Ejemplo **Hogar Seguro**:

► Podemos definir tres actores:

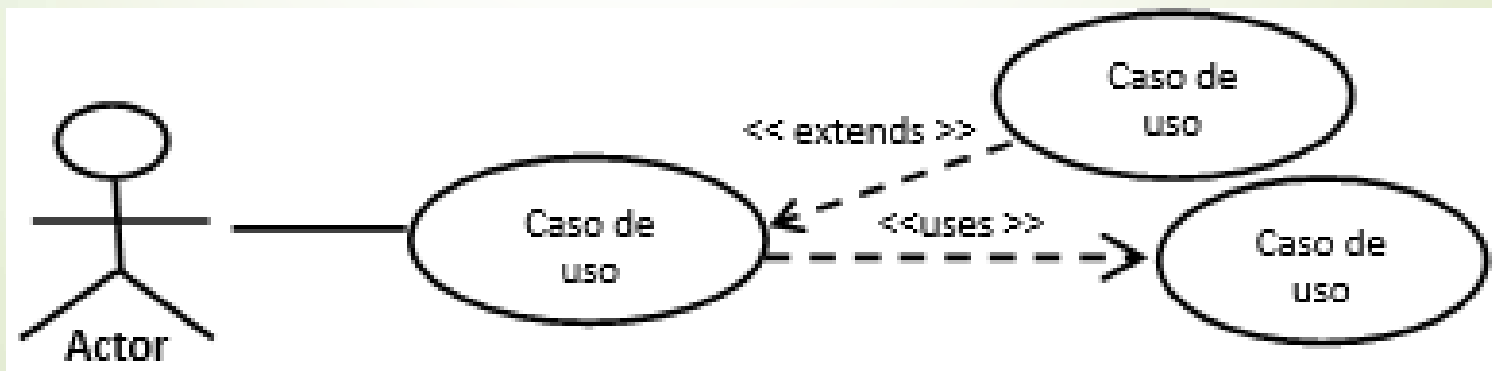
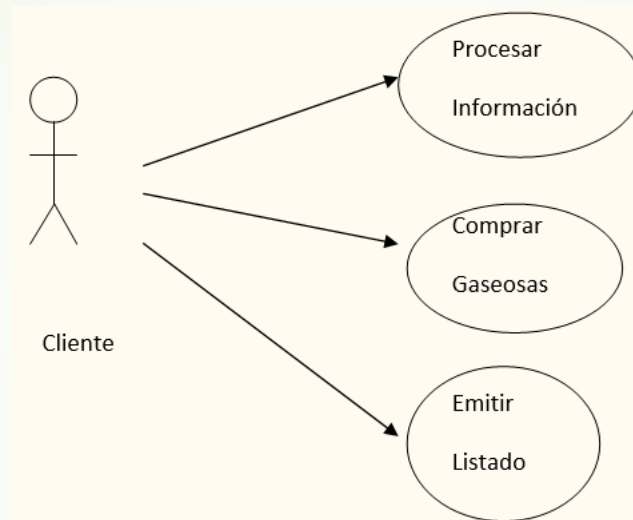
- El **propietario** (usuario).
- Los **sensores** (dispositivos adjuntos al sistema).
- El **subsistema de monitorización y respuesta** (la estación central que monitoriza a Hogar Seguro).

► El actor **propietario** interactúa con el producto en un número de formas diferentes:

- Introduce una contraseña para permitir el resto de las interacciones.
- Interroga acerca del estado de una zona de seguridad.
- Presiona el botón de pánico en una emergencia.
- Activa/Desactiva el sistema de seguridad.

CASOS DE USO

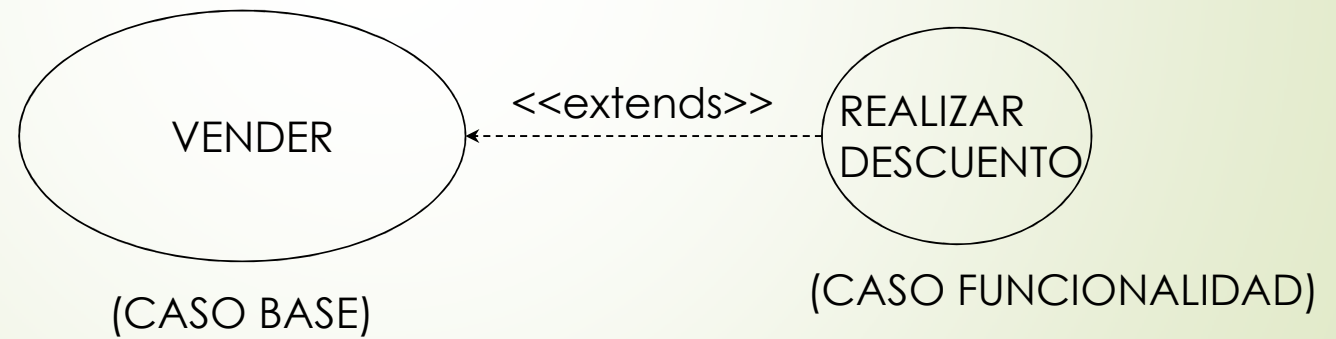
► Diagramas de Caso de uso



CASOS DE USO

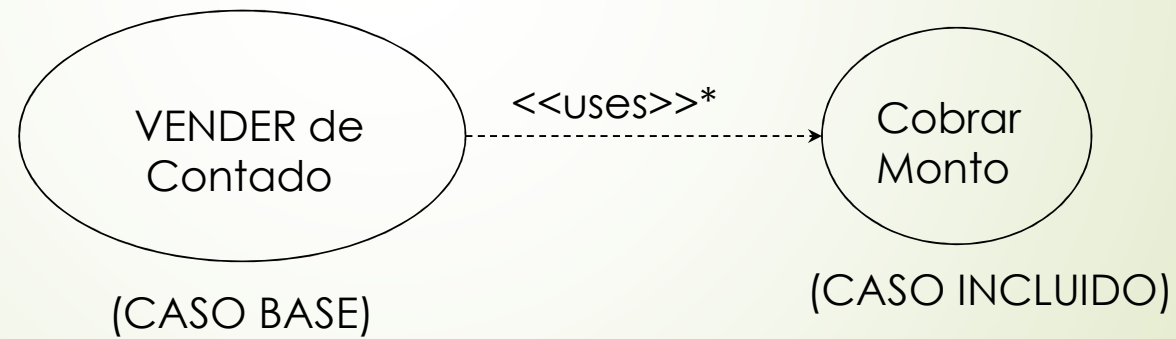
- **Relación de EXTENSION:**

- Extiende funcionalidad.
- No es esencial



CASOS DE USO

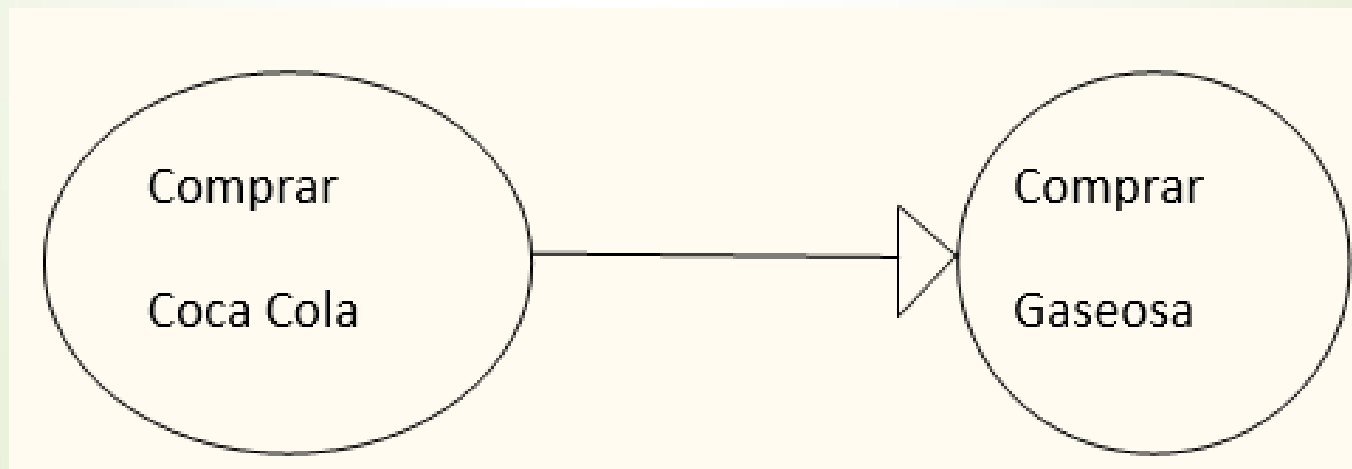
- **Relación de INCLUSION:**
 - Es esencial, el 1º no podría existir sin el caso de uso incluido.



*<<include>>

CASOS DE USO

- **Relación de GENERALIZACION:**
 - Especifica una relación de generalización/especialización (HERENCIA).



3. Modelado CRC

- Un modelo CRC (Clases – Responsabilidades - Colaboraciones) es realmente una colección de tarjetas índice estándar que representan clases.
- Las tarjetas están divididas en tres secciones:
 - Una cabecera con el nombre de la clase
 - En el cuerpo, a la izquierda, las responsabilidades de la clase.
 - En el cuerpo, a la derecha, los colaboradores.

Nombre de la clase:	
Tipo de la clase: (dispositivo, propiedad, rol,...)	
Características de la clase: (tangible, atómica, concurrente,...)	
Responsabilidades	Colaboradores

3. Modelado CRC

► Tipificación de Clases:

- Entidades Externas, Cosas, Ocurrencias o eventos
- Roles, Unidades organizativas, etc

Se agregan:

- **Clases dispositivo:** Modelan entidades externas tales como sensores, motores y teclados.
- **Clases propiedad:** Representan alguna propiedad importante del entorno del problema (ej.: establecimiento de créditos en el contexto de una aplicación de préstamos hipotecarios).
- **Clases interacción:** Modelan interacciones que ocurren entre otros objetos (ej.: una adquisición o una licencia).

3. Modelado CRC

- **Características de las Clases:**
 - **Tangibilidad:**
 - Información **Tangible** o **Abstracta**.
 - **Inclusividad:**
 - La clase es **Atómica** o **Agregada**.
 - **Secuencialidad:**
 - La clase es **Concurrente** ó **Secuencial**.
 - **Persistencia:**
 - Puede ser **Transitoria**, **Temporal** ó **Permanente**.
 - **Integridad:**
 - Puede ser **Corrompible** o **Segura**.

3. Modelado CRC

► Responsabilidades:

► Atributos:

- Representar información estable que debe manejar los objetos para cumplir sus objetivos.

► Operaciones:

- Representan los comportamientos de la clase.

3. Modelado CRC

► 5 Pautas para especificar responsabilidades:

1. La inteligencia del sistema debe distribuirse de manera igualitaria.

Se llama inteligencia de un sistema a lo que la clase sabe o es capaz de hacer. (Clases tontas e inteligentes).

La distribución igualitaria de inteligencia **incrementa la cohesión** y los **efectos laterales** tienden a amortiguarse.

Para determinar la inteligencia de cada clase, se analiza la lista de responsabilidades.

3. Modelado CRC

► 5 Pautas para especificar responsabilidades:

2. Cada responsabilidad debe establecerse lo más general posible.
3. La información y el comportamiento asociado a ella, debe encontrarse dentro de la misma clase.

Esto implementa el **principio de encapsulamiento**.

3. Modelado CRC

► **5 Pautas para especificar responsabilidades:**

4. La información sobre un elemento debe estar localizada dentro de una clase, no distribuida a través de varias clases.

Si la información está distribuida, el software se torna más difícil de mantener y probar

3. Modelado CRC

► 5 Pautas para especificar responsabilidades:

5. Compartir responsabilidades entre clases relacionadas cuando sea apropiado.

Existen muchos casos en los cuales una gran variedad de objetos exhibe el mismo comportamiento, al mismo tiempo (ej.: los objetos jugador, cuerpo_del_jugador, brazos_del_jugador y cabeza_del_jugador en un juego de vídeo tienen sus propios atributos pero comparten las responsabilidades de **actualizar** y **visualizar** cuando se acciona la palanca de mando (Joystick)).

3. Modelado CRC

- **Colaboradores:**

- Las clases cumplen con sus responsabilidades en una de dos maneras:
 - Una clase puede usar sus propias operaciones para manipular sus propios atributos, cumpliendo con una responsabilidad particular.
 - Una clase puede colaborar con otras clases.

3. Modelado CRC

► Colaboradores:

- Las colaboraciones representan solicitudes de un cliente a un servidor en el cumplimiento de una responsabilidad del cliente. Una colaboración es la **realización de un contrato** entre el cliente y el servidor.
- Una colaboración simple fluye en una dirección, representando una solicitud del cliente al servidor.
- Desde el punto de vista del cliente, cada una de sus colaboraciones está asociada con una responsabilidad particular implementado por el servidor.