



Universidad de Buenos Aires  
Facultad de Ciencias Económicas



**Tema a Exponer:**

4.2.3. Métodos Estructurados de Análisis y Diseño.

Métodos Orientados a Objetos. Conceptualización.

Objetivos y características de los principales artefactos utilizados.

# Objetivos de Aprendizaje

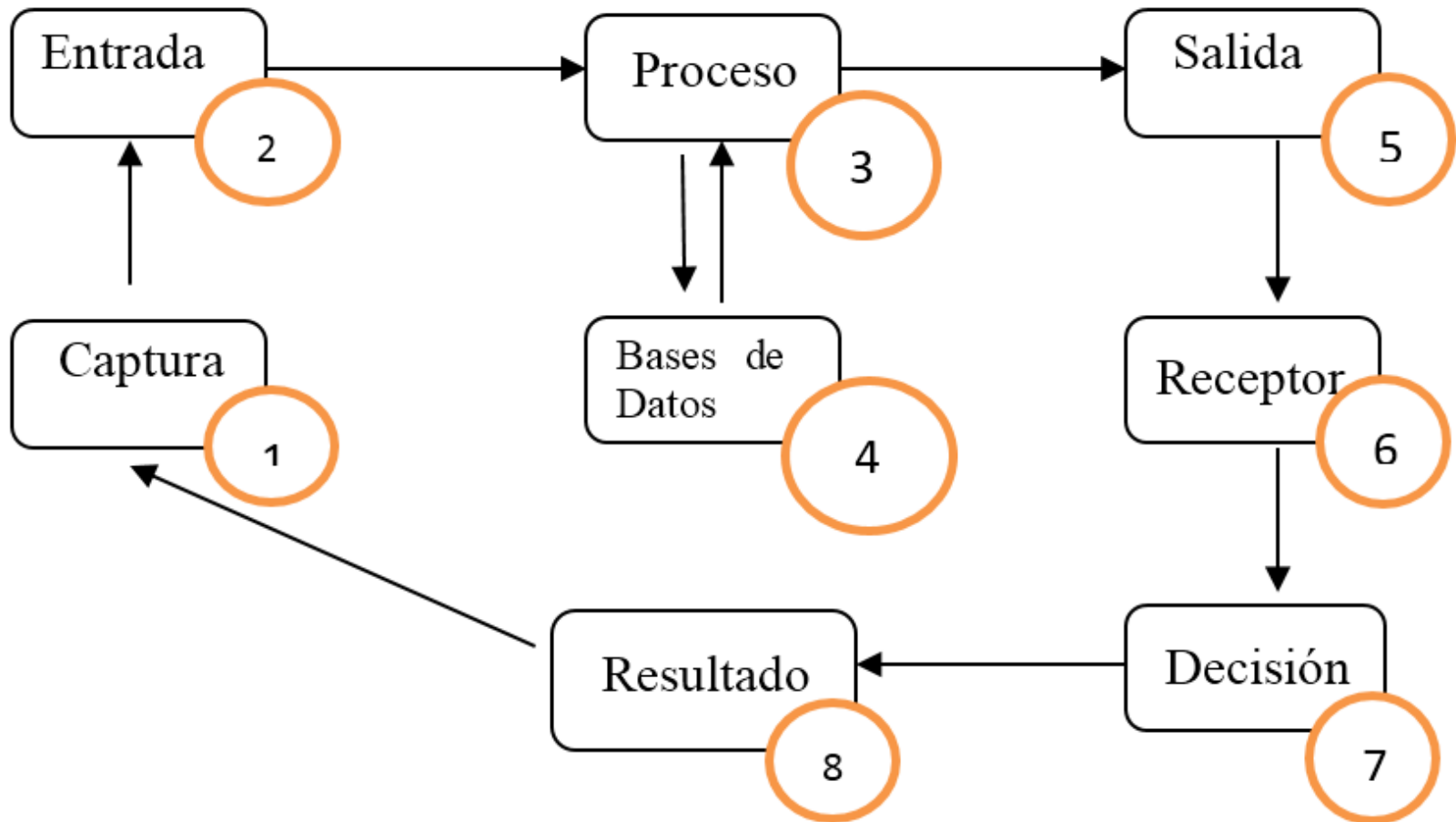
- Generales de la Unidad
  - Comprender
    - el concepto de Ciclo de Vida de los Sistemas de Información
    - las alternativas de desarrollo y adquisición,
    - las metodologías de evaluación y selección de productos de software y hardware,
    - los modelos de desarrollo de sistemas y las actividades necesarias para la implementación y mantenimiento de Sistemas de Información en las organizaciones
  - Específicos de la Unidad 4.2.3
    - Comprender los métodos estructurados de Análisis y Diseño orientados a Objetos

# Bibliografía

## Utilizada para el Dictado del Tema

- Sistemas de Información Gerencial - Tecnología para agregar valor a las Organizaciones.
  - Autores: J.C. Briano | C. Freijedo | P. Rota | G. Tricoci | C. Waldbott
  - Prentice Hall – Pearson Education – año 2011
  - **Capítulo 12: Metodologías de Análisis y Diseño.**

# Un sistema de información



# En los sistemas de información concurren...

- Tecnologías de Modelos de Desarrollo
- Tecnologías de Gestión de Proyectos
  - Selección
  - Desarrollo
  - Incorporación
  - Operación de los Sistemas
- Tecnologías de Análisis y Diseño de Software
  - Tradicionales
  - Modernas

# Una metodología

- Propuestas teóricas para el desarrollo de sistemas que incluyen los artefactos (básicamente procesos y herramientas) para desarrollar la aplicación.


# Todas las metodologías son “estructuradas” (aunque no se llamen así)

- Es “estructurada” porque sigue una serie de pasos o reglas respecto de cómo se debe:
  - Analizar
  - Documentar
  - Convertir los requisitos de un negocio en
    - Especificaciones (Funcionales y Técnicas)
    - Software
    - Configuraciones de hardware y
    - Procedimientos relacionados.



# H-S-B-C-P-P-S-C

	Procesos de los Sistemas de Información			
Componentes	Captura	Procesamiento	Almacenamiento	Distribucion
Hardware				
Software				
Bases de Datos				
Comunicaciones				
Personas				
Procedimientos y Politicas				
Servicios				
Contexto				



Relaciones entre procesos y componentes

# Una metodología de análisis es fundamental para...

- Articular sus elementos en forma lógica, organizado según un cierto orden.
- Servir como elemento de comunicación y como herramienta de documentación.
- Lograr que el analista interprete las necesidades, acordando la tarea a realizar (tener en claro qué se hace y no se hace, sin generar falsas expectativas).
- Lograr que los constructores interpreten las especificaciones, construyendo lo diseñado.

# Metodologías de Análisis y Diseño

- Estructuradas - ADE
  - Construcción de los sistemas desde dos visiones complementarias:
    - **Visión desde los Procesos**
      - Diagrama de Flujo de Datos
    - **Visión desde los Datos**
      - Diagramas de Entidad - Relación
- Orientadas a Objetos - ADOO
  - Une los **Datos** y los **Procesos** en **OBJETOS**

# **Análisis y Diseño Estructurado**


## **Visión desde los Procesos**

# Diagrama de Flujo de Datos

- Describe el sistema como una red de "procesos" conectados mediante "flujos de datos" entre:
  - ellos mismos,
  - con agentes externos (usuarios u otras aplicaciones) y
  - con almacenamientos de información

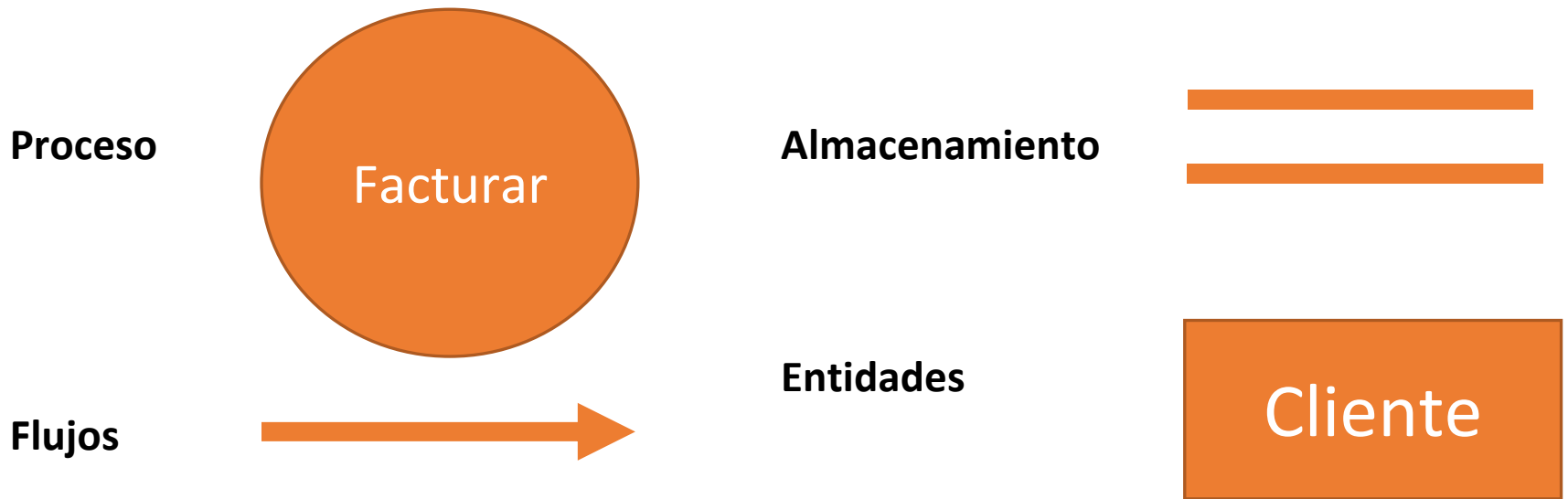
# DFD - Niveles

- Diagrama de Contexto o DFD de nivel 0,
  - Exponer la interacción entre el sistema y los agentes externos que actúan como fuentes y destinos de los datos.
  - Muestra todo el sistema como un proceso único.
- Este DFD se "explota" (expande a mayor detalle) en el nivel 1,
  - Se desagregan los principales procesos del sistema modelado y
  - su relación con los almacenamientos de información internos del sistema y los agentes externos señalados en el nivel 0.
- Sucesivamente cada proceso se "explota" en el nivel siguiente, **respetando la relación con agentes externos y almacenes de información**, y agregando los almacenes internos de ese nivel.
- Al llegar al último nivel de descomposición, el comportamiento del proceso se detalla para su codificación **fuera del DFD**, utilizando:
  - Lenguaje estructurado / pseudo código
  - Tablas de decisión
  - Árboles de decisión

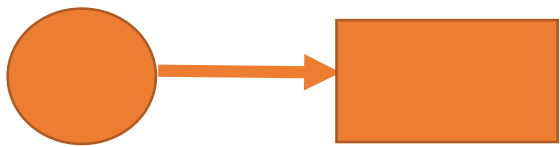


Crítica: la metodología no está auto-contenida

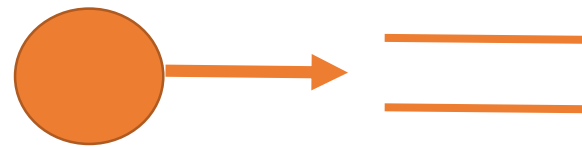
# DFD - Elementos



# DFD – Interacción entre Componentes



Emisión



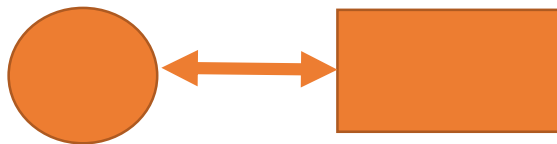
Altas / Bajas



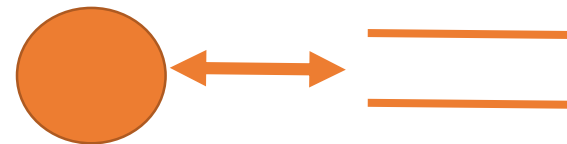
Recepción



Consulta

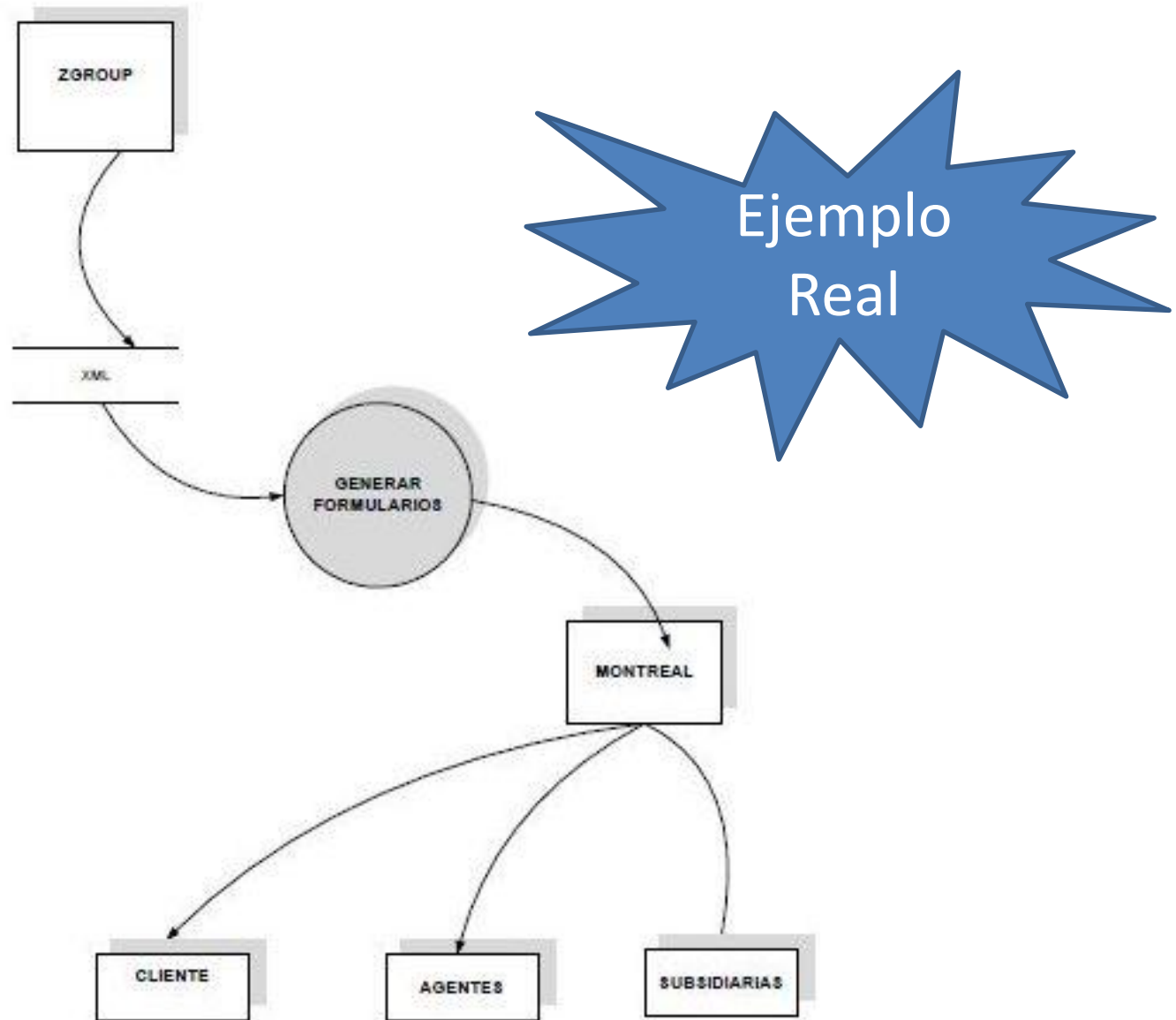


Interacción

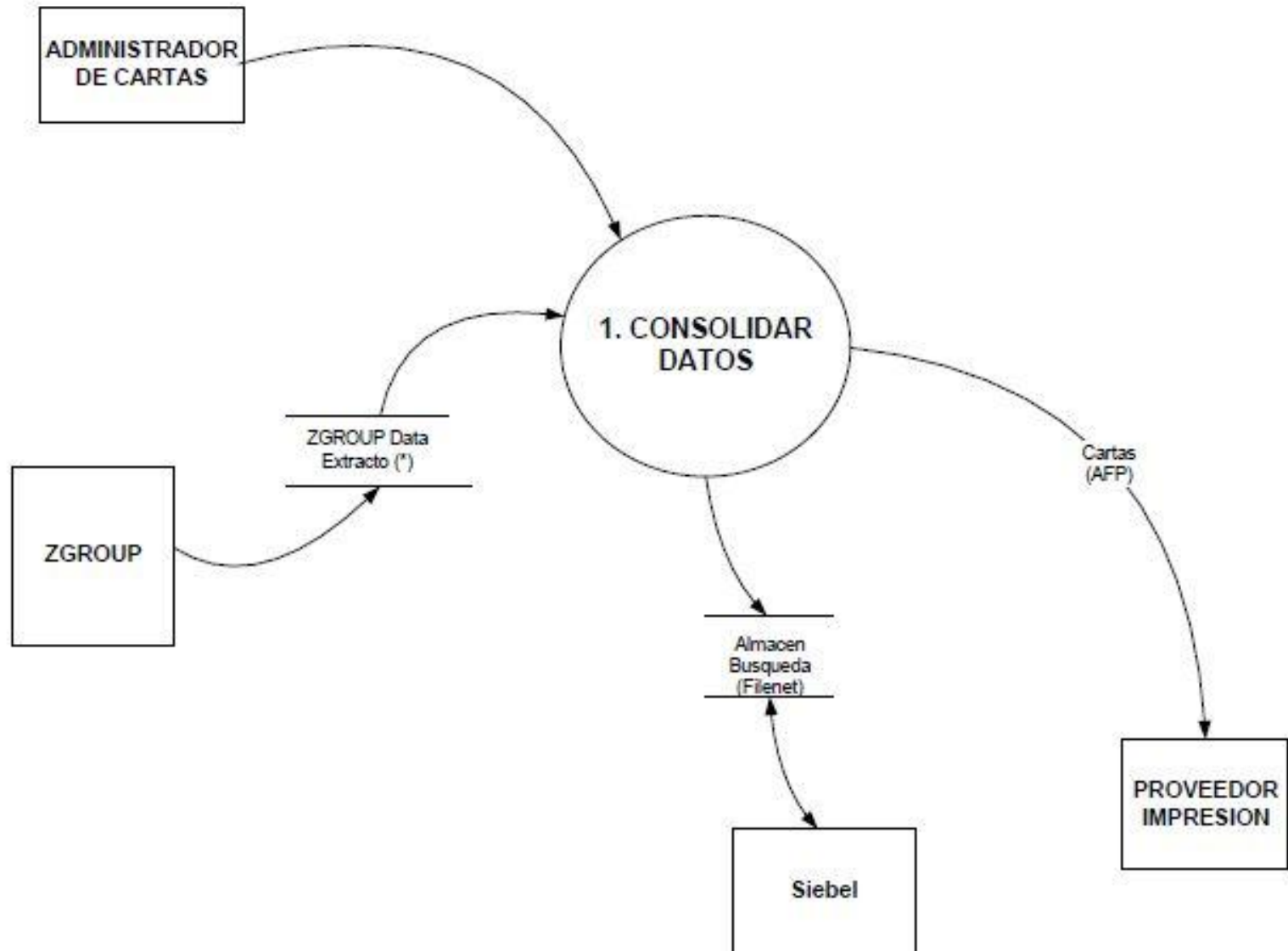


Actualización

# DFD – Diagrama de Contexto

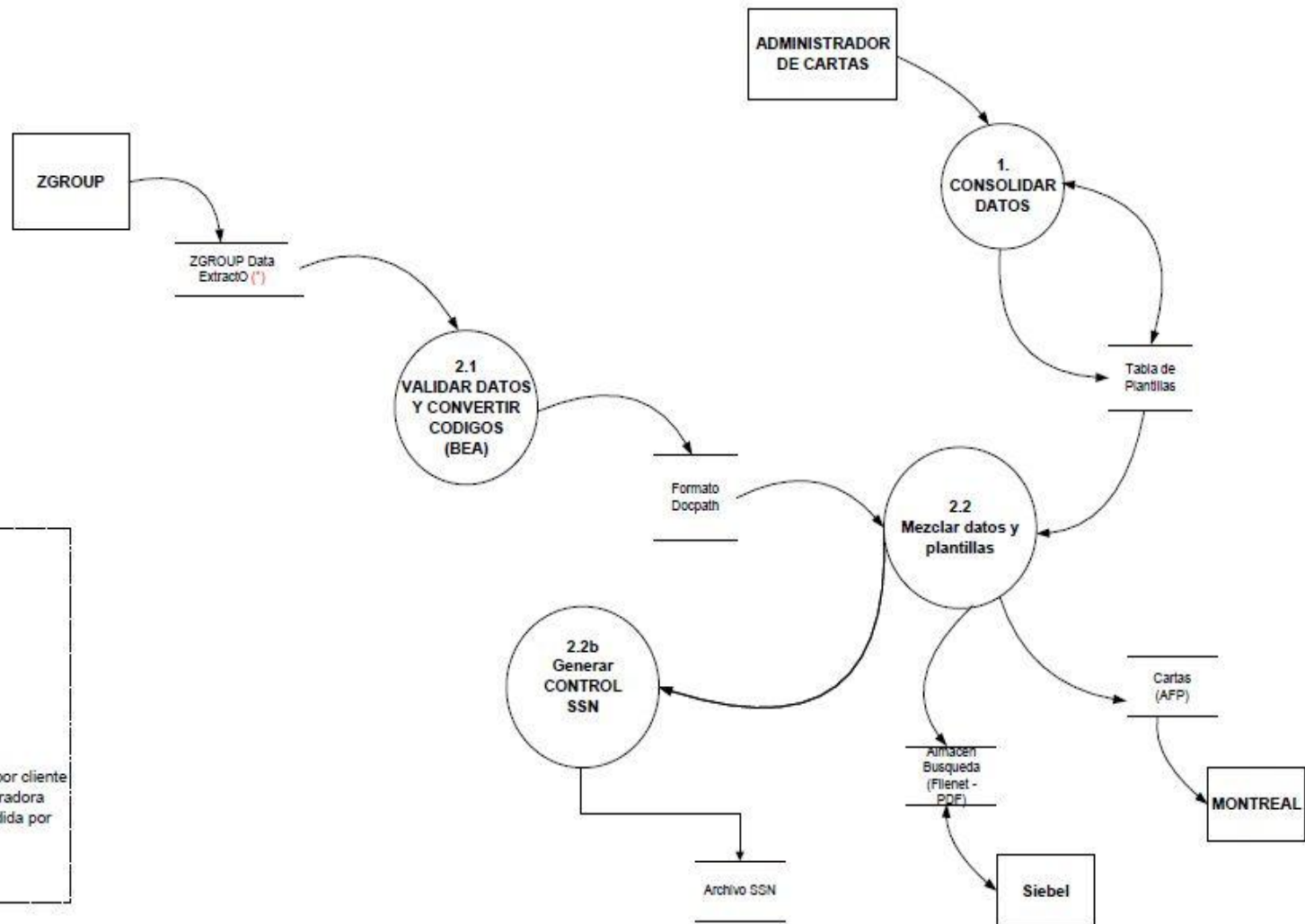


# DFD – Nivel 1



# DFD – Nivel 2

## CARTAS – DFD – Nivel 2



(\*) Notas:

H1 Nuevos Clientes

- Carta de Bienvenida
- Poliza
- Condiciones Comerciales

H2 Cambios en el Contrato

- Endoso

H3 Finalizacion de Cobertura

- Carta de cancelación de póliza pedida por cliente
- Carta de cancelación pedida por Aseguradora
- Carta de cancelación de subsidiaria pedida por cliente

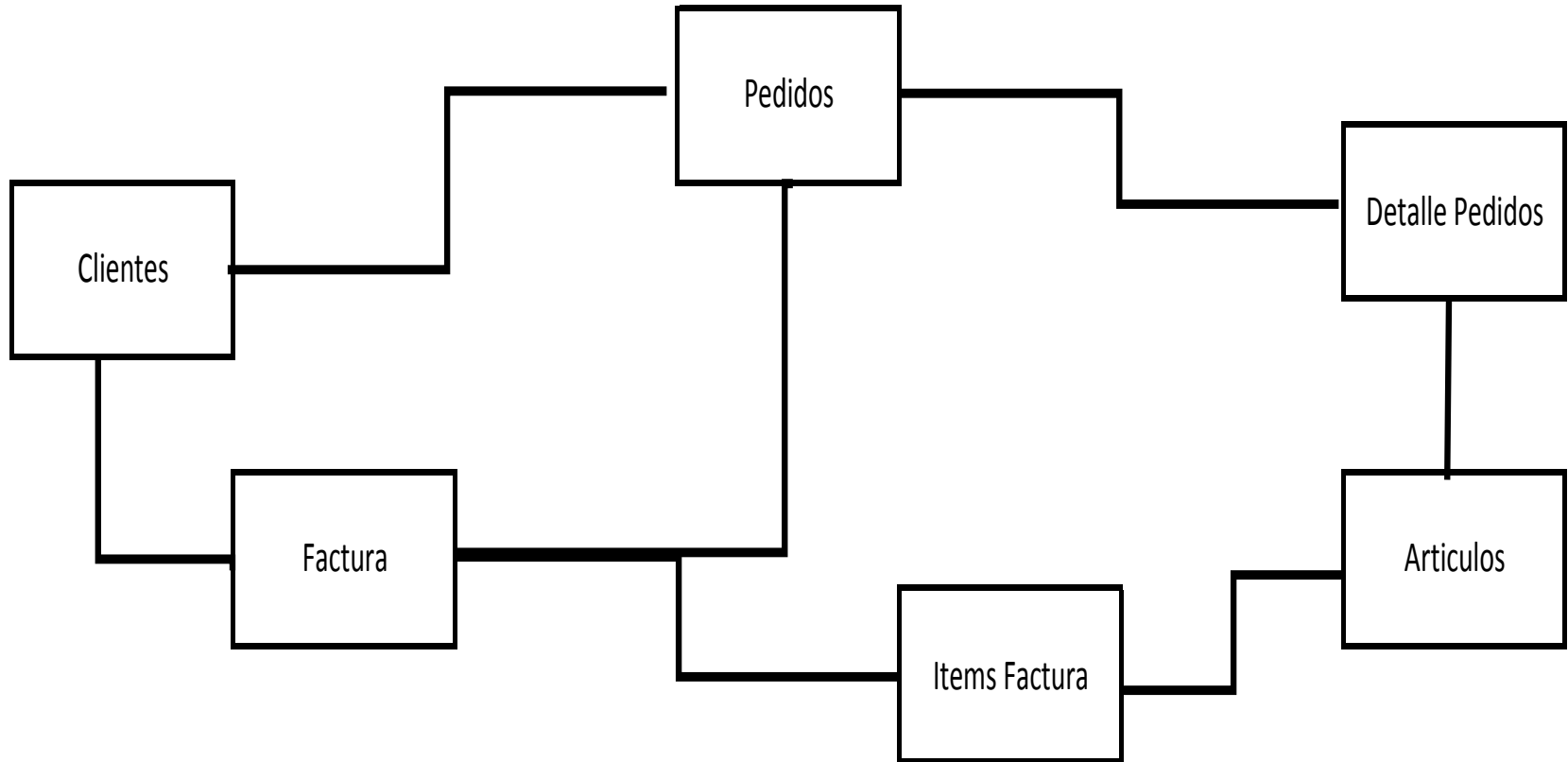
# **Análisis y Diseño Estructurado**

## **Visión desde los Datos**

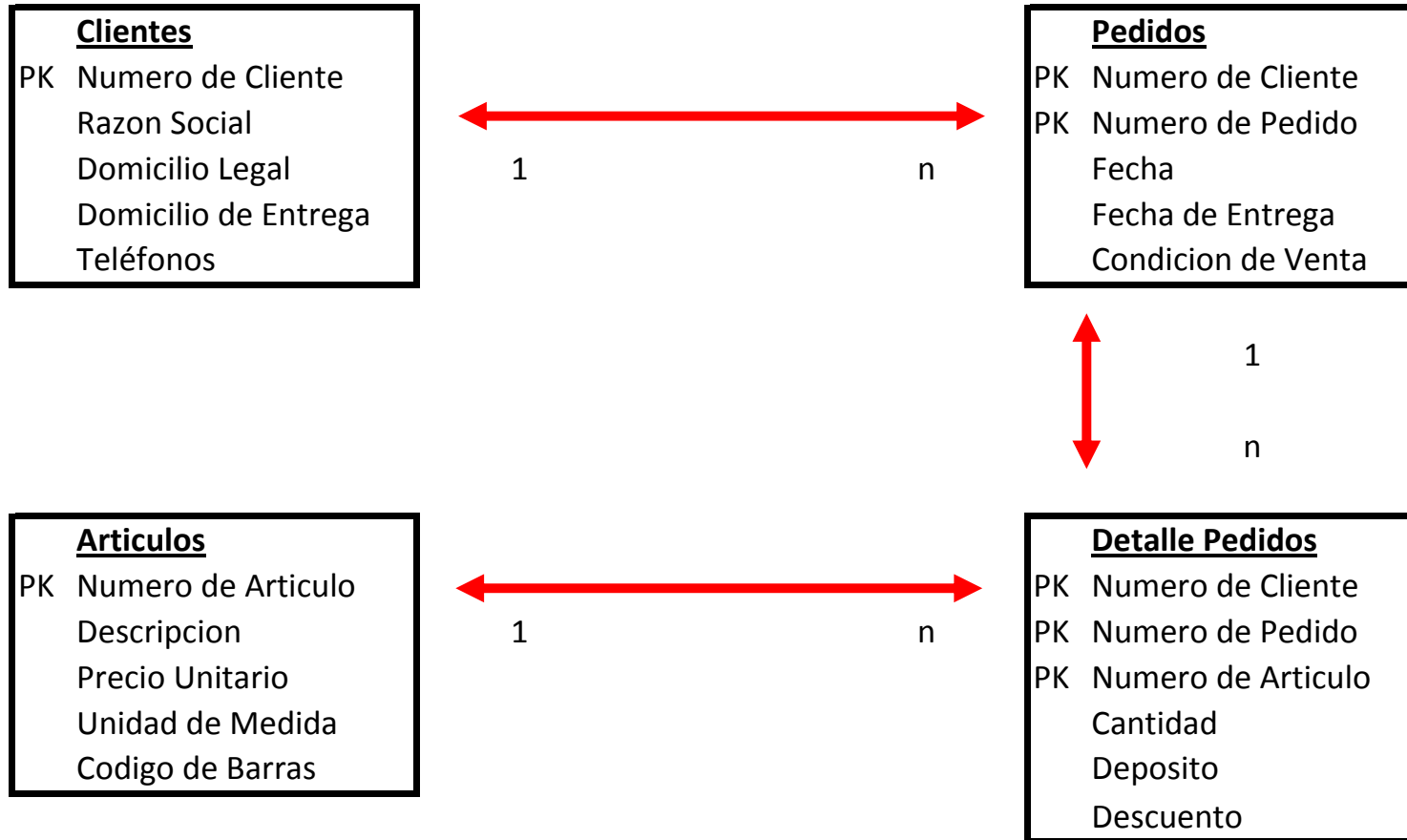
# Diagrama Entidad - Relación

- Formulación de la estructura lógica de datos (en la cual son agrupados en entidades) para soportar los procesos del sistema
  - Utilizando la técnica de normalización y
  - graficando el resultante en el diagrama de entidad relación,
- La técnica de normalización parte de:
  - la identificación de todos los elementos de la base,
  - analiza las relaciones subyacentes entre ellos y
  - permite determinar la mejor forma de organizar los datos en tablas, en función de esas relaciones.
- Aplicación de principios de álgebra.

# Diagrama Entidad – Relación Alto Nivel



# Diagrama Entidad – Relación Bajo Nivel



# DER: La Normalización

- Formalizada por E. F. Codd en “A relational Model of Data for large shared data Banks”, 1970.
- Los estudios de Codd demostraron que todas las relaciones entre datos pueden:
  - resumirse a relaciones simples entre tablas de dos dimensiones (filas y columnas), y
  - que la estructura así determinada es la más simple que puede establecerse para representar adecuadamente el sistema objeto,
  - dando mayor facilidad tanto para responder preguntas, que pueden resolverse con los elementos ya contenidos, o
  - para responder nuevos requisitos, que necesiten el agregado de elementos adicionales.
    - Recordar: Archivos Maestros y Transaccionales

## DER: Normalización

- Las entidades así creadas se denominan relacionales; y esta teoría es la base sobre la cual se desarrollan los sistemas administradores o de gestión de bases de datos relacionales.
  - No es necesario contar con una base de datos relacional para reconocer la estructura lógica de datos, que puede implementarse tanto con bases relacionales como con bases jerárquicas o archivos tradicionales.
- El proceso de normalización garantiza que la estructura de datos así determinada es la que mejor representa la realidad subyacente a los datos requeridos por el sistema.

## DER: Normalización

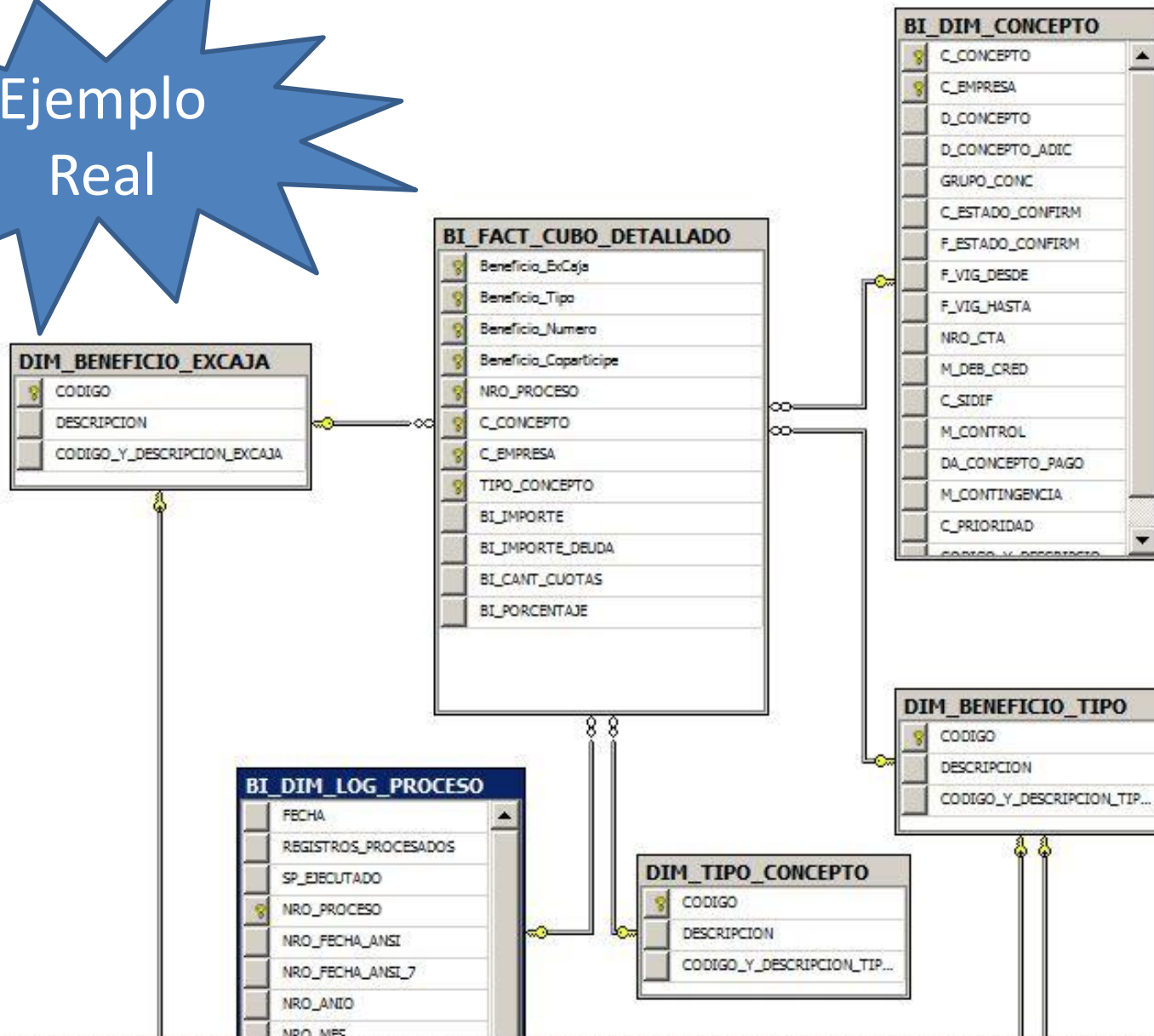
- Luego de consensuado el diseño lógico, se realiza su implementación física.
- Para esta implementación se tienen en cuenta las restricciones tecnológicas.
  - Muchas veces es necesario “des normalizar” para alcanzar rendimiento en los sistemas (restricciones impuestas por el nivel tecnológico disponible como ser capacidad de almacenamiento y tiempo de respuesta)
- No obstante se busca definir adecuadamente la estructura lógica, como marco de referencia de cualquier tarea posterior.

# DER: Normalización

- Se hace conveniente contar con herramientas adicionales:
- Diccionario de Datos (DD)
  - Repositorio integrado de todos los datos ingresados, producidos, administrados y entregados por el sistema.
- Diagrama de Transición de Estados (DTE)
  - Modelización del comportamiento.
  - Representa el comportamiento de un sistema exponiendo los eventos que producen que el sistema cambie de estado y destaca qué acciones se llevan a cabo como consecuencia de ese cambio.

# Normalización en SQL Server

Ejemplo  
Real



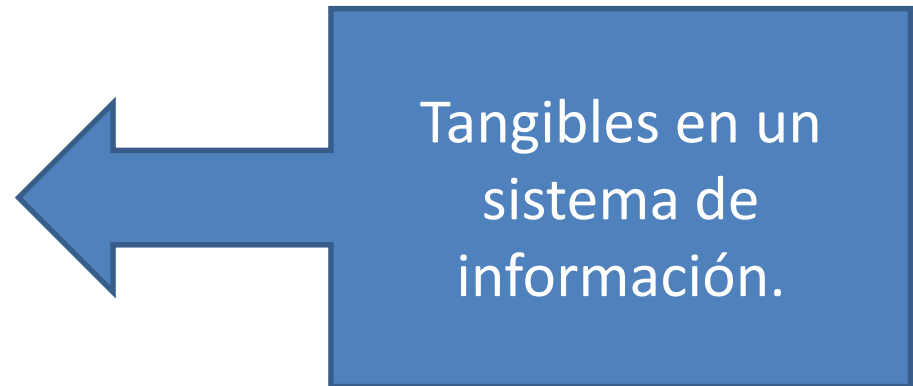
# **Análisis y Diseño Orientado a Objetos**

# Métodos Orientados a Objetos

- A diferencia de los métodos estructurados, que separan datos de procesos, el enfoque de análisis y diseño orientado a objetos (ADOO) **une datos y procesos en artefactos denominados "objetos"**.
  - Mientras que el enfoque tradicional se basa en el **análisis de eventos** y la determinación de su **equivalente lógico**,
  - el enfoque Orientado a Objetos **requiere que esos eventos pertenezcan a un "objeto" identificable**.

# Métodos Orientados a Objetos

- Un objeto puede ser:
  - un lugar,
  - una persona
  - una cosa relevante para el sistema,
  - Ejemplos:
    - Un cliente,
    - una factura,
    - un empleado,
    - un proveedor.



# Métodos Orientados a Objetos

- ADOO supone un avance en cuanto a la reutilización e integración aplicativa con relación a los métodos estructurados
  - donde los datos son compartidos mientras que
  - los procesos son específicos para cada aplicación.
- Las actividades de desarrollo se centran en los objetos.
  - El software se organiza a partir de los elementos que existen en el dominio del problema.
  - Crítica: se debe recurrir a otras técnicas como “base” de análisis (DFD y DER)

# Métodos Orientados a Objetos

- ADOO tiene por objetivo:
  - La construcción de un modelo que interprete la complejidad subyacente en el sistema objeto y
  - la determinación de su equivalente lógico,
- No necesariamente la aplicación de herramientas ADOO implican la programación orientadas a objetos.
  - Aunque funciona mejor cuando ambas van de la mano...
  - Lenguajes OOP (Java - VB y C# .NET – PHP)

# Métodos Orientados a Objetos

- En los términos de ADOO un objeto es todo conjunto integrado por componentes esenciales:
- **Atributos** (datos organizados).
- **Servicios** (referentes lógicos de los procesos de transformación, operaciones, los cuales reciben y entregan información al exterior del objeto por medio de parámetros).
- **Métodos** (forma en que se implementan los servicios)
  - Un mismo servicio puede implementarse con diferentes métodos, dependiendo de la tecnología que se utilice, siendo esto transparente para el usuario).

# Métodos Orientados a Objetos

- Atributos + Servicios + Métodos = "encapsulado"
- Protege los datos del uso arbitrario.
- Oculta los detalles de la implantación interna a los usuarios de un objeto,
  - los usuarios conocen los servicios que puede solicitar del objeto,
  - pero desconocen los detalles de cómo se llevan a cabo.
- Separar el comportamiento de su implantación
  - permite la modificación de ésta sin que se tengan que modificar las aplicaciones que lo utilizan, en la medida que se mantengan los servicios.

# Métodos Orientados a Objetos

- Ejemplo:
- El objeto Factura puede tener como servicios:
  - Nombre del Cliente Destinatario
  - Importe Total de la Factura
  - Condiciones de Venta
  - Impuestos y Retenciones
  - Información Fiscal (Ej. Factura Electrónica)
  - Etc....

# Métodos Orientados a Objetos

- Los distintos objetos se comunican por "mensajes".
  - Un mensaje solicita un servicio que ejecute el método apropiado y, en su caso, realice una modificación de datos y/o produzca una respuesta.
  - El mensaje que constituye contiene el nombre del objeto, el nombre del servicio y, según corresponda, un grupo de parámetros.
    - Proliferación de "Web Services"
- De esta forma se pueden "armar" aplicaciones nuevas combinando, mediante mensajes, objetos existentes, integrando en objetos y construyendo los objetos no existentes.
- Un objeto puede estar compuesto por otros objetos, formando un objeto complejo.

## A modo de ejemplo...

- Consulta de Saldo Bancario en ADOO
  - el objeto que provee (tiene el servicio) de "dar saldo de cuenta" será el mismo si el saldo se consulta desde una posición de caja, desde un cajero automático o desde internet,
  - será el mismo si se lo solicita para determinar si se paga o no un cheque contra ese saldo.
- En cambio, si se realizase por ADE Tradicional
  - La consulta desde los procesos y los datos debe repetirse para cada instancia que se necesite (saldo en ventanilla, ATMs, e-banking, IVRs, etc.)

# Métodos Orientados a Objetos – Características 1/3

## Clasificación

- Una clase es un grupo de objetos que tiene atributos y comportamientos similares.

## Identidad o instanciación

- Objetos con iguales atributos y servicios son distinguibles entre si, debido a que tienen una característica distintiva de "identidad".
  - Por ejemplo, dentro del objeto "factura", el número de factura (ya sea legal o interno) le da "identidad".

# Métodos Orientados a Objetos – Características 2/3

## Jerarquía y herencia

- Las clases se encuentran relacionadas jerárquicamente, y comparten atributos y servicios tomando como base esa relación jerárquica.
- Una clase puede incluir sub-clases de nivel jerárquico inferior.
- Esta es una característica fundamental y trascendente, ya que permite que conociendo el comportamiento de la "clase" se sabe que la subclase tiene el mismo comportamiento
  - más otros comportamientos adicionales específicos de ella.

# Métodos Orientados a Objetos – Características 3/3

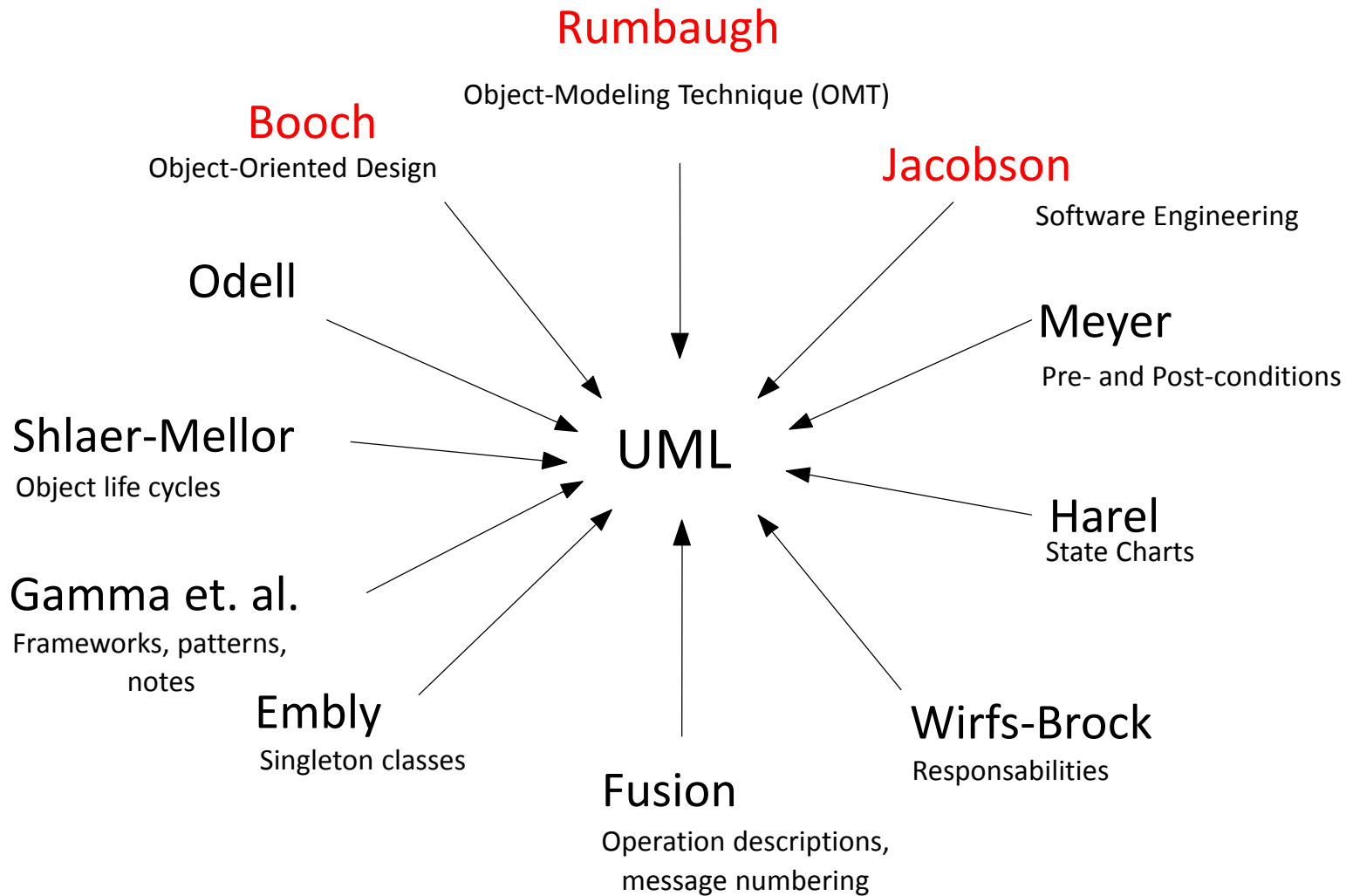
## Polimorfismo (múltiples formas)

- Un mismo servicio puede comportarse de manera diferente en distintas instancias de una misma clase, por aplicación de un método diferente.
- Así un usuario no necesita conocer el método aplicado para una operación o servicio, y, a la vez, se pueden agregar instancias nuevas a una clase en la medida en que el objeto de la instancia tenga el servicio con su método.
  - El control de CUIT / CBU es el mismo para clientes y proveedores -> Mismo “Objeto”: se lo invoca en desde una instancia distinta para que haga los mismo:  
Validar la CUIT

# Métodos Orientados a Objetos - Herramientas

- Encontramos muchas herramientas utilizadas en metodologías orientadas a objetos
  - algunas de ellas superpuestas.
- Esto hace que los artefactos correspondientes a las metodologías orientadas a objetos...
  - resulten complejas y
  - requieran una mayor formación para su interpretación que los artefactos de diseño estructurado.
    - La formación arranca en ADE tradicional...

# ADOO – Iniciativas Históricas



## ADOO ->UML

- Un esfuerzo de unificación de estas iniciativas se realizó con la construcción del lenguaje unificado de modelado, UML (Unified Modeling Language).
- Diseñado en su primera versión por un trabajo conjunto de autores del enfoque de diseño orientado a objetos:
  - James Rumbaugh, Grady Booch e Ivar Jacobson.
  - Fusionaron sus investigaciones en las Herramientas de UML

# UML

- UML puede ser utilizado por cualquier metodología de análisis y diseño orientada por objetos para expresar los diseños.
- “Lenguaje cuyo vocabulario y reglas se centran en especificar, construir, visualizar y documentar los objetos en una forma conceptual y física de un sistema” (Booch, Jacobson y Rumbaugh).

# “Artefactos” de Modelado UML

## Diagramas estructurales

- **1. Diagrama de clases**
- 2. Diagrama de componentes
- 3. Diagrama de objetos
- 4. Diagrama de estructura compuesta
- 5. Diagrama de despliegue
- 6. Diagrama de paquetes



## Diagramas de comportamiento

- **7. Diagrama de actividades**
- **8. Diagrama de casos de uso**
- 9. Diagrama de estados

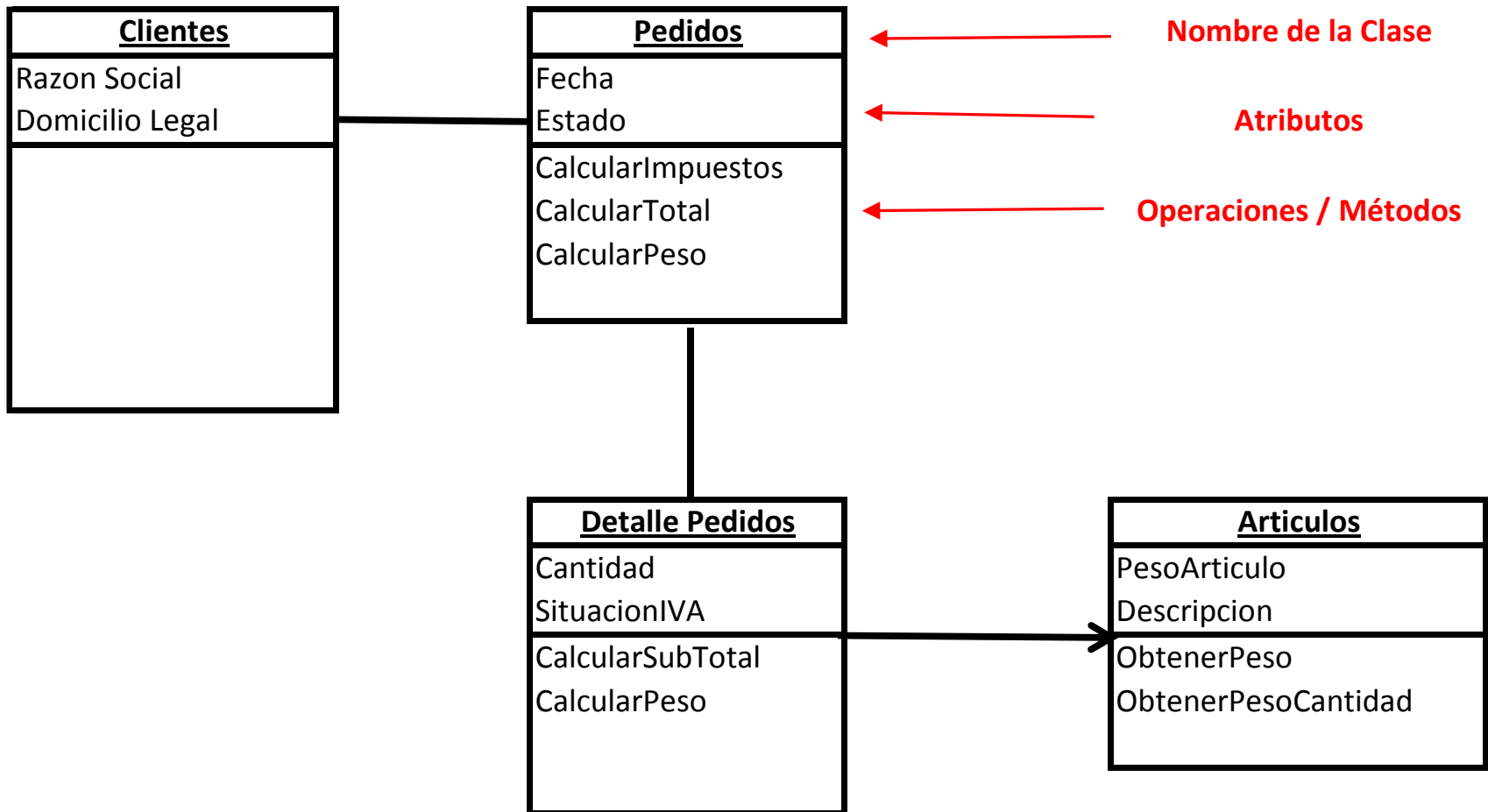


## Diagramas de Interacción (subtipo de diagramas de comportamiento)

- **10. Diagrama de secuencia**
- 11. Diagrama de comunicación
- 12. Diagrama de tiempos
- 13. Diagrama global de interacciones



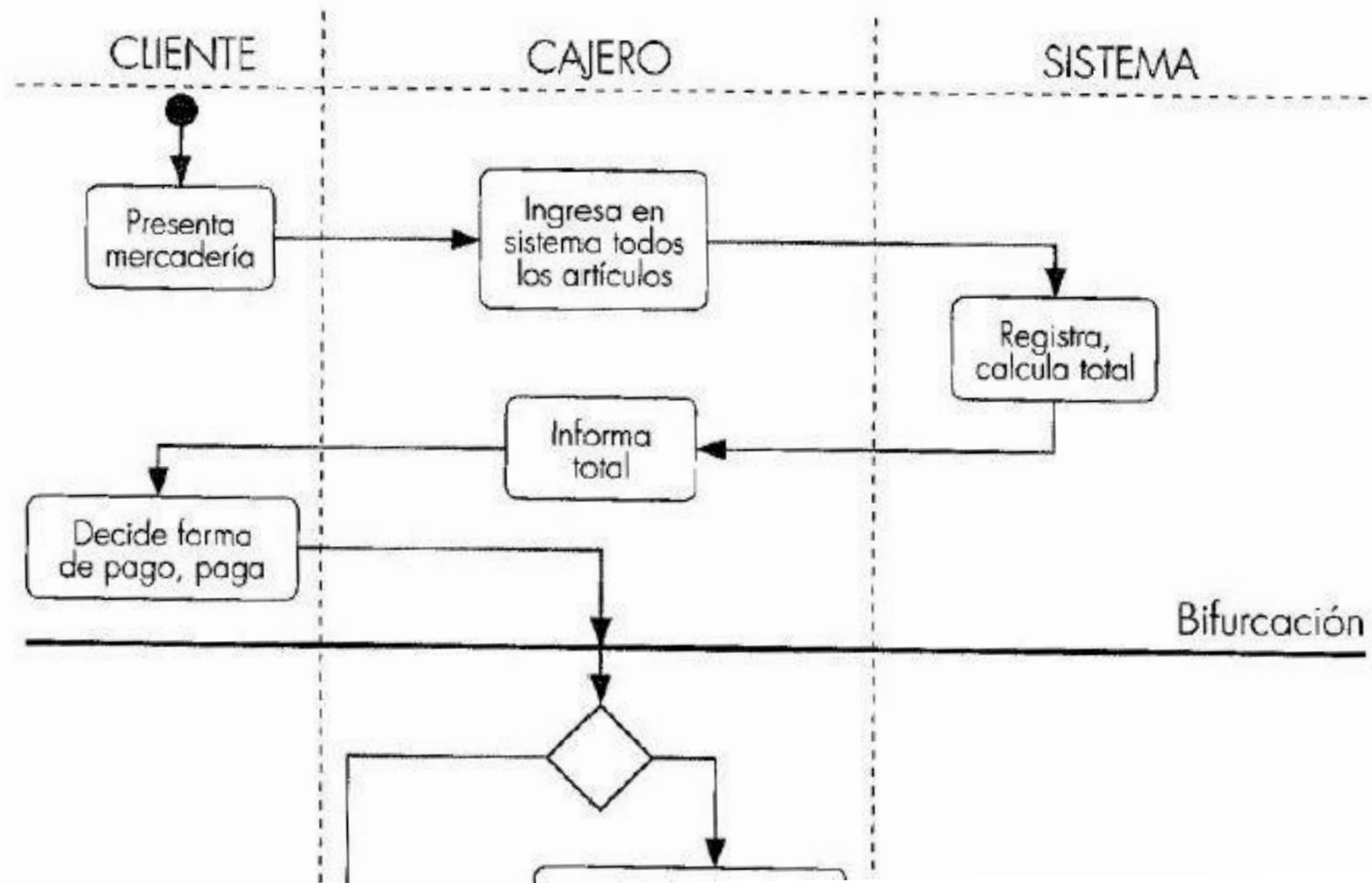
# Diagrama de Clases



# Diagrama de Actividades

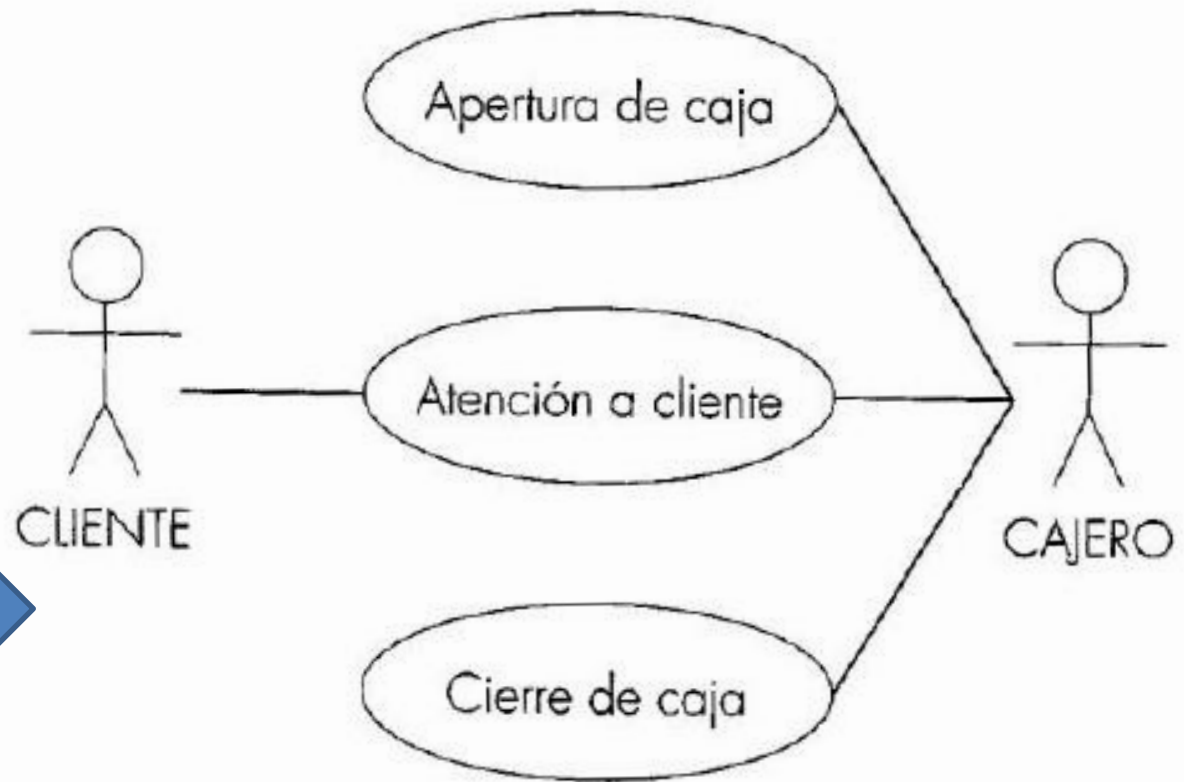
Exponen las actividades de un caso de uso en la forma en que se van dando, incluyendo actividades paralelas y decisiones tomadas.

Tiene una conformación similar al cursograma.

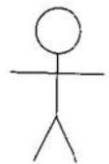


# Diagrama de Casos de Uso

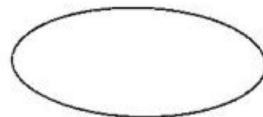
*Actividad diaria  
de un cajero en un  
supermercado*



Más difundido  
para  
documentación  
funcional



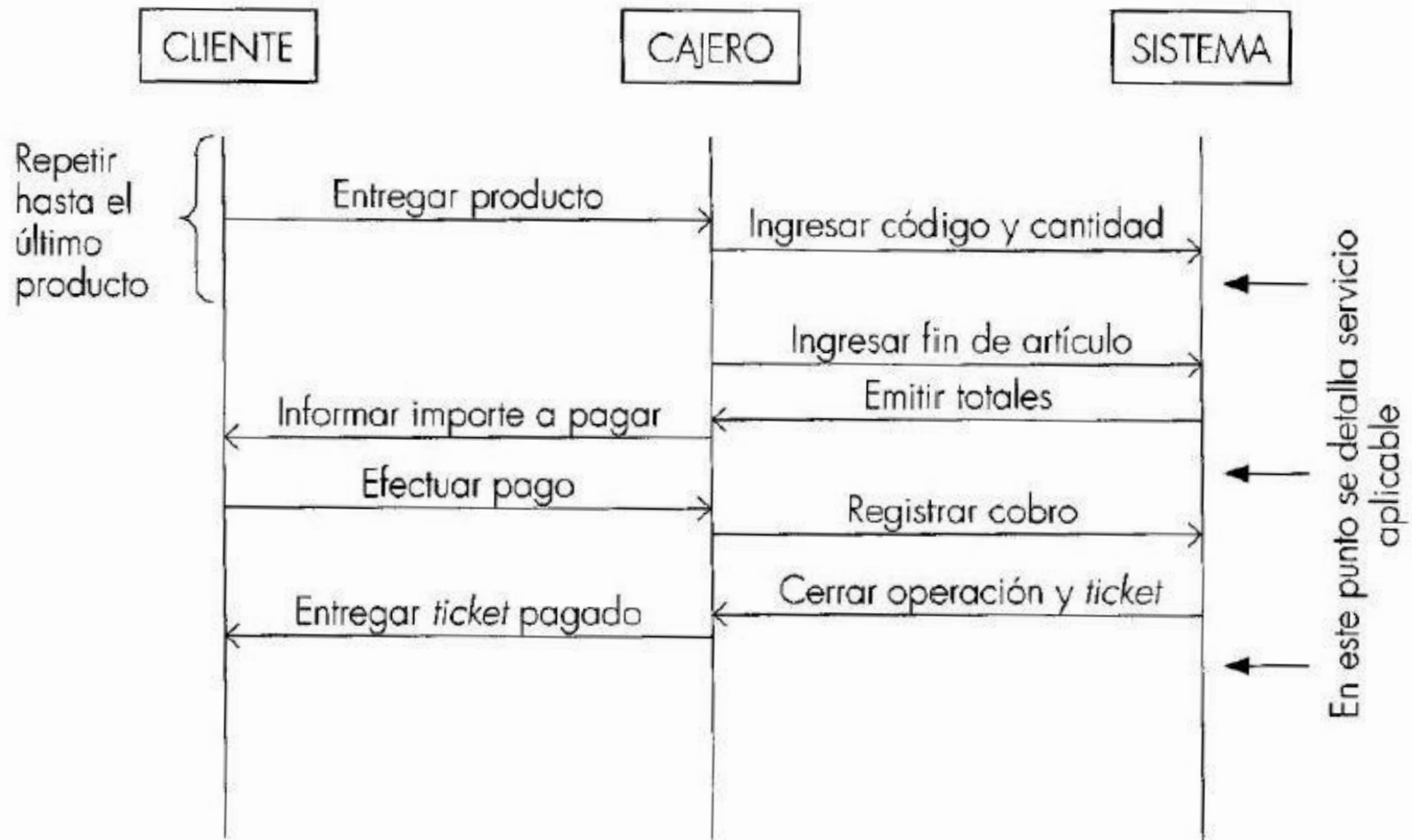
Actores



Funciones

# Diagrama de Secuencia

Muestran la sucesión de actividades entre clases o instancias.



# Resumen y Conclusiones

<b>Análisis y Diseño Estructurado (ADE)</b>	<b>Análisis y Diseño Orientado a Objetos (ADOO)</b>
Examinan los sistemas desde el punto de vista de sus Funciones o Tareas a realizar (Entrada – Proceso – Salida)	Examina el sistema como un conjunto de objetos que interactúan entre sí
Divide los Datos de los Procesos	Une Datos y Procesos (Encapsulamiento)
Utilizan herramientas que analizan el sistema por separado (DFD / DER / DD / Diagramas de Transición de estado, etc.)	Tratan al sistema como un todo en su estructura, comportamiento e interacciones.
Es la metodología utilizada tradicionalmente y con mayor adopción en la industria	Principalmente utilizado en conjunto con técnicas de programación modernas y en la gestión de proyectos “ágil”
El diseño del software comienza una vez finalizado el análisis.	El diseño puede comenzar antes de concluir el análisis.

# Conclusiones

- Sin embargo, ambas ADE y ADOO son:
- Metodologías creadas para desarrollo de sistemas
- Siguen una serie de reglas o pasos para
  - Analizar y Documentar
  - Convertir los requisitos de un negocio en Especificaciones (Funcionales y Técnicas)
  - Desarrollar Software
  - Describir un sistema de información en forma lógica



Universidad de Buenos Aires  
Facultad de Ciencias Económicas



¡Muchas Gracias!

Aníbal Mario Mazza Fraquelli