

Facultad de Informática - UNLP  
Orientación a Objetos I - 2023

# Introducción al Análisis y Diseño Orientado a Objetos

Prof. Diego Torres

- Martes: Presentación del tema
- Jueves: Consultas teóricas - puedo llevar ejercicio para poner en práctica lo visto.

- Análisis

El **análisis** pone énfasis en una **investigación del problema** y los **requisitos**, en lugar de ponerlo en la solución.

- Diseño

El **diseño** pone énfasis en una **solución conceptual** que **satisface los requisitos**, en lugar de ponerlo en la implementación.

## *Ejercicio - Libros*

- En un sitio de comercio electrónico se venden libros, que pueden ser digitales o impresos. El comercio mantiene un control sobre su inventario.
- Los libros tienen:
  - Título
  - Autor
  - Edición
- El precio final de un libro digital es el precio de costo más una comisión del 2%, el precio final de un libro impreso tiene una comisión del 1% pero se agregan \$20 de gastos de envío.

## Ejercicio - Libros

- Administración de clientes
  - Pueden ser **normales** o **VIP**.
  - Cuentan con un crédito para comprar libros.
  - Pueden hacer recargas de su crédito.
    - **A los clientes VIP**, con cada recarga mayor a \$50 se les bonifica el 5% del valor de la misma.
  - Todos los clientes deben saber qué libros compraron
- Un cliente puede pasar de ser Normal a VIP si supera los \$200 de gasto en el último mes.
- Para poder realizar compras, los usuarios deben registrarse como clientes del sitio con sus datos personales (nombre completo, dirección, teléfono, dirección de correo electrónico).
- El cliente puede comprar libros y también cancelar su compra.

# Casos de Uso

## Visto con la profesora Gabriela Perez

# Actores y casos de uso

Los casos de uso se definen para satisfacer los objetivos de usuario o actores principales.

- Identificar los actores principales (y actores secundarios)
- Identificar los objetivos de usuario de cada actor
- Definir los casos de uso

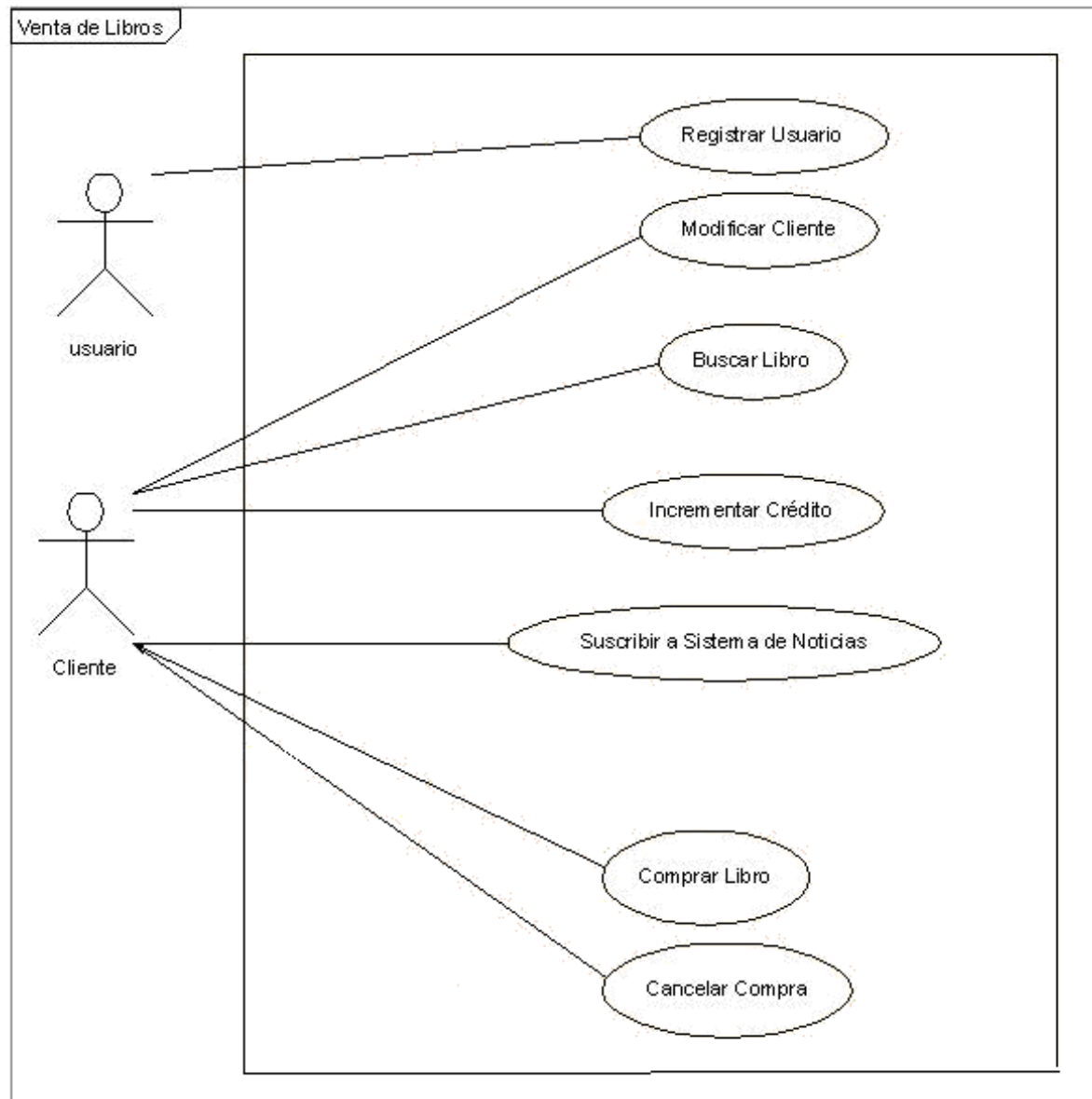
El actor principal y los objetivos de usuario dependen  
del límite del Sistema

# Diagramas de casos de uso

- Los casos de uso se definen para satisfacer los objetivos de usuarios o actores principales.
- El diagrama de casos de Uso proporciona información visual concisa del sistema, los actores externos y cómo lo utilizan.
- Los casos de uso y los actores deben tener “buenos” nombres.
  - Casos de uso: Verbo sustantivo
  - Actor: rol- subsistema- dispositivo



# Diagramas de casos de uso



# Casos de Uso- Tipos de formalidad

Existen tres tipos o grados de formalidad:

- Breve

Es un resumen conciso que no ocupa más de un párrafo. Se describe el escenario principal con éxito (curso normal).

- Informal

La descripción puede abarcar varios párrafos, pero no demasiados, especificando varios escenarios. Se caracteriza por un estilo informal de escritura.

- Completo

Es el formato más elaborado, ya que se describen con detalle todos los pasos y variaciones (curso normal y alternativos). Cuenta con otras secciones como pre y post condiciones, curso de error, etc.

# Ejemplo- Formato Completo

Título: Comprar un libro (Agregarlo al carrito)

Actor: Cliente

Prioridad: Alta

Descripción: Un cliente, una vez que ha ingresado al sistema, selecciona un libro para agregar y realiza la acción de agregar al carrito. Por cada libro, el Sistema agrega el libro al carrito de compras y guarda ese stock. El sistema calcula el precio del libro de acuerdo al tipo del mismo y a la categoría del cliente.

Precondiciones:

- El cliente es un cliente del Sistema
- El cliente posee un carrito asociado.

Postcondiciones:

- Se agregó el libro al carrito.
- Se decrementó el stock
- ... ..

# En Objetos 1 los encuentran así

## Ejercicio 15. Alquiler de propiedades

**Nota:** este ejercicio es del estilo de los que encontrarán en la evaluación parcial

Necesitamos que usted implemente OOBnB, un sistema para publicar propiedades en alquiler, y para alquilarlas. Identifique objetos y responsabilidades. Diseñe e implemente. No olvide los tests de unidad. El sistema ofrece la siguiente funcionalidad:

**Registrar usuarios:** Se provee nombre, dirección, dni. El sistema da de alta el usuario. El sistema retorna el Usuario. El usuario no tiene propiedades en alquiler. El usuario no tiene ninguna reserva de propiedad. El usuario no ha alquilado nunca una propiedad.

**Registrar una propiedad en alquiler:** Se provee nombre, descripción, precio por noche, y dirección. Se provee el usuario propietario. El sistema da de alta la propiedad y la retorna. La propiedad no tiene ninguna fecha ocupada.

**Buscar propiedades disponibles en un período:** Se indica el período (fecha de inicio y fecha de fin). Retorna todas las propiedades que se encuentran disponibles desde la fecha de inicio (inclusive) hasta el día de fin (inclusive).

**Hacer una reserva:** Se indica la propiedad, el período y el usuario para quien se hace la reserva (el inquilino). Si la propiedad está libre, se genera la reserva (que queda registrada en el sistema). La propiedad pasa a estar ocupada en esas fechas. Si la propiedad no está libre no hace nada y retorna null. Ver notas al final de este ejercicio sobre cómo podría resolver este punto.

**Calcular el precio de una reserva:** dada una reserva, obtener el precio a partir del precio por noche de la propiedad y la cantidad de noches de la reserva.

**Eliminar reserva:** Dada una reserva, si la fecha de inicio de la reserva es posterior a la fecha actual se elimina la reserva. La propiedad pasa a estar disponible en esas fechas.

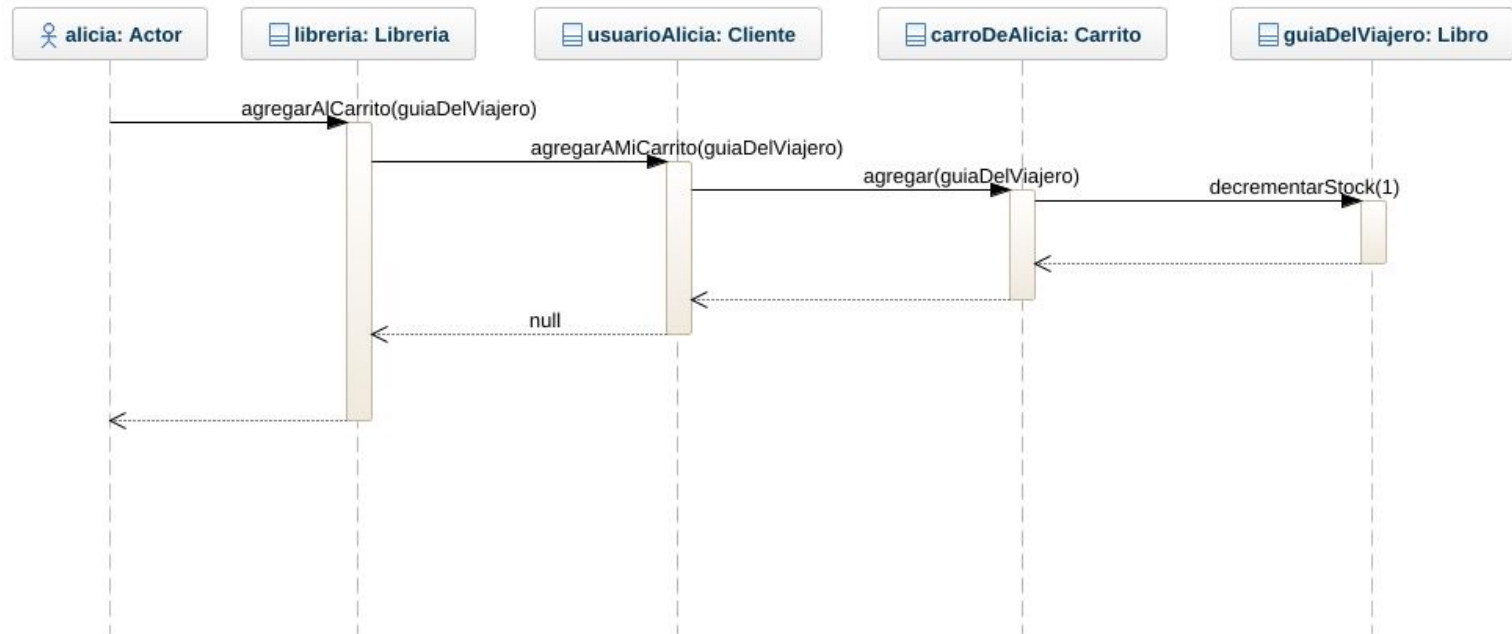
**Obtener las reservas de un usuario:** dado un usuario, obtener todas las reservas que ha efectuado (pasadas o futuras).

# Representación de los Diagramas de Secuencia del Sistema (DSS)

Visto con los profesores Gabriela  
Perez y Alejandro Fernandez

# Ejemplo de un DSS: agregar producto

Los DSS se derivan de los casos de uso

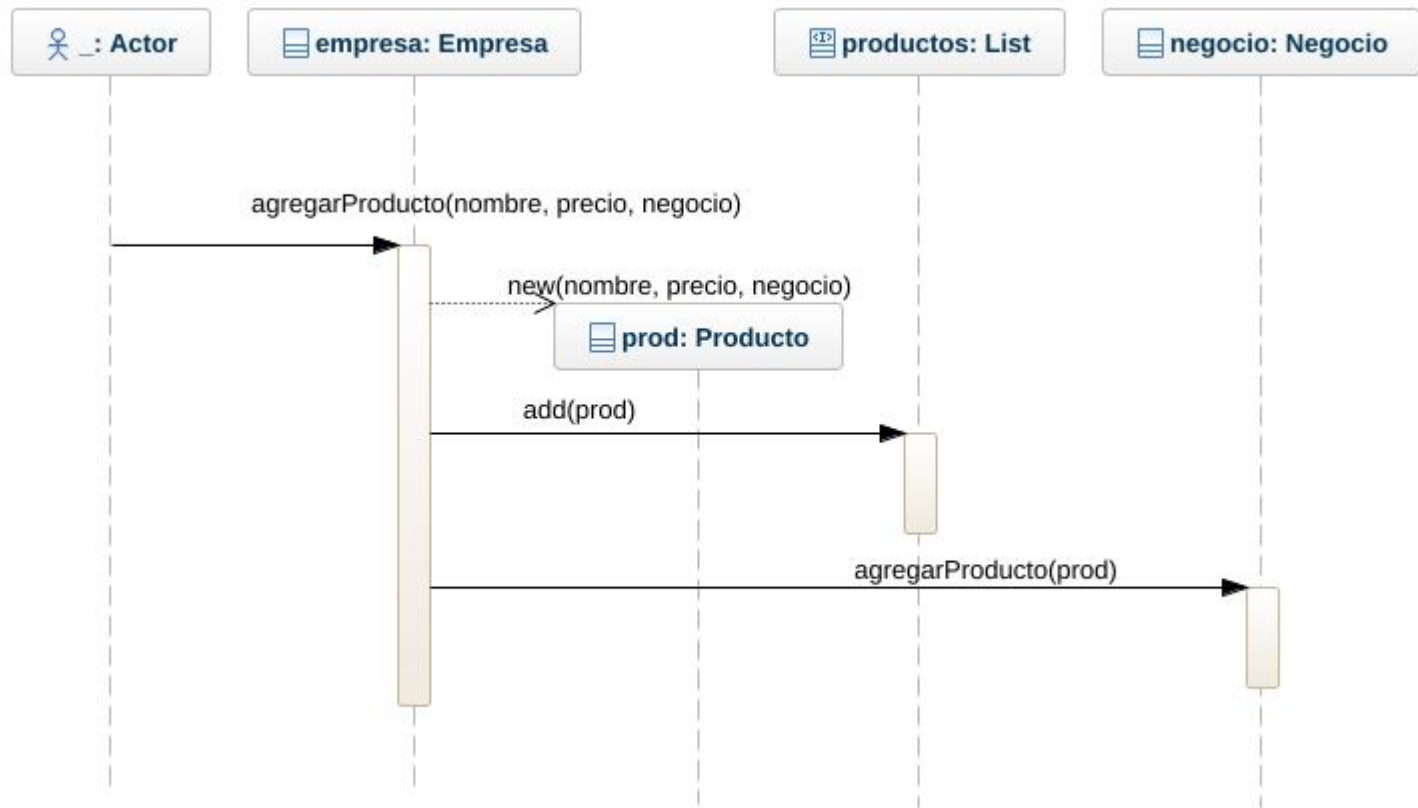


El actor genera el evento del sistema agregar al carrito

# Ejemplo de un DSS: agregar producto

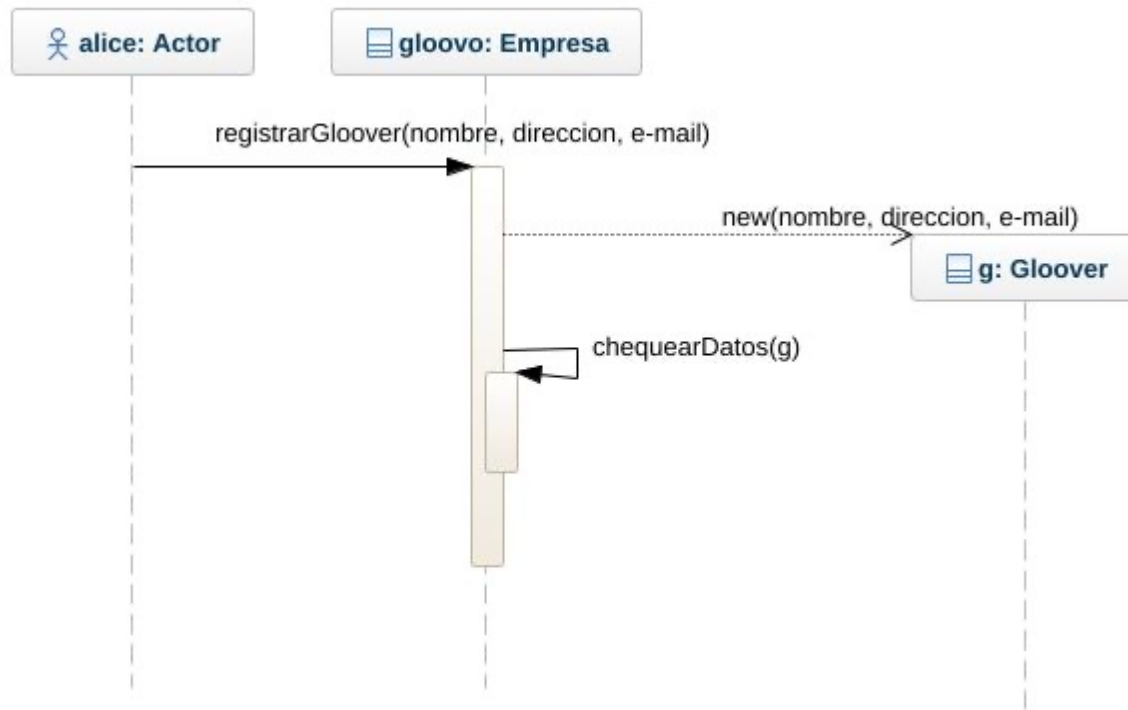
Los DSS se derivan de los casos de uso

Otra forma de multi objeto



El actor genera el evento del sistema agregarProducto

# Ejemplo de un DSS: registrarGloover





# Modelo del Dominio

# Identificación de clases conceptuales

La tarea central es identificar las clases conceptuales relacionadas con el escenario que se está diseñando.

Es mejor especificar en exceso un modelo del dominio con muchas clases conceptuales de grano fino que especificar por defecto

## Consejos:

- Usar nombres del **dominio del problema**, no de la solución.
- Omitir detalles irrelevantes
- No inventar nuevos conceptos (evitar **sinónimos**).
- Descubrir conceptos del **mundo real**

## Estrategias:

- Identificación de frases nominales.
- Utilización de una lista de categorías de clases conceptuales.

# Identificación de clases conceptuales- Frases nominales

Encontrar conceptos (y sus atributos) mediante la identificación de los **sustantivos** en la descripción textual del dominio del problema.

**Ejemplo:**

- 1- El **Cliente** selecciona un **libro** para agregar al **carrito**.
- 2- El sistema agrega el **libro** al **carrito**.
- 4- El carrito agrega el **libro** y se presenta la **descripción**,  
**precio** y **suma** parcial.

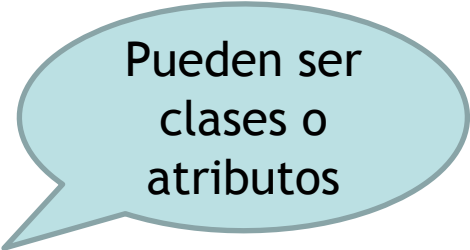
# Identificación de clases conceptuales- Lista de categorías

<b>Categoría de Clase Conceptual</b>	<b>Ejemplos</b>
<b>Objeto físico o tangible</b>	Libro impreso
<b>Especificación de una cosa</b>	Especificación del producto, descripción
<b>Lugar</b>	--
<b>Transacción</b>	Compra, pago, cancelación
<b>Roles de la gente</b>	cliente
<b>Contenedor de cosas</b>	Catálogo de libros, carrito
<b>Cosas en un contenedor</b>	Libro,
<b>Otros sistemas</b>	--
<b>Hechos</b>	cancelación, venta, pago
<b>Reglas y políticas</b>	Política de cancelación
<b>Registros financieros/laborales</b>	Factura/ Recibo de compra
<b>Manuales, documentos</b>	Reglas de cancelación, cambios de categoría de cliente

# Construyendo el Modelo del Dominio

Pasos a seguir:

- 1- Listar los conceptos candidatos
- 2- Graficarlos en un Modelo del Dominio.
- 3- Agregar atributos a los conceptos.
- 4- Agregar asociaciones entre conceptos.



Pueden ser  
clases o  
atributos

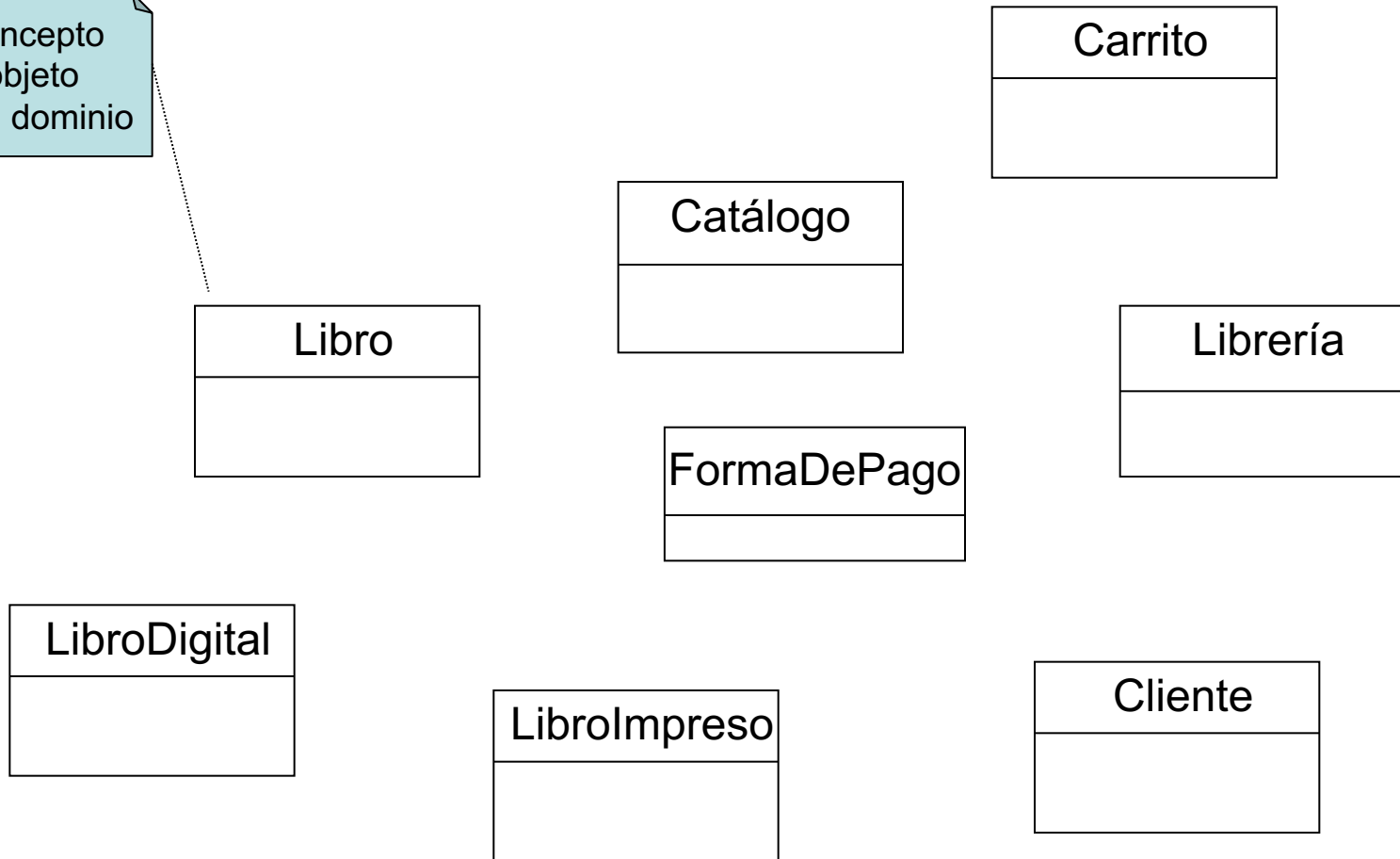
## (1) Conceptos candidatos (Clases y atributos candidatos)

- Librería
- Libro
- Libro digital
- Género
- Libro impreso
- Catálogo de libros
- Carrito
- Precio
- Detalle/LíneaDeVenta
- EspecificaciónDeProducto
- Pago
- Cliente
- ...

## (2) Modelo del Dominio- Visualización de conceptos

Un **Modelo del Dominio** es una representación visual de las clases conceptuales del mundo real en un dominio de interés.

Concepto  
u objeto  
del dominio



### (3) Agregar atributos

Se identifican los atributos que son necesarios para satisfacer los requerimientos de información de los casos de uso en desarrollo.

Los atributos en un modelo deberían ser, preferiblemente, atributos simples o tipos de datos primitivos.

Los tipos de datos de los atributos más comunes incluyen:

- Boolean
- Número
- String (texto)
- Temporales (Hora / Fecha)

Recuerde relacionar las clases conceptuales con asociaciones, no con atributos



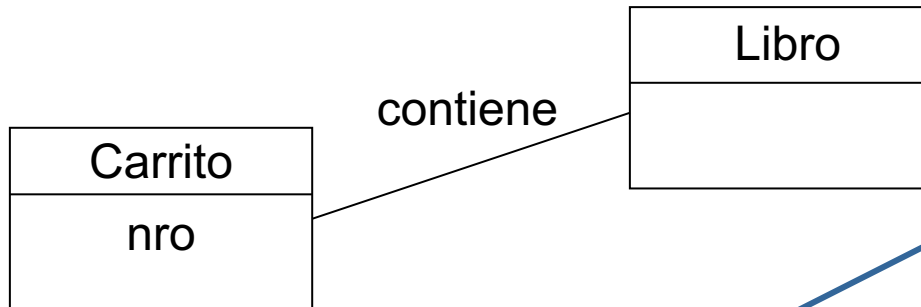
Represente lo que inicialmente podría considerarse un tipo de dato primitivo como una clase conceptual, si:

- Está compuesto de secciones separadas
- Tiene operaciones asociadas
- Tiene otros atributos
- Es una cantidad con una unidad
- Es una abstracción de uno o más tipos con esas cualidades

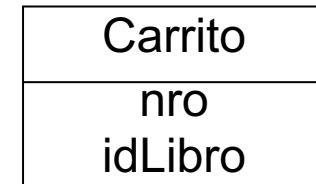
# Atributo como claves ajenas

¿Debería usar un atributo para relacionar clases conceptuales?

Mejor



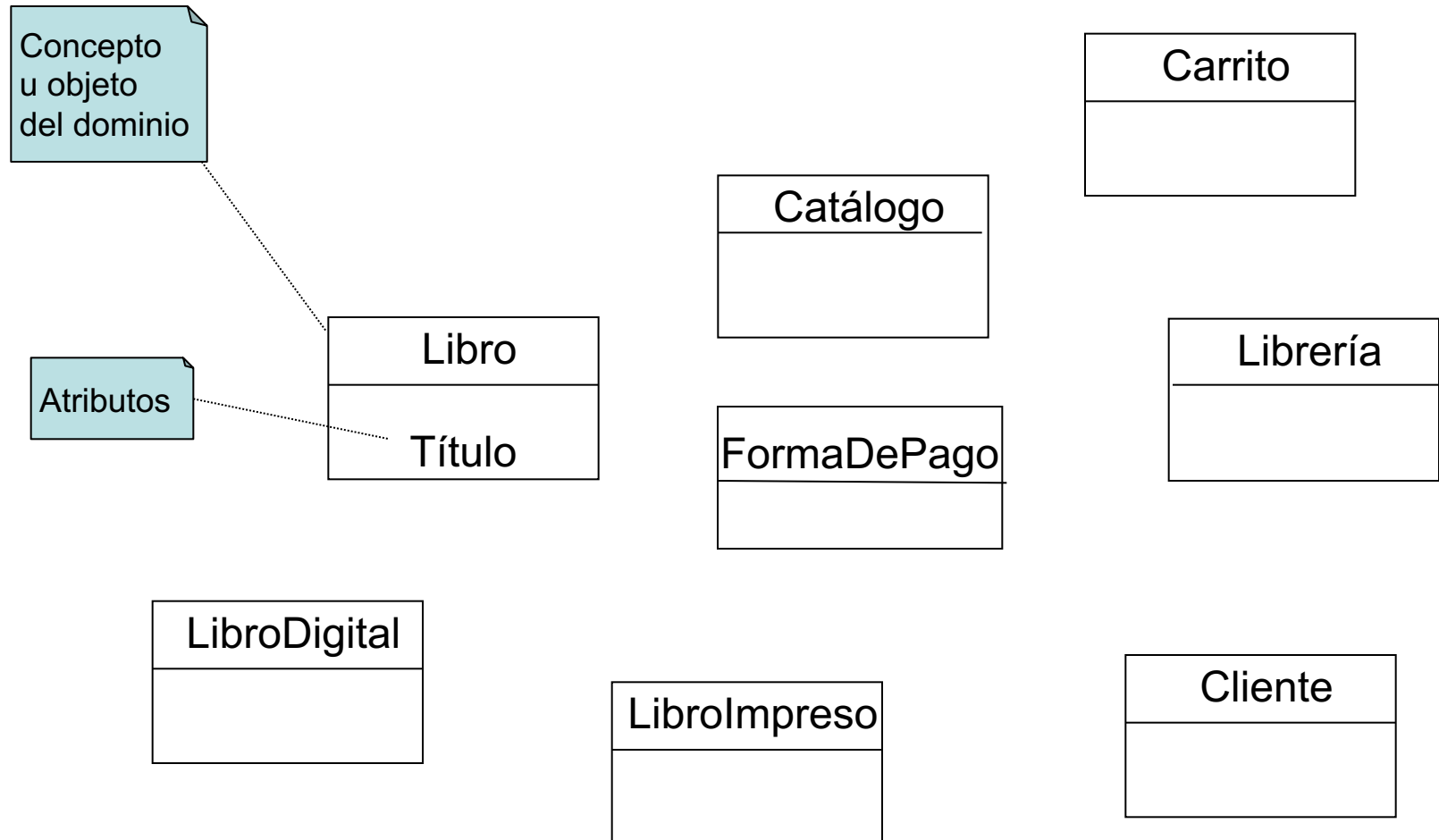
Peor



La mejor manera de expresar que un concepto utiliza a otro es, nuevamente, con una asociación; no con un atributo de clave ajena

# Modelo del Dominio- Visualización de atributos

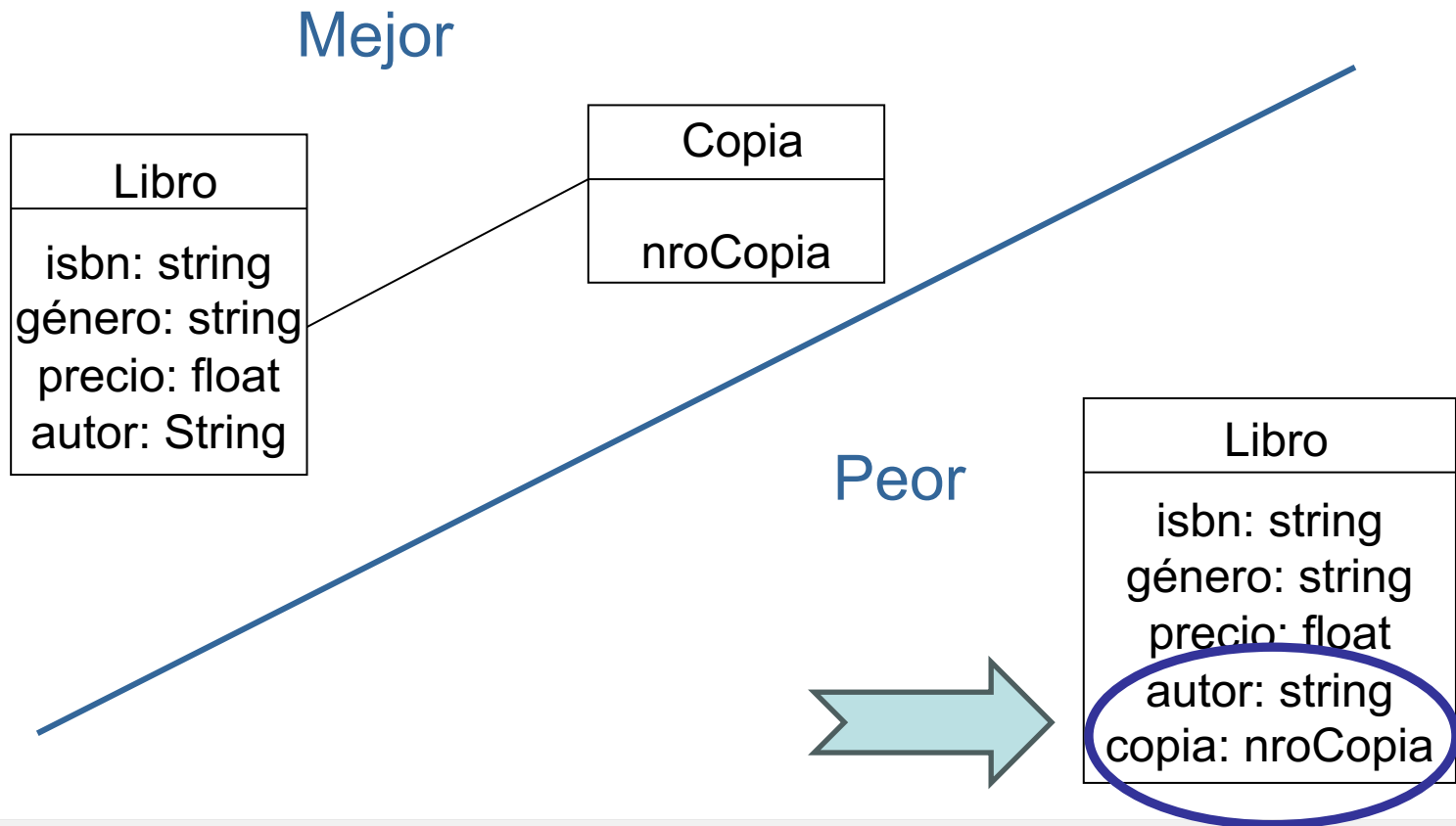
Un **Modelo del Dominio** es una representación visual de las clases conceptuales del mundo real en un dominio de interés.



## (4) Agregando asociaciones- Lista de asociaciones comunes

Categoría	Ejemplo
<b>A es una parte física de B</b>	No aplicable
<b>A es una parte lógica de B</b>	Detalle-Carrito
<b>A está físicamente contenido en B</b>	No aplicable
<b>A está lógicamente contenido en B</b>	Libro-Catálogo
<b>A es una descripción para B</b>	EspecificaciónDeProducto - Libro
<b>A es un miembro de B</b>	Cliente-Librería
<b>A usa o maneja a B</b>	Cliente Carrito
<b>A se comunica con B</b>	Cliente- Librería
<b>A está relacionado con la transacción B</b>	Cliente- Pago Cliente- Agregar al carrito
<b>A es una transacción relacionada con otra transacción B</b>	Pago- Compra
<b>A es dueño de B</b>	Cliente- Carrito

# ¿ Todo en un concepto o un concepto con especificación?



Agregue clases conceptuales de especificación cuando necesite la descripción de un artículo o servicio, o si al eliminar las instancias que describen, se pierde información.

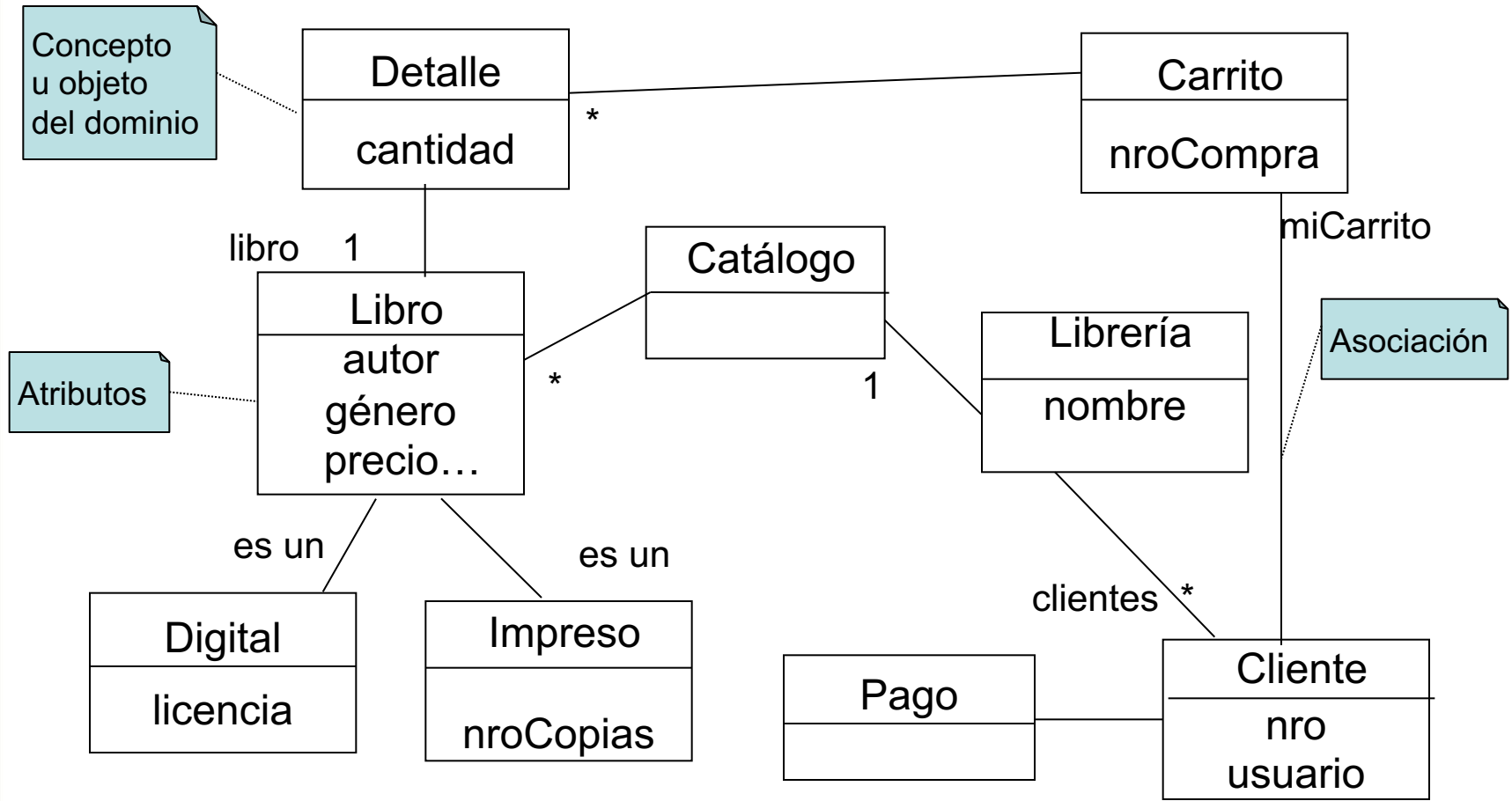
# Agregando asociaciones

Algunos tips:

- Focalizar las asociaciones que necesitan ser preservadas por un lapso de tiempo.
- Evitar mostrar asociaciones redundantes o derivadas.
- Es más importante identificar clases conceptuales que asociaciones conceptuales.
- Demasiadas asociaciones pueden oscurecer el Modelo del Dominio.
- Recuerde agregar multiplicidades.
- Recuerde agregar roles.

# Modelo del Dominio Parcial

Revisar: Sinónimos (Detalle/LineaDeVenta, Libro/ EspecificaciónDelProducto), Conceptos no considerados (Copia), relaciones, multiplicidades, roles.



# Contratos de las operaciones



## Contratos: Describiendo casos de uso

Son una de las formas de describir comportamiento del sistema en forma detallada. Describen pre y post condiciones para las operaciones.

# Secciones del contrato

Las secciones de un contrato son:

- *Operación*: nombre de la operación y parámetros.
- *Precondiciones*: Suposiciones relevantes sobre el estado del sistema o de los objetos del Modelo del Dominio, antes de la ejecución de la operación.
- *Postcondiciones*: el estado del sistema o de los objetos del Modelo del Dominio, después de que se complete la ejecución de la operación.

Las precondiciones:

- Suposiciones relevantes sobre el estado del sistema o de los objetos del Modelo del Dominio, antes de la ejecución de la operación.
- No se validarán dentro de la operación, sino que se asumirán como verdaderas.
- Son suposiciones no triviales que el lector debe saber que se hicieron.
  - Son declarativas (expréselas así)

Ejemplo:

- El cliente está registrado como usuario del Sistema.

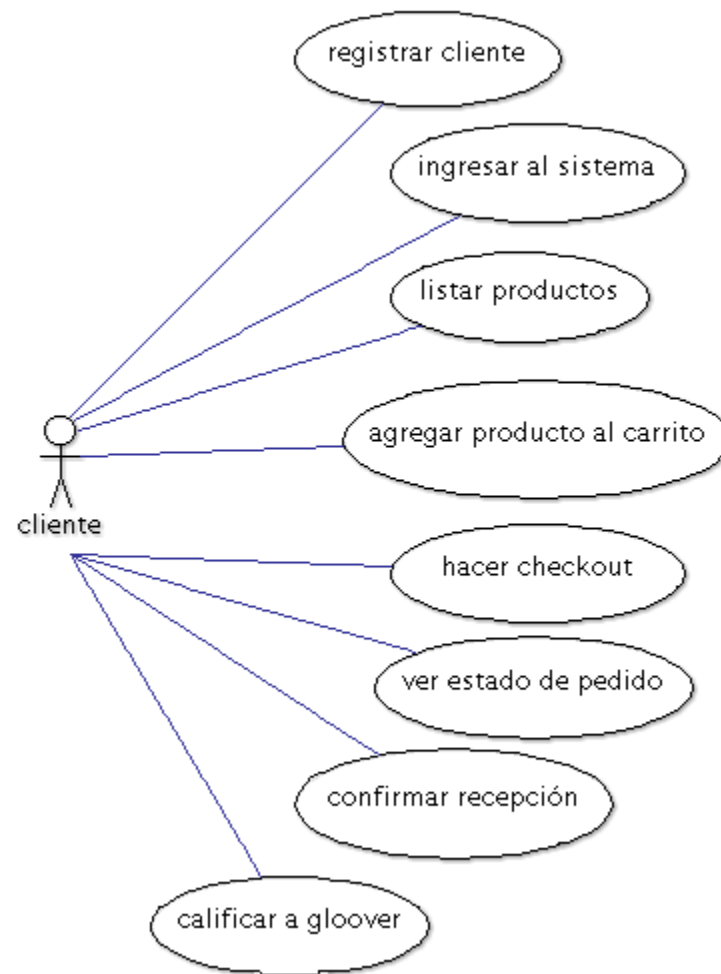
Las postcondiciones:

- Describen cambios en el estado de los objetos del Modelo del Dominio:
  - Creación y eliminación de instancias
  - Modificación de atributos
  - Creación o ruptura de asociaciones
- Son declarativas (expréselas así)

Ejemplo:

- Se creó un nuevo Cliente
- El cliente fue incorporado al Sistema

# Parte del diagrama de Use Cases de Gloovo



# Contrato - Ejemplo con operaciones de Gloovo

*Operación:* registrar Cliente (nombre:String, mail:String, dirección:String, password: String)

*Precondiciones:*

- El cliente no está registrado en el Sistema Gloovo.

*Postcondiciones:*

- Se creó un nuevo cliente.
- Se creó un nuevo carrito para el cliente.
- Se asoció al cliente con el carrito.
- Se agregó el cliente a la colección de clientes de la EmpresaDePedidos.

# Contrato - Ejemplo con operaciones de Gloovo

*Operación:* checkout pedido (c: Cliente)

*Precondiciones:*

- El cliente está registrado en el Sistema Gloovo.
- Existe un carrito, con productos, asociado al cliente.

*Postcondiciones:*

- Se creó un nuevo pedido con el carrito, la dirección de entrega y forma de pago.
- Se agregó el pedido a la colección de pedidos del cliente.
- Se agregó el pedido a la colección de pedidos de la EmpresaDePedidos.
- Se vació el carrito del cliente.

- Crear diagramas de interacción que muestran cómo los objetos se comunican con el objetivo de cumplir con los requerimientos capturados en la etapa de análisis.
- A partir de los diagramas de interacción, diseñar diagramas de clases representando las clases que serán implementadas.

Crear diagramas de interacción requiere la aplicación de Principios o Heurísticas para la Asignación de Responsabilidades.



# Principios o Heurísticas para la Asignación de Responsabilidades

Continuará ...