



# ABCSOFT

*Manual del Usuario*

*Análisis de Sistemas Orientado a Objetos*

Lima-Perú

**2004**

**CONTENIDO**

1.0 TEORIA DE SISTEMAS .....	3
1.1 Concepto de Sistema.....	3
1.2 Jerarquía de los Sistemas.....	3
1.3 Elementos de un sistema .....	5
1.4 Elementos de la Retroalimentación.....	6
1.5 Clasificación de los Sistemas y Enfoque de Sistema .....	6
2.0 CICLO DE VIDA DEL DESARROLLO DE SISTEMAS .....	9
2.1 Ciclo de Vida en Cascada, Fases.....	11
2.2 Ciclo de Vida basada en Prototipos.....	13
2.3 Ciclo de Vida del Sistema Orientado a Objetos .....	15
2.4 Ciclo de Vida en Espiral.....	15
3.0 HERRAMIENTAS Y PROCEDIMIENTOS PARA DESARROLLAR UN SISTEMA.....	16
3.1 La Investigación Preliminar .....	16
3.2 El Plan de Entrevista .....	16
3.3 El Cuestionario - Partes y Tipos de preguntas .....	17
3.4 Recopilación de los datos .....	17
4.0 LOS OBJETOS DEL SISTEMA .....	18
4.1 Definición de un objeto.....	18
4.2 Tipos de Objetos - Objetos - Clases .....	19
4.3 Elementos de los Objetos .....	19
4.3 Jerarquía de los Objetos.....	20
4.5 Características de los Objetos.....	20
5.0 EL MODELO CONCEPTUAL DEL SISTEMA.....	24
5.1 Modelo real - Realidad de los Objetos.....	24
5.2 Modelo de Objetos del Sistema - Transformaciones .....	24
5.3 El Diagrama de Eventos del Sistema - Capas y Niveles .....	25
6.0 ESTRUCTURA DE LA CLASIFICACION DE OBJETOS .....	27
6.1 Estructura de Generalización - Especialización.....	27
6.2 Tipos de Estructuras de Generalización.....	27
6.3 Estructura todo Parte - Tipos.....	28
6.4 Relaciones Entre los Objetos .....	28
6.5 Cardinalidad entre los Objetos .....	29
7.0 DEFINICIONES DE LA ESTRUCTURA DEL OBJETO.....	30
7.1 Atributos del Modelo de Objetos .....	30
7.2 Métodos del modelo de objetos.....	30
7.3 Diagrama de Relaciones del Objeto.....	31
7.4 Diagrama de Eventos del Modelo de Objetos .....	31
8.0 DICCIONARIO DE DATOS DEL SISTEMA .....	32
8.1 Definición del Diccionario de Datos.....	32
8.2 Estructura del dato .....	32
8.3 Conversión del Modelo de Objetos a un Modelo de Datos.....	33
9.0 ANALISIS DE OBJETOS CON UML.....	34
9.1 Diagrama de Caso de Uso .....	34
9.2 Diagrama de Clases y de Objetos.....	36
9.3 Diagrama de Componentes.....	36
9.4 Diagrama de Estado.....	37
9.5 Diagrama de Actividad .....	38
9.6 Diagrama de Secuencia .....	39
9.7 Diagrama de Colaboración.....	40
BIBLIOGRAFIA.....	41

## 1.0 TEORÍA DE SISTEMAS

El término 'sistema' es de uso cotidiano. Hablamos de:

- sistema educativo,
- sistema métrico decimal,
- sistema financiero,
- sistema de información,
- sistema solar
- sistema nervioso,
- sistema judicial, etc.

¿Por qué la palabra 'sistema' ha alcanzado tanta popularidad? La respuesta es que un 'sistema' implica integridad, totalidad y unificación de partes para lograr un funcionamiento óptimo.

### 1.1 Concepto de Sistema

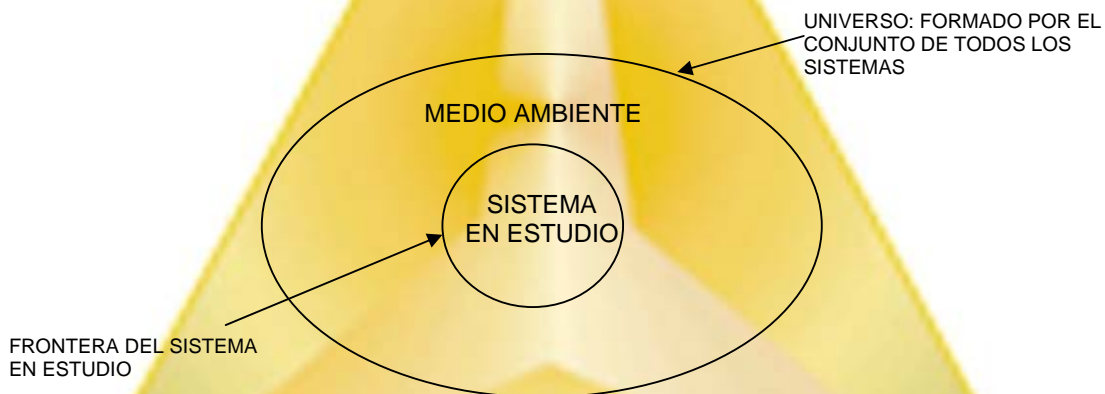
'Sistema es un conjunto de elementos independientes orientados hacia la realización de un objetivo determinado'.

Esta definición conlleva hacer algunas consideraciones:

El sistema está formado por elementos, estos elementos pueden ser máquinas, equipos, personas, recursos materiales en general, ideas, etc.

El objetivo de la organización de estos elementos es común a cada uno de ellos y a la vez al sistema como un todo.

Para un sistema el elemento más importante es el objetivo, cuando el objetivo se modifica entonces significa que el sistema ya no es el mismo, a un nuevo objetivo corresponde un nuevo sistema.



**Fig. 01.** El sistema y su Medio Ambiente

Un sistema es un "subsistema" de un sistema mayor y de igual manera un subsistema cualquiera recibe el tratamiento de sistema. Las interconexiones y las interacciones entre subsistemas se llaman interfaces. Las interfaces ocurren en la frontera y toman la forma de entrada y salidas.

### 1.2 Jerarquía de los Sistemas

Un sistema complejo es difícil de comprender cuando se considera como un todo, por lo tanto, el sistema se descompone en subsistema. La frontera e interfaces deben estar bien definidas, de tal manera que la suma de los subsistemas constituye un sistema completo. Este proceso de descomposición se continúa con los subsistemas que se dividen en subsistemas más pequeños hasta que el menor de los subsistemas tenga un tamaño manejable. Un subsistema es un elemento de un suprasistema o macrosistema (sistema mayor)

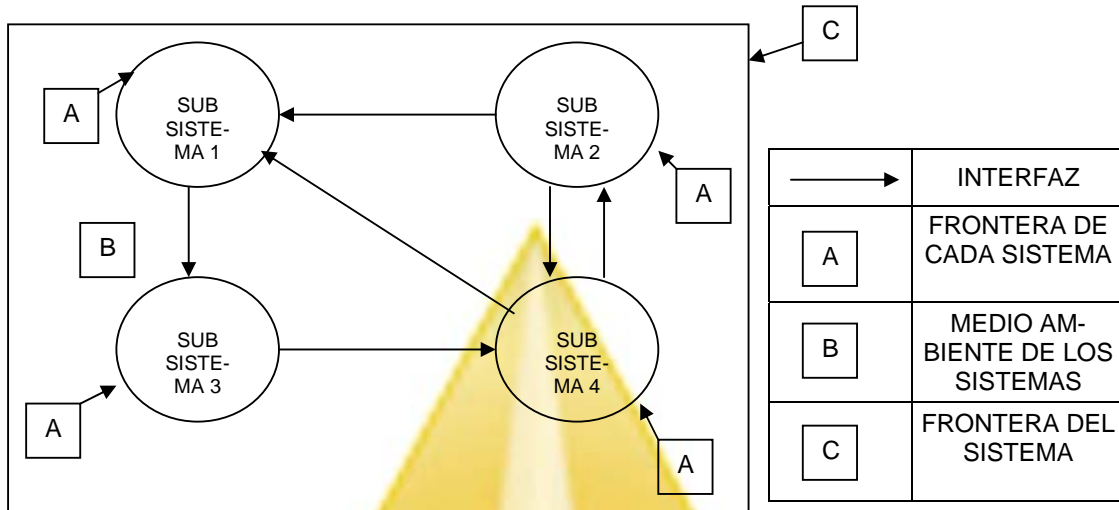
De acuerdo con su descomposición los sistemas se clasifican en: macrosistemas, mesosistemas y microsistemas



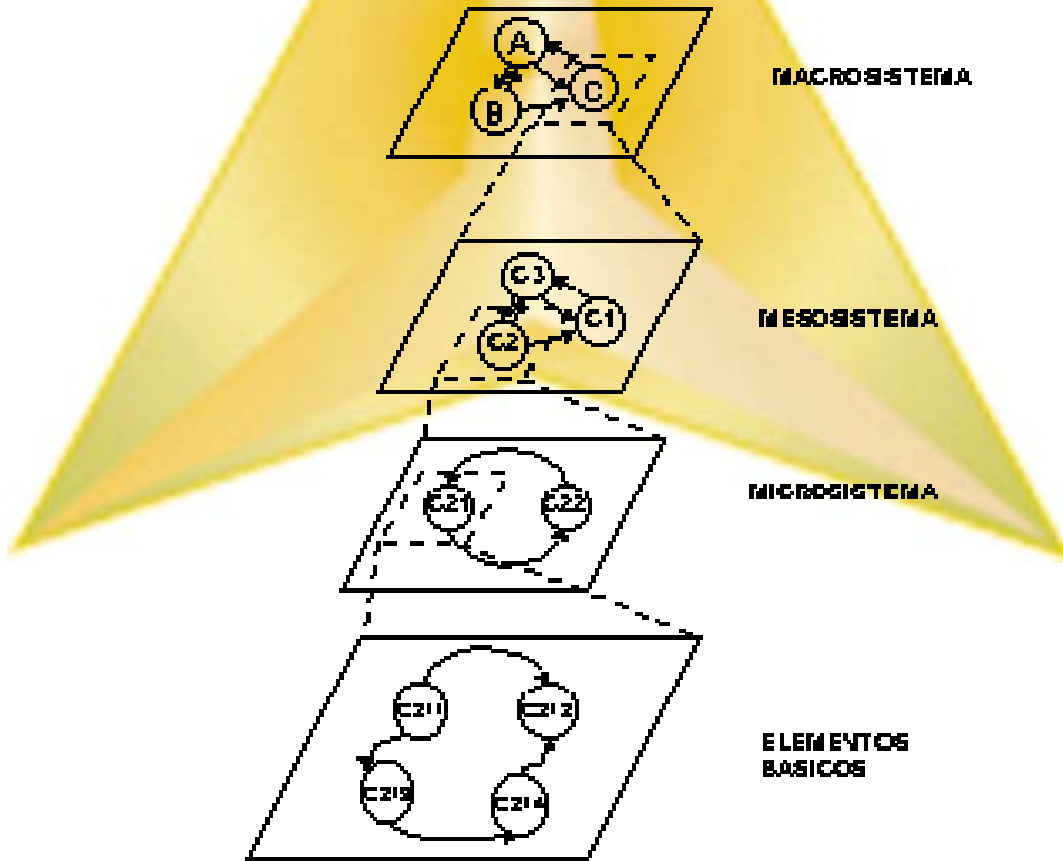
**MACROSISTEMAS:** Sistemas grandes que están compuestos por subsistemas menores.

**MESOSISTEMAS:** Sistemas intermedios que forman parte de los sistemas mayores y que están compuestos por sistemas menores.

**MICROSISTEMAS:** Sistemas pequeños que forman parte de los sistemas mayores y están formados por elementos básicos.



**Fig. 02** El Universo del Sistema



**Fig. 03** Jerarquía de los Sistemas

### 1.3 Elementos de un sistema

Todo sistema de información se compone de elementos y estos elementos pueden ser básicos que comprende los siguientes:

**ENTRADA.**- Se define como el elemento que recibe todas las entradas del sistema como pueden ser los datos que este procesará, mediante este elemento se puede decir que el indicador por el cual el sistema inicia su ciclo de procesamiento. La función de entrada caracteriza la fuerza de alimentación de datos o material al sistema para un determinado proceso. La entrada del sistema se encarga de capturar los datos elementales que el sistema necesita; de validar los datos del sistema, no permitiendo el ingreso de aquellos datos que el sistema no necesita.

**PROCESO:** El proceso se define como el elemento que posibilita la transformación de los datos, mediante las operaciones del sistema. El proceso se le puede también definir como aquel que cambia constantemente a través del tiempo. El proceso se puede realizar las siguientes operaciones; operaciones aritméticas (suma, resta, multiplicación y división), operaciones lógicas como son (Comparación, fusión, clasificación, intercalación, etc.).

**SALIDA:** El resultado de un proceso son las salidas. La salida puede definirse como el fin para el cual se unen los elementos, las características y las relaciona del sistema. Por lo tanto la salida es congruente con el objetivo, cuya definición es similar. En un sentido bastante real, la salida representa la finalidad, meta u objetivo para los cuales se organiza el sistema.

Los otros elementos que el sistema puede tener, los cuales interactúan entre el sistema y su medio ambiente son.

**AMBIENTE:** El ambiente es donde el sistema se desarrolla y interactúa, esta fuera del sistema y no controlado por él. El ambiente no sólo es una cosa que está fuera del control de los sistemas, sino también algo que determina en parte el funcionamiento del mismo.

**RETROALIMENTACION:** Se define como una función de un sistema la cual permite que el sistema vuelva hacer el ciclo de procesamiento tomando los datos del medio ambiente, para luego ser procesado y entregado al mismo. El proceso de retroalimentación consiste en que la información no utilizada con el medio ambiente, o sea, que nuevamente se toma la información del medio ambiente y luego esta es entrega al sistema en forma de datos elementales que el sistema lo utilizará.



Fig. 04 Elementos del Sistema

#### MODELO GENERAL DE UN SISTEMA

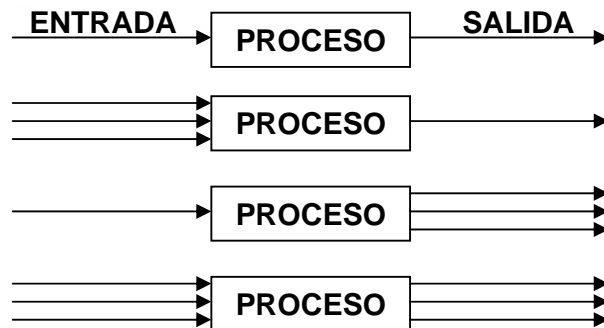


Fig. 05 Elementos del Sistema (Modelo General)



CAJA NEGRA



Fig. 06 Caja Negra

MODELO SIMPLIFICADO



Fig. 07 Elementos del Sistema (Modelo Simplificado)

### 1.4 Elementos de la Retroalimentación

Las partes de la retroalimentación son las que a continuación se presenta; estas partes miden el proceso de retroalimentación de los sistemas de información; ello se debe a los sensores, que permite determinar el tipo de información que deben ingresar a los sistemas.

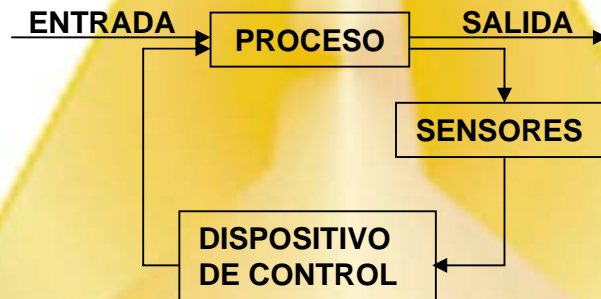


Fig. 08 Elementos de la Retroalimentación

**ACTIVIDADES DE LA RETROALIMENTACION:** La retroalimentación es un elemento del sistema el cual le permite volver a empezar; volver a empezar nuevamente con el procesamiento de los datos tomados de su medio ambiente, el cual el usuario los proporciona. Estas actividades son las siguientes: (figura 09)

### 1.5 Clasificación de los Sistemas y Enfoque de Sistema

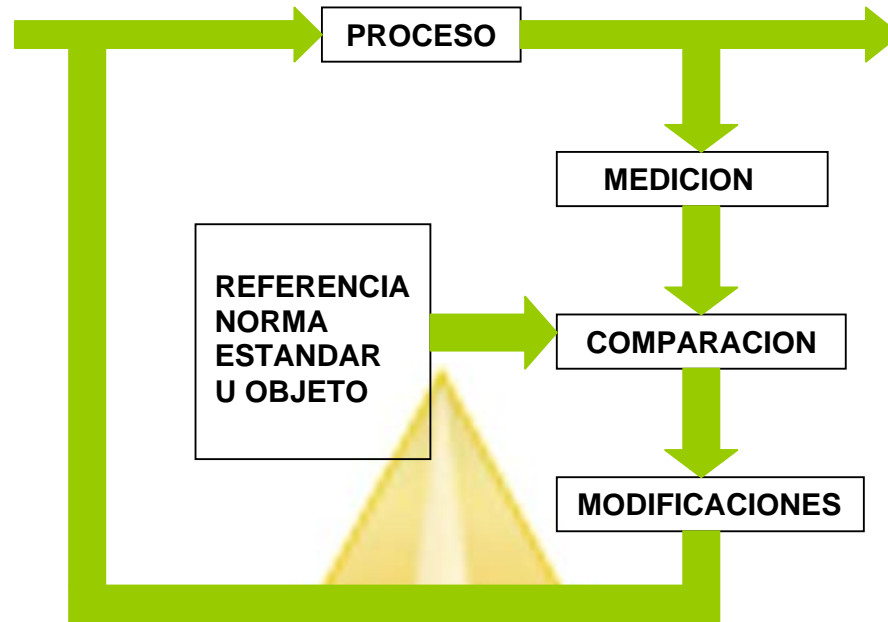
#### CLASIFICACION DE LOS SISTEMAS

Conocida la definición de sistema, podemos profundizar en su estructura general. Las siguientes categorías son importantes para al estudio de los sistemas de información, clasificaremos a los sistemas en dos categorías: **sistemas naturales y sistemas hechos por el hombre (artificiales).**

**SISTEMAS NATURALES,** Existen en la naturaleza y sirven a sus propios fines. Los sistemas naturales se dividen en dos subcategorías básicas: **sistemas físicos y sistemas vivientes.**

Los sistemas físicos incluyen ejemplos tan variados como:





**Fig. 09** Elementos de la Retroalimentación

- 1 Sistemas estelares: galaxias, sistema solar, etc.
- 2 Sistemas geológicos: ríos, cordilleras, volcanes, etc.
- 3 Sistemas moleculares: organizaciones complejas de átomos

Los sistemas vivos comprenden toda clase de animales y plantas que nos rodean al igual que a la raza humana.

**SISTEMAS HECHOS POR EL HOMBRE**, Son sistemas construidos, organizados y mantenidos por humanos, e incluyen:

**Sistemas sociales:** costumbres, doctrinas, legislación, etc.

**Una colección organizada y disciplinada de ideas:** codificación de libros para bibliotecas, la matemática, etc.

**Sistemas conceptuales:** Son construcciones simbólicas como la lógica, la matemática, la música. Una subclase de estos sistemas son los abstractos (ciencia) que fueron desarrollados por el hombre en su afán de comprender el mundo.

**Sistemas de transporte:** redes de carreteras, represas y canales, aerolíneas, ómnibus, barcos, etc.

**Sistemas de comunicación:** teléfono, télex, correo, etc.

**Sistemas de manufactura:** fábricas, líneas de ensamblado, etc.

**Sistemas financieros:** contabilidad, bolsa de valores, cuenta corrientes, etc.

Los sistemas naturales como los sistemas hechos por el hombre pueden ser: **abiertos o cerrados**

<b>SISTEMA ABIERTO</b>	Cuando se interrelaciona con su medio ambiente. Todo sistema abierto es capaz de responder a los estímulos provenientes de su medio ambiente de manera adecuada, produciendo cambios en su interior que le permiten seguir existiendo. Todo sistema que tiene entre sus componentes seres vivos es un sistema abierto.
<b>SISTEMA CERRADO</b>	Cuando no interactúa con su medio ambiente o cuando su medio ambiente no cambia. El sistema cerrado no responde a los estímulos del medio ambiente, como ocurre con una ecuación o un experimento de investigación.

Dentro de la clasificación de los sistemas hechos por el hombre hay sistemas adicionales que son de utilidad para los sistemas de información:



<b>SISTEMAS DUROS</b>	Son sistemas que para una misma entrada tienen asociada la misma salida, ejemplo de estos sistemas son los equipos de radio, televisión, la calculadora etc.
<b>SISTEMAS BLANDOS</b>	Son sistemas en que la salida no depende de la entrada; normalmente interviene el hombre, ejemplo de estos sistemas son el juego de ajedrez, una evaluación escrita, etc.
<b>SISTEMAS ADAPTATIVOS</b>	Es aquel que reacciona ante estímulos de su ambiente mediante cambios que mejoren su funcionamiento. Estos sistemas se caracterizan por poseer capacidades de aprendizaje de automodificarse y poseen una fuente propia de energía. Ejemplo de estos sistemas son el hombre, los animales, los negocios prósperos, etc.
<b>SISTEMAS NO ADAPTATIVOS</b>	Que no reaccionan oportunamente o simplemente no reacciona ante estímulos externos o cambios en el ambiente. Por ejemplo las computadoras no pueden conectarse por si mismas a una fuente de energía duradera, "aprender" a mortificarse y repararse sin intervención del hombre.
<b>SISTEMA PROBABILISTICO</b>	Se puede definir desde el punto de vista de comportamiento probable; hay cierto grado de error que siempre está asociado a la predicción de lo que hará este sistema. Un sistema de inventarlos es un ejemplo de sistema probabilístico. La demanda promedio y el tiempo promedio para el abastecimiento, etc., se pueden definir, pero el valor exacto en un momento dado no se conoce.
<b>SISTEMA HOMBRE MAQUINA</b>	Son sistemas en donde el hombre y la máquina desempeñan algunas actividades en el cumplimiento de una meta. Los elementos de las maquinas son relativamente cerrados y determinísticos, mientras que los elementos humanos del sistema son abiertos y probabilísticos. Son posibles varias combinaciones de hombres y máquinas. Los sistemas de información son un ejemplo de este tipo de sistemas.

### **ENFOQUE SISTEMICO**

Es un método de investigación, una forma de pensar, que enfatiza el sistema total, en vez de sistemas componentes, se esfuerza por optimizar la eficacia del sistema total. Ver gráfico en lugar de mejorarla eficacia de sistemas componentes. Como resultado del enfoque sistémico se produce el mejoramiento de los sistemas.

Es un proceso mediante el cual:

- (1) se identifican necesidades,
- (2) se seleccionan problemas,
- (3) se determinan los requisitos para la solución de problemas,
- (4) se escogen soluciones entre las alternativas,
- (5) se aplican métodos (metodología), medios (herramientas),
- (6) se evalúan los resultados y se efectúan las revisiones que requiera todo o parte del sistema, de modo que se eliminen las carencias y los errores.

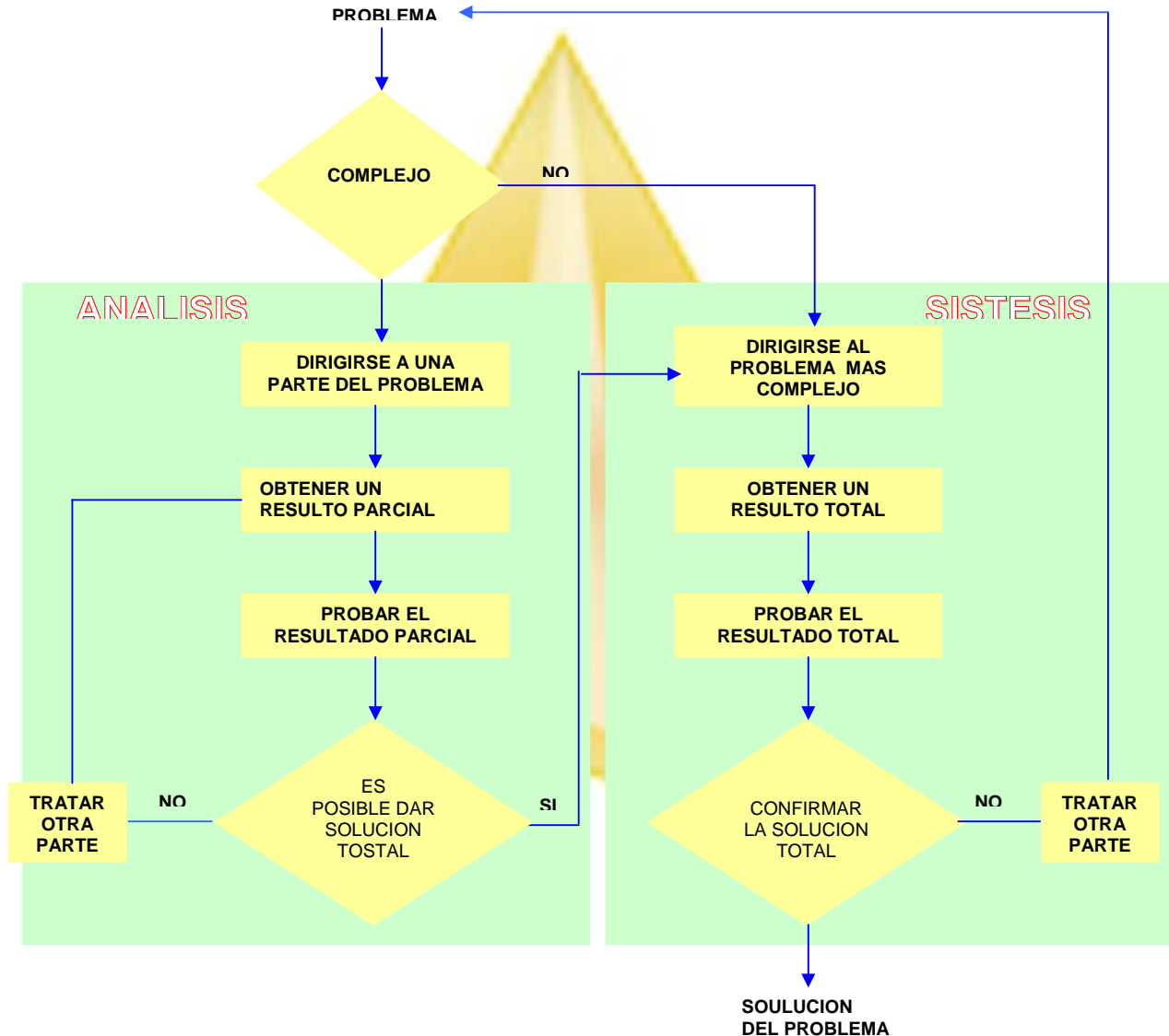
El enfoque de sistemas no sólo comprende la ejecución de una serie de pasos organizados, sino una técnica para analizar problema. Dado un problema complejo, el enfoque recomendado es subdividir el problema en subproblemas. Este proceso se conoce como la etapa de análisis. Cuando se han entendido todas las partes del problema de manera que se puede apuntar una solución, hay que volver a reunir aquéllas para encontrar una salida integral. El conjunto de soluciones de los subproblemas se llama la etapa de **SINTESIS**.

En la figura que se presenta (Fig. 10); aparece un diagrama de los pasos lógicos al subdividir un problema complejo en fracciones menores y resolverlas después. Los pasos del 1 al 5 describen el proceso analítico. Se toma una subparte en el paso 1 obteniendo el resulta-



do en el paso 2. Este resultado se prueba en el paso 3, si como consecuencia de la prueba es posible la solución para el problema complejo, el proceso continua en los pasos del 6 al 10, donde se juntan las subpartes (etapa de síntesis) en una solución total. Si los resultados de la prueba en el paso 3 indican que no hay todavía una solución total, se trata con otra parte del problema en el paso 5 repitiendo del paso 2 al 4. Se continúa este ciclo tomando las partes del problema hasta encontrar una solución integral.

Si el problema que afronta el analista es sencillo, no es necesario subdividido en partes, sino tomarlo entero. Si los resultados de la prueba de la solución general en el paso 8 no confirman haber llegado a una salida aceptable, sería aconsejable volver a plantear el problema. Este nuevo planeamiento puede llevar a un análisis de las partes. Cuando se ratifique la solución total en el paso 9, el problema ha sido resuelto.



**Fig. 10** Enfoque Sistémico



## 2.0 CICLO DE VIDA DEL DESARROLLO DE SISTEMAS

El desarrollo de sistemas, un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un sistema de la empresa que necesita mejoras.

El método del ciclo de vida para desarrollo de sistemas (SDLC) es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. Esta sección examina cada una de las seis actividades que constituyen el ciclo de vida de desarrollo de sistemas. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se siguen para, efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases de desarrollo; algunos componentes en la fase de análisis mientras que otros en etapas avanzadas de diseño.

El método del ciclo de vida- para desarrollo de sistemas constan de Siguietes actividades:

1. investigación preliminar
2. Determinación de los requerimientos del sistema
3. Desarrollo del sistema prototipo
4. Diseño del sistema
1. Desarrollo de software
2. Prueba de los sistemas
3. Puesta en marcha

Los sistemas de información no se crean por generación espontánea, se sigue una metodología para su construcción, llamado 'ciclo de vida de un sistema de información'. Una metodología es un sistema ordenado de proceder para la obtención de un fin. Adoptada la metodología están también adoptadas las herramientas que vamos a utilizar. Debemos entender por herramienta a cualquier dispositivo que, cuando se emplea en forma apropiada, mejora la realización de una tarea.

Los sistemas por más simples o complejos, tienen las siguientes etapas:

Etapa de nacimiento  
Etapa de desarrollo  
Etapa de producción  
Etapa de muerte

Para ciclos de vida en cascada o tradicional y basado en prototipos. La diferencia sustancial entre ambos modelos de ciclo de vida, radica en el tiempo invertido para el desarrollo de un sistema de información; los modelos basados en prototipos permiten desarrollar sistemas de información en menos tiempo con ayuda de herramientas muy potentes como son los lenguajes de cuarta generación (4GL) y las herramientas CASE.

Un prototipo es un sistema que funciona, muestra la función central de un sistema que se está modelando, está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impresa o presentada en pantalla. Es la primera versión de un sistema de información.

El ciclo de vida sirve para:

- Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas.
- Proporciona los puntos de control y revisión, para que los niveles directivos de una organización controlen el proyecto.
- Guía el desarrollo de todos. Los proyectos de sistemas de información en una organización (no es razonable que cada proyecto tenga diferente metodología de desarrollo).

Por ciclo de vida de un sistema de información se entiende al conjunto de fases por las que pasa a lo largo del tiempo, desde la fase de estudio y concepción hasta la realización, explotación y mantenimiento.

Las fases o etapas de este ciclo de vida son definidas por las diversas metodologías y hay diferentes variantes de las mismas. En general, todas las metodologías coinciden en una secuencia lógica de fases.

## 2.1 Ciclo de Vida en Cascada, Fases

El ciclo de vida en cascada se basa en la elaboración de un sistema basado en etapas o fases de desarrollo, donde cada etapa es la culminación de la anterior y en donde no existe un retorno a la fase anterior. Culminado una fase el analista procede con la siguiente fase del desarrollo del sistema de información hasta la implementación del sistema en la organización que lo necesita.

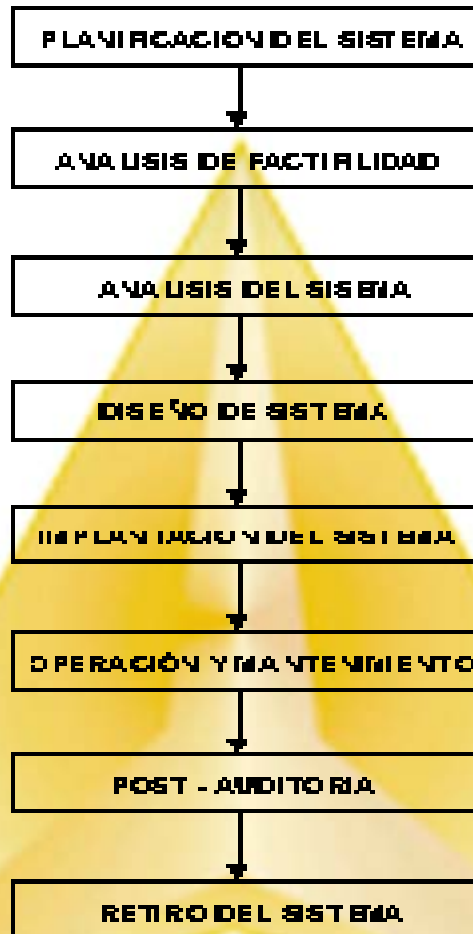


Fig. 11 Ciclo de Vida en Cascada

### FASES DEL CICLO DE VIDA EN CASCADA

FASE	PASO	ACTIVIDAD
1 Planificación del Sistema.	Percepción de necesidades	Clarificar el problema, identificar necesidades e iniciar estudios.
	Plan de desarrollo de sistemas de información	Determina el desarrollo del sistema con el plan estratégico de la organización
2 Análisis de factibilidad	Evalúa el sistema existente. Propone y Evaluar sistemas alternativos	Determina su adecuación. Determina factibilidad técnica, operacional y economía de las alternativas.



3 Análisis del sistema.	Analiza el sistema existente y la organización.	Determina la fortaleza y debilidades del sistema existente y documentar actividades y entidades de la organización
	Determina requerimientos de información	Establece necesidades de información de los usuarios e identificar entidades de datos y atributos.
4. Diseño del sistema	Diseño de entrada y salida.	Diseña formatos de datos, formatos de pantallas de entrada y salida y reportes.
	Diseño de procedimientos y lógica de procesamiento.	Especifica formatos de datos, mecanismos de acceso y características de la base de datos.
	Especifica la estructura de la base de datos.	Identifica formatos de datos, mecanismos de acceso y características de la base de datos.
	Genera plan de conversión.	Especifica procedimientos e itinerarios para la conversión al nuevo sistema.
	Ratificación final de los usuarios	Congela las especificaciones y el diseño.
5 Implementación	Desarrolla programas y procedimientos	Pruebas de código, documentación de programa, desarrollar instrucciones para los usuarios y manuales de procedimientos
	Crea o convertir la base de datos	Crear la nueva base de datos requerida por el sistema y convertir el registro existente de datos al formato de base de datos.
	Proba la aceptación de sistema	Estimular su uso y determinar si responde a los requerimientos
	Entrenamiento de los usuarios	Desarrolla materiales para el entrenamiento y conducir el entrenamiento.
	Conduce un curso paralelo.	Hacer funcionar paralelamente al sistema nuevo con el sistema antiguo, hasta que el nuevo sistema este listo para operar independientemente.
6 Operación y mantenimiento	Opera el sistema	Hacer funcionar el nuevo sistema como sistema de producción de la organización
	Modifica el sistema	Mantener el sistema corrigiendo los errores y haciendo las modificaciones correspondientes
7 Post Autoría	Revisa los costos y beneficios	Revisar periódicamente el rendimiento y las metas de la organización para determinar donde hay que hacer cambios en el sistema.
8 Retiro	Abandono o reemplazo del sistema antiguo por un nuevo sistema	Indica el desarrollo de un nuevo sistema vía ciclo de vida de desarrollo de sistema.

## 2.2 Ciclo de Vida basado en Prototipos

Este ciclo se basa para sistema ya existentes los cuales serán modificados o para prototipo de sistemas que se basan en sistemas anteriores, en donde el analista realiza una planificación del nuevo sistema basándose en sistemas que tiene la misma estructura para ello realizan análisis previo del nuevo sistema, evalúan, analizan, diseñan y luego implementa.

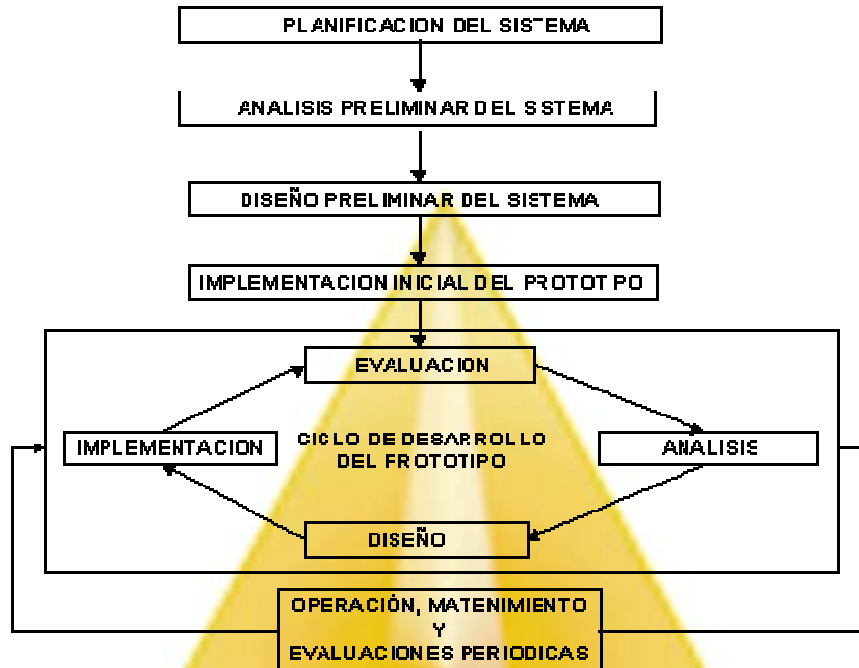


Fig. 12 Ciclo de Vida Basado en Prototipos

### FASES DEL CICLO DE VIDA BASADO EN PROTOINPOS

FASE	PASO	ACTIVIDAD
1. Planificación del sistema	Percepción de necesidades	Clarificar problemas, identificar necesidades e iniciar estudios.
	Elabora el plan de desarrollo de sistemas de información.	Relaciona el desarrollo del sistema con las metas organizacionales vía inclusión de los directivos en el proceso de planificación.
	Análisis de factibilidad	Realiza estudios preliminares de factibilidad para evaluar la adecuación del sistema existente.
2. Análisis preliminar del sistema	Análisis del sistema existente	Determina la fortaleza y debilidades del sistema existente.
	Determina requerimientos de información	Establece una comprensión preliminar, de las necesidades de información de los usuarios e identificar las entidades y atributos iniciales de datos.
	Establece las especificaciones del prototipo inicial	Generar especificaciones para el prototipo inicial, incluyendo salidas y componentes de procesamiento.



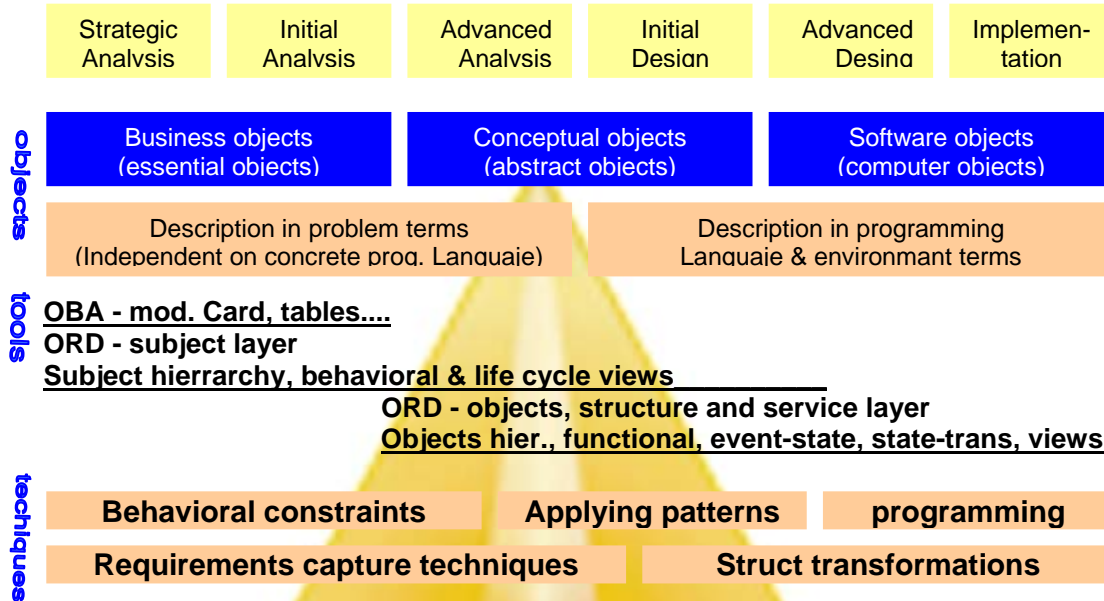


3. Diseño preliminar del sistema	Diseño de lógica de procesamiento, entradas/ salidas y manual de procedimientos para el prototipo inicial	Involucra a los usuarios como participantes activos en el diseño de formatos impresos, pantallas y procedimientos automatizados y manuales.
	Especifica la estructura de la base de datos para el prototipo inicial	Diseña la estructura de la base de datos inicial y mecanismos preliminares de acceso de datos.
4. Implantación inicial del prototipo	Diseña procedimientos y programas de prototipos	Codificar programas con lenguajes de cuarta generación, prueba de la documentación de los programas prototipo, especificar instrucciones iniciales y procedimientos manuales.
	Construye la base de datos prototipo	Usar un manejador de base de datos para establecer la base de datos inicial con datos reales.
	Entrega el prototipo inicial y entrenar a los usuarios.	Hace accesible el prototipo a los usuarios con documentación preliminar y seminarios de entrenamiento.
5. Ciclo de vida del prototipo	Evaluación	Estimular a los usuarios para que trabajen con el nuevo prototipo y detectar deficiencias o necesidades de ampliación o modificaciones, dejar el ciclo si el sistema es satisfactorio y completo.
	Análisis	Análisis del sistema para desarrollar un nuevo prototipo; especifica nuevos requerimientos, modificaciones y ampliaciones del prototipo sobre la base de la experiencia de los usuarios.
	Diseño	Revisa los formatos de entradas y salidas, lógicas de procesamiento, estructura de la base de datos y procedimientos manuales para el nuevo prototipo
	Implantación	Modificar el código existente, cambiar la documentación y procedimientos manuales existentes; y se entrega el nuevo prototipo a los usuarios.
6. Operación y mantenimiento	Opera el sistema	Hacer funcionar el último prototipo como sistema de producción.
	Evaluaciones perdidas	Evaluar periódicamente el prototipo operando para determinar cuando debe reiniciarse un ciclo de prototipo

### 2.3 Ciclo de Vida del Sistema Orientado a Objetos

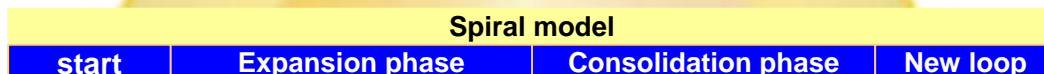
El ciclo de vida en sistemas orientados a objetos se basa en un análisis inicial, luego en un análisis avanzado, para luego diseñar el objeto, y esto se realiza con cada uno de los objetos del sistema; para ello se emplean técnicas y herramientas para su modelado de objetos.

#### BORM CONCLUSION - PROJECT LIFE CYCLE

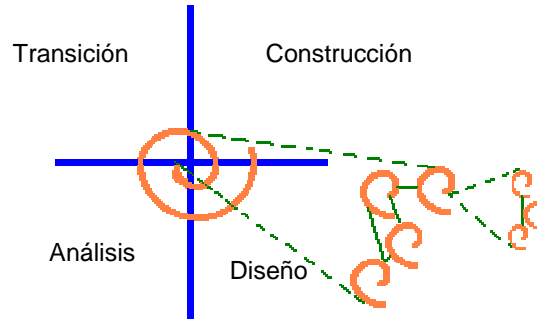


### 2.4 Ciclo de Vida en Espiral

El ciclo de vida en espiral consiste en iniciar el estudio del objeto con un análisis inicial, luego expandir la fase con un análisis avanzado; luego consolida la fase del análisis del objeto con un diseño inicial; luego se pasa a un diseño avanzado de objeto; para luego repetir nuevamente los pasos con otro objeto o el mismo objeto hasta conseguir los resultados esperados con el sistema.



En términos del ciclo de vida del desarrollo de sistemas (SDLC), la ingeniería concurrente (concurrent engineering, CE) es un método sistemático para el desarrollo integrado y concurrente de sistemas automatizados. Primero, esto significa que la ce implica preparar los diversos componentes de un sistema de manera interoperable. Por ejemplo, la figura siguiente ilustra que se puede poner en práctica una sola espiral de desarrollo en vez de varias. A fin de que, dentro de un sistema, estos componentes se comuniquen entre sí, los equipos de proyectos también deberán comunicarse. En un entorno de ce, es importante la comunicación directa entre proyectos. Sin embargo, si demasiada comunicación causa retraso en los proyectos, tendrá que reducir el número de proyectos interrelacionados. En la ce, se espera que cada espiral sea un componente por derecho propio, pero aun así no deberá desarrollar en su propio espacio vacío.



Procedimiento en espiral para la metodología de desarrollo que emplea desarrollo de sistemas concurrentes

**Fig. 13** Ciclo de Vida en Espiral

### 3.0 HERRAMIENTAS Y PROCEDIMIENTOS PARA DESARROLLAR UN SISTEMA

Los sistemas de información no se crean espontáneamente, se siguen métodos para su construcción y se emplean herramientas que facilitan su elaboración. El método es un sistema ordenado de procesar para la obtención de un fin. Determinado el método es también definir la herramienta a ser utilizada. La herramienta es cualquier dispositivo que cuando se emplea en forma adecuada, mejora la realización de una tarea.

#### 3.1 La Investigación Preliminar

Se puede iniciar una petición por muchas razones, pero la clave es que alguien, ya sea la alta dirección, un empleado o un especialista de sistema, inicie un requerimiento para recibir ayuda de un sistema. Cuando ese requerimiento se realiza, la primera actividad de sistema; es decir, la investigación preliminar, se inicia. Esta actividad tiene las siguientes partes:

- Clasificación del requerimiento
- Estudio de factibilidad
- Aprobación del requerimiento
- El cuestionario
- Recopilación de los datos

#### 3.2 El Plan de Entrevista

Una forma de poder recolectar la información que requiere todo el sistema de información que se emplean en una organización; es un usando un método llamada "Plan de Entrevistas"; esta técnica consiste en programar todas las entrevistas que el analista de sistemas tiene que realizar con todas las personas involucradas en el problema de manejo de información de las áreas de la organización afectada. Este método consiste en determinar el horario y el día de la semana más apropiado para realizar la entrevista con la persona que nos dará la información que requerimos; la entrevista no debe durar demasiado y se debe tener todas las preguntas listas o mejor dicho el cuestionario de preguntas previamente preparadas; para que la entrevista tenga el éxito esperado, el entrevistador debe ser una persona de aspecto agradable, edu-



cado y que tenga un nivel de conocimiento para poder absolver cualquier duda que se presente con las preguntas en la entrevista. Luego se tomará nota o grabará la entrevista, para luego ser procesada las respuestas del entrevistado.

### **3.3 El Cuestionario - Partes y Tipos de preguntas**

Es un documento que se emplea en la determinación de los requerimientos del proyecto de sistema para la recopilación de la información, que es aplicada a grandes grupos que no se pueden entrevistar en forma individual. El cuestionario debe recoger la problemática en estudio como objetivo principal, los procedimientos y procesos que se realizan en las áreas en estudio.

Las características del cuestionario son las siguientes:

- (1) Las preguntas pueden ser: Abiertas o Cerradas.
- (2) Deben ser entendidos y comprendido por cualquier persona inclusive por un niño de ocho años. No se emplea términos muy técnicos.
- (3) Debe reflejar en las respuestas los objetivos que se persigue en el estudio del proyecto de sistema.
- (4) No debe ser muy extenso o mezclar varios estudios en uno solo.
- (5) Las alternativas de las respuestas deben ser claras y precisas.

#### **Partes del Cuestionario**

- (1) Logotipo de la Institución que esta realizando el estudio del proyecto de sistemas.
- (2) El título del estudio del proyecto de sistema.
- (3) Introducción; que explique de se trata el estudio del proyecto de sistema que esta realizando.
- (4) El cuerpo del cuestionario de preguntas sobre el proyecto de sistemas.

### **3.4 Recopilación de los datos**

La recopilación de los datos, que es el objetivo de un estudio de levantamiento de información para un sistema, se puede llevar a cabo por medio de las encuestas, que son realizada a los usuarios del nuevo proyecto de sistema; usuario colaboradores y que participan en la elaboración del nuevo sistema. Otra forma de recopilar información para el sistema, es por medio de las entrevistas que se realizan con los usuarios del nuevo sistema; esta entrevista se realiza en forma programada de acuerdo al plan de entrevistas. Los manuales y reglamento de la organización pueden ayudar mucho al analista en la recopilación de los datos que requiere el nuevo sistema, ya que estos documentos tienen toda la información del funcionamiento de la organización y como se lleva a cabo todos los procesos, actividades y tareas de las diferentes áreas que tiene la organización. Otra forma de obtener los datos necesarios para el nuevo sistema, es mediante la observación que realiza el analista de sistema de los procesos que se realizan en las diferentes áreas de la organización sobre todo donde esta el problema que el nuevo sistema deberá resolver.



## 4.0 LOS OBJETOS DEL SISTEMA

Las técnicas orientadas a objetos mejoran la capacidad del profesional de la computación en diversos y sorprendente modo. La metodología tradicional para el desarrollo de sistemas de información, si, son, por naturaleza de carácter general. Como tales, contienen un conjunto ideas de métodos, técnicas y vías de acción que en realidad nunca pueden ser seguidas al pie de la letra. Se deben afirmarse en relación con la situación que se tiene delante. A veces se omiten algunos pasos, se modifican o se añaden otros. Las guías de acción a menudo se cambian o se ignoran para adecuarse a circunstancias especiales, como las capacidades tecnológicas y de desarrollo, la aplicación de factores eventos.

### 4.1 Definición de un objeto

Las personas nos formamos conceptos desde temprana edad. Cada concepto es una idea particular o una comprensión de nuestro mundo. Los conceptos adquiridos nos permiten sentir y razonar acerca de las cosas en el mundo. A estas cosas a las que se aplican nuestros conceptos se llaman *objetos*. Un objeto podría ser real o abstracto, como los ejemplos siguientes:

- Una factura
- Una organización
- Una figura en un programa de dibujo (como MacDraw)
- Una pantalla con la que interactúa un usuario
- Un campo o nodo de la pantalla de una herramienta CASE
- Un mecanismo en un dispositivo de robótica todo un plano de ingeniería
- Un componente de un plano de ingeniería
- Un texto y fotografías utilizados en el plano de un periódico
- Un avión
- El vuelo de un avión
- Una reservación aérea
- Un icono en la pantalla a la que un usuario puede apuntar y "abrir"
- Un proceso para llenar un pedido
- El proceso para escribir esta línea

En el análisis y diseño orientados a objetos (OO), nos interesa el comportamiento del objeto. Si construimos software, los módulos de software OO se basan en los tipos de objetos. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento. Las operaciones se codifican como *métodos*. La representación en software OO del objeto es entonces una colección de *tipos de datos y métodos*. En software OO. Un objeto es cualquier cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos.

Un objeto puede estar compuesto por otros objetos. Estos últimos, a su vez, pueden estar compuestos de objetos, del mismo modo que una máquina está formada por partes y éstas, también, están formadas por otras partes. Esta estructura intrincada de los objetos permite definir objetos muy complejos.

A menudo se describe la orientación a objetos en términos de estructura y comportamiento. La palabra estructura es una metáfora visual, espacial, que alude a una visión estática de cómo están dispuestos los objetos en el espacio. La estructura puede especificar varias configuraciones de objetos como empleados, documentos y diseños de ingeniería. En contraste, el comportamiento se refiere a cómo se modifica nuestro mundo a lo largo del tiempo. Por ejemplo, el comportamiento, puede contratar un empleado o indicarnos que el empleado ha llegado a su edad de retiro. En suma, describe los procesos que consultan o modifican a los objetos.



## 4.2 Tipos de Objetos - Objetos - Clases

Los conceptos que poseemos se aplican a tipos determinados de objetos. Por ejemplo, empleado se aplica a los objetos que son personas empleadas por alguna organización. Algunas instancias de *empleado* podrían ser Juan Pérez, María Sánchez, etcétera. En el análisis orientado a objetos, estos conceptos se llaman tipos de objetos; las instancias se llaman objetos.

*Un tipo de objeto es una categoría de objeto.*  
*Un objeto es una instancia de un tipo de objeto.*

Nuestra realidad está formada por los tipos de objetos (es decir, conceptos) que poseemos y por los *objetos* a los que se aplican nuestros conceptos. Un objeto es cualquier cosa a la que es aplicable un tipo de objeto (es decir, un concepto). Es un caso particular de un tipo de objeto.

Los términos objeto e instancia pueden ser utilizados en forma indistinta. Un ejemplo sencillo de un objeto es la pluma sobre la mesa de su tía, esa instancia específica, individual de una pluma. Otros objetos podrían incluir la ciudad en la que vive, usted mismo, su ocupación, cierto proceso, un evento en particular, un momento en el tiempo, un registro en una base de datos, una porción de datos, un sonido, una imagen, una señal óptica, el número 42, un pulso magnético, un documento, un vector, una matriz o mi perro Millicent. Por favor observe que no todos los objetos son datos, aun cuando todos los datos son objetos. En el análisis orientado a objetos se investiga objetos sin perjuicio del hecho de que se vayan o no a convertir en datos. De esta forma, analizamos la comprensión humana antes de tratar con bits, bytes, campos y registros.

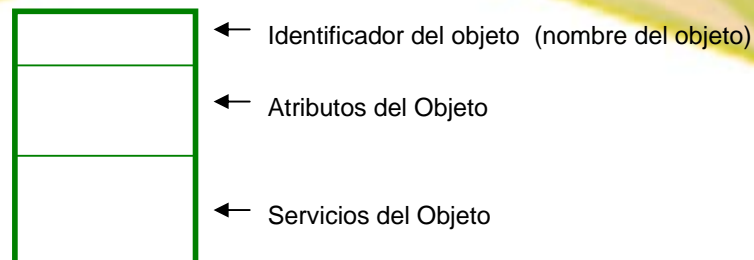
## 4.3 Elementos de los Objetos

Los elementos de un objeto; son aquellos que describen la composición de un objeto, como también sus partes que las conforman; estos de los objetos son las siguientes:

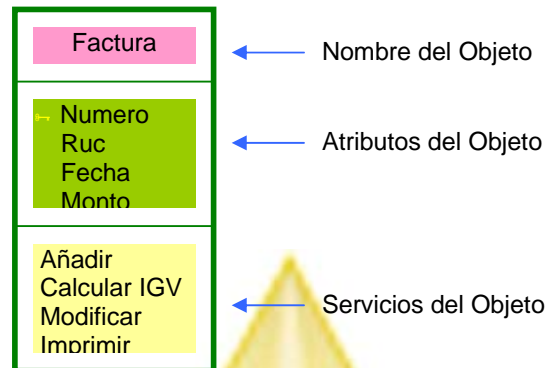
**Identificador del objeto:** El identificador del objeto es el nombre con se conoce y se identifica un objeto dentro de todos los objetos que existen en el mundo real. También permite identificar y describir al tipo de objeto de su categoría que pertenece.

**Atributos del Objeto:** Los atributos del objeto, son los datos que pertenecen al objeto, y de esta manera ea como se conoce objeto, sus características y sus valores del objeto. Los atributos describen la forma del objeto, como este es dentro del mundo real; estos atributos pueden subdividirse en más atributos elementales que puedan describir mejor al objeto.

**Servicios del Objeto:** Los servicios del objeto, son los procesos o procedimientos programados, que determina el comportamiento del objeto, los que el objeto puede hacer y hace con sus datos que posee. Un servicio, es un conjunto de proceso que el objeto hará de acuerdo al estado en que se encuentra y el comportamiento que determina que el objeto cambie de estado y realice y servicio.



**Fig. 14** Partes del Objeto



**Fig. 15** El objeto, sus partes, atributos y servicios

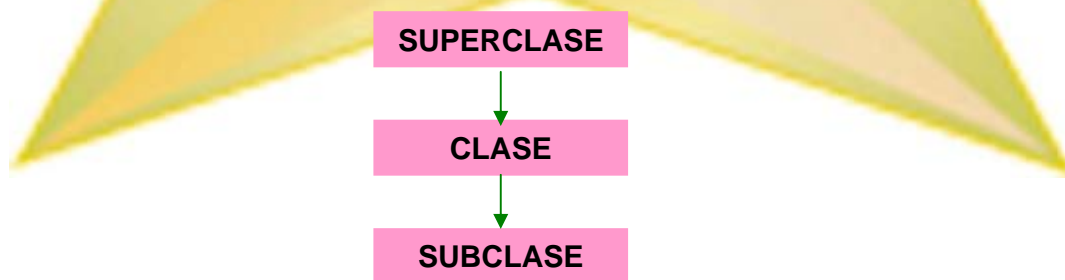
### 4.3 Jerarquía de los Objetos

Los objetos se subdividen en jerarquías que determinan el grado de relación que existen entre los objetos y su dominio sobre uno y otro objeto. Esta jerarquía determina el grado de mando que tiene los objetos dentro de un sistema. El objeto que tiene el más alto grado de jerarquía se le conoce con el nombre de objeto Padre o también con el nombre de objeto Superior; mientras que los objetos de rangos inferiores son conocidos como objetos "Hijo", también se les conocen como objetos derivados del objeto superior. Los objetos derivados se les conocen así porque son derivados de un objeto superior y, además, los atributos que tiene este objeto son atributos heredados del objeto superior. Los objetos también se les puede definir como clases y aquellos derivados como una subclase. El grado de jerarquía que tiene los objetos son los siguientes:

**Superclase:** Son las clases de mayor jerarquía que existe dentro de los objetos de un sistema. Las superclases no se presenta principalmente dentro de un conjunto de objetos.

**Clases:** Son objetos que se clasifican como clases de objetos, los cuales pertenecen a un tipo de categoría de objeto y representa a los objetos de una clase dentro del sistema.

**Subclase:** Son los objetos derivados de la clase, estos objetos reciben como herencia los atributos que tiene los objetos clase.

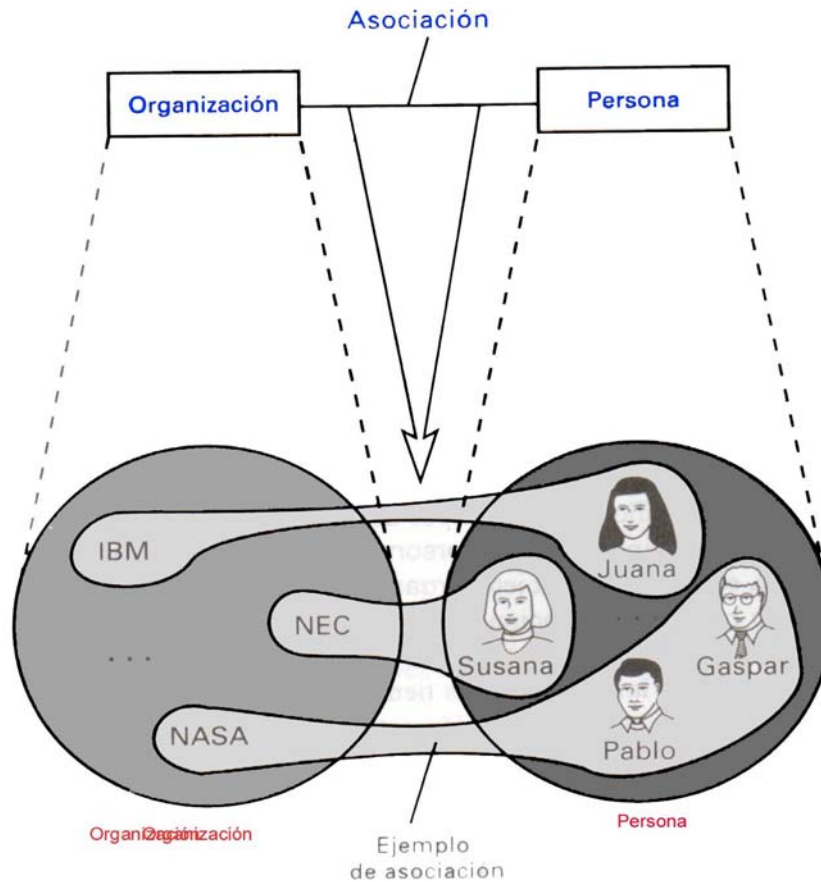


**Fig. 16** Jerarquía de las Clases

### 4.5 Características de los Objetos

Los objetos del mundo real se presentan con características que los identifican uno de los otros, y así determinar la naturaleza de los objetos. Estas características permiten describir las acciones que se ejercen sobre los objetos de un sistema; esta característica es:

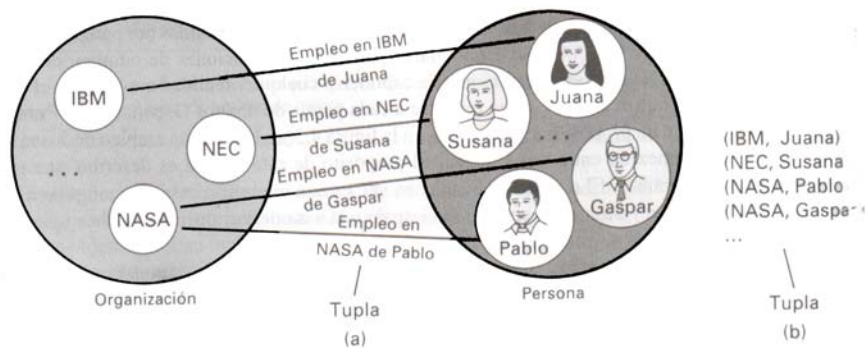
**Asociación:** La asociación proporciona una manera de vincular objetos de varios tipos de una manera significativa. Las asociaciones entre los objetos están representados por líneas, esta representación una conexión entre los tipos de objetos; la utilización de las conexiones nos



**Fig. 17** Asociación de Objetos

permite mapear los objetos de un conjunto a objetos de otro; también podemos decir que una asociación es un tipo de relación entre los objetos. Los tipos de relación y los mapeos son dos técnicas para describir las asociaciones entre los objetos.

**Relaciones:** La relación proporciona una manera de establecer un vínculo entre una instancia de un objeto y la instancia del otro objeto; como decir establecer una conexión entre parejas de objetos que se vincula mediante una asociación. Las relaciones se pueden establecer entre más de dos objetos; y así se tendrá una relación ternaria de objetos; entre cuatro objetos una relación de cuartetos, y así sucesivamente.



**Fig. 18** Relación de Objetos



**Mapeos:** Las relaciones nos permiten tratar las asociaciones como entidades relacionadas. Los mapeos, sin embargo, no ven en su totalidad el conjunto. Nos permiten, es cambio, iniciar desde simplemente una parte del todo y atravesar (o más bien mapear) hasta llegar a la otra parte de ese todo. El tipo de relación, como concepto y término, es de uso común. La noción de mapeo entre tipos de objetos también es común. Sin embargo, el mapeo determina el concepto de relación que existe entre los objetos.

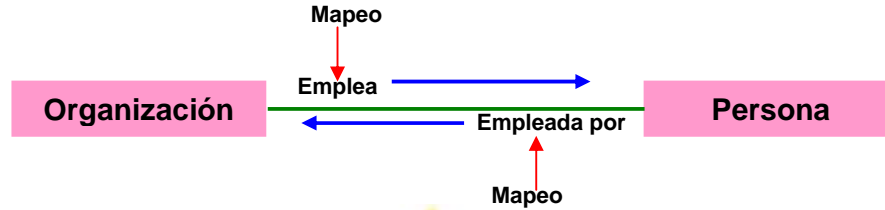


Fig. 18 Mapeo entre objetos

**Encapsulamiento:** El empaque de conjunto de datos y métodos se llama encapsulado. El objeto esconde sus datos de los demás objetos y permite el acceso a los datos mediante sus propios métodos. Esto recibe el nombre de ocultación de la información. El encapsulado evita la corrupción de los datos de un objeto.

El encapsulado oculta los detalles de su implantación interna a los usuarios de un objeto. Los usuarios se dan cuenta de las operaciones que pueden solicitar del objeto pero desconocen los detalles de cómo se lleva a cabo la operación. Todos los detalles específicos de los datos del objeto y la codificación de sus operaciones están fuera del alcance del usuario. Encapsulado es el resultado (o acto) de ocultar los detalles de implantación de un objeto respecto de su usuario.



Fig. 19 Composición del Objeto

**Mensajes:** Para que un objeto haga algo, le enviamos una solicitud. Esta hace que se produzca una operación. La operación ejecuta el método apropiado y, de manera opcional, produce una respuesta. El mensaje que constituye la solicitud contiene el nombre del objeto, el nombre de una operación y, a veces, un grupo de parámetros.

La programación orientada a objetos es una forma de diseño modular en la que, con frecuencia, el mundo se piensa en términos de objetos, operaciones, métodos y mensajes que se transfieren entre tales objetos. Un mensaje es una *solicitud* para que se lleve a cabo la *operación* indicada y se produzca el resultado. En consecuencia, las implantaciones OO se refieren a los mensajes como solicitudes.

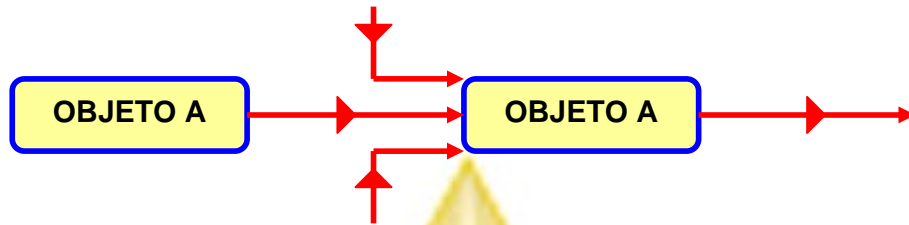
Una *solicitud* invoca una operación específica, con uno o más objetos como parámetros. Los objetos pueden ser muy complejos, puesto que pueden contener muchos subobjetos, éstos a su vez pueden contener subobjetos, etc. La persona que utilice el objeto no tiene que conocer su complejidad interna, sino la forma de comunicarse con él y la forma en que responde.

Por ejemplo, usted se puede comunicar con la videocasetera al enviarle *solicitudes* por medio de un sintonizador de control remoto. Responde el aparato mediante determinada acción y presenta las respuestas en la pantalla. Todos los objetos del tipo Sony AH-8500 se controlan



mediante el mismo tipo de *interfaz*. Las solicitudes del control remoto no se comunicarán con una JVC HR-S6600 VCR, puesto que ésta necesita la interfaz distinta.

El usuario de una oficina puede "hacer clic" en un icono que representa al objeto "directorio". El objeto responde al desplegar en la pantalla una lista de nombres. El usuario puede utilizar distintas operaciones para la interacción con esta lista (pasar a una parte de la lista y apuntar hacia el nombre de cierta persona). El software resalta entonces el nombre.



**Fig. 20** Mensajes entre objetos

**Blob:** Las computadoras actuales almacenan grandes cadenas de bits que representan imágenes, diagramas, enunciados o tal vez música o vídeo. Cada uno de esos objetos es conocido como BLOB (siglas en inglés de Binary Large Object, o sea, objeto binario de gran tamaño). Los BLOBs tienen métodos que permiten mostrarlos o utilizarlos. Por ejemplo, se utilizan técnicas de compactación de modo que las imágenes se puedan almacenar en un número menor de bits. Los BLOBs disponen de métodos que permiten mostrarlos en pantallas distintas, Pueden tener métodos de seguridad, como el cifrado o descifrado, En el procesamiento de imágenes, éstas se pueden analizar de manera rápida con baja resolución y lentamente con resolución media, y presentarse después con alta resolución.

La creciente aceptación del procesamiento de imágenes aumentará la importancia de las bases de datos orientadas a objetos que puedan administrar los BLOBs de manera eficiente. Enviaremos solicitudes a los BLOBs para indicarles que se exhiban a sí mismos, se cifren a sí mismos, se ligen para su edición, etc. En una base de datos *OO*, *factura* se podría mantener en una forma gráfica hecha a mano junto con las instrucciones orales del cliente, así como un registro alfanumérico. La imagen gráfica es un objeto; las instrucciones orales son otro objeto; el registro alfanumérico es otro. Juntos forman un objeto único *factura*.

**Herencia:** Un tipo de objeto de alto nivel puede especializarse en tipos de objeto de bajo nivel. Un tipo de objeto puede tener subtipos. Por ejemplo, el tipo de objeto *persona* puede tener subtipos *civil* y *militar*. *Militar* puede tener subtipos *oficial* y *enrolado*. *Oficial* puede tener subtipos *teniente*, *capitán* y *mayor*, y también subtipos como *marino* o *rapador*, *oficial en servicio activo* u *oficial retirado*. Existe una jerarquía de tipos, subtipos, subsubtipos, etcétera.

Una clase implanta el tipo de objeto. Una subclase *hereda* propiedades de su clase padre; una sub-subclase hereda propiedades de las subclases; etc. Una subclase puede *heredar* la estructura de datos y los métodos, o algunos de los métodos, de su superclase. También tiene sus métodos e incluso tipos de datos propios. La subclase tiene los mismos métodos que su superclase, pero tiene también el método *G*. A veces, una clase hereda propiedades de más de una superclase. Esto recibe el nombre de *herencia múltiple*





## 5.0 EL MODELO CONCEPTUAL DEL SISTEMA

### 5.1 Modelo real - Realidad de los Objetos

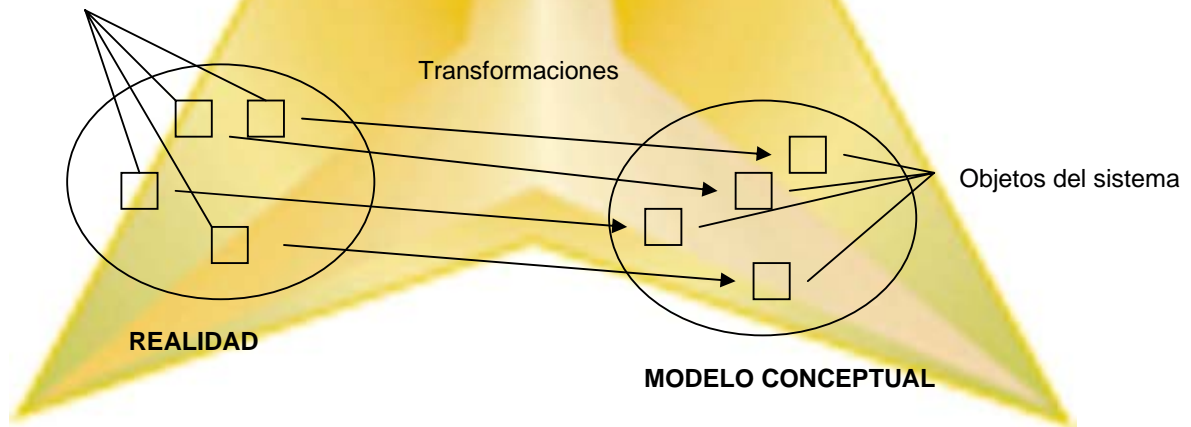
En el mundo real los objetos son todo aquello que se puede sentir, ver y medir; por lo tanto los objetos de la realidad tienen estas características y representan una realidad que puede ser llevada a un sistema. Estos objetos que actúan y escenifican en la realidad son llevados a un modelo de objeto; estos objetos serán, los objetos que se transformarán en objetos del sistema y los que el computador y software, los utilizarán. Estos objetos pueden ser los siguientes:



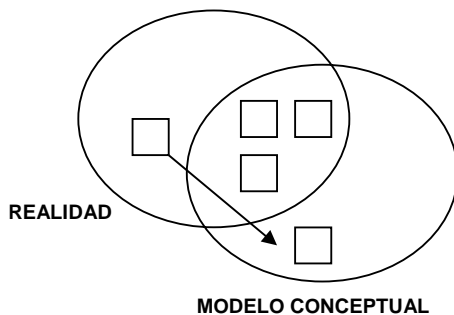
Fig. 21 Los objetos

### 5.2 Modelo de Objetos del Sistema - Transformaciones

Elementos de la realidad



### PARADIGMA DEL TÉRMINO USADO ORIENTADO A OBJETO



Entidades y estructuras del programa que corresponde en un grado más preciso de la realidad.

Podemos mirar dentro de la programación OO como objetos del mundo virtual real.

**Problemas que soluciona el modelo conceptual**

- (1) Identificación de las entidades (objetos, métodos, metodología,...)
- (2) Nombra las entidades del mundo real con las entidades del modelo conceptual.
- (3) Especificaciones de las entidades en espacio y tiempo.
- (4) Identificación y especificaciones de la estructura del objeto.

**Herramientas para el AOO y DOO**

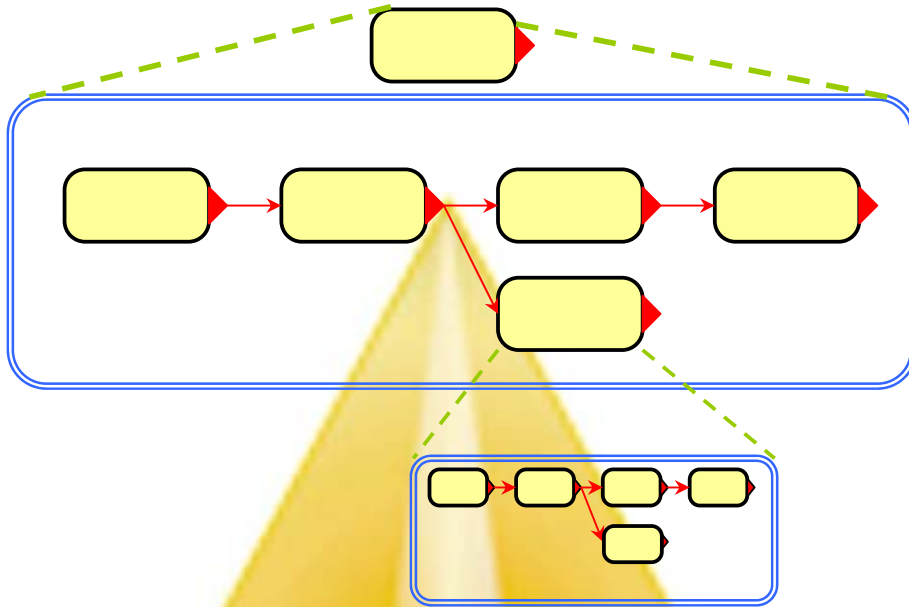
- (1) ORD Diagrama de relación de objetos
- (2) STD Diagrama de transición de estado.

**5.3 El Diagrama de Eventos del Sistema - Capas y Niveles**

El diagrama de evento representa los servicios de los objetos del sistema; los objetos realizan servicios y se comunican entre sí; el diagrama de evento representa las operaciones de los servicios que realizan los objetos del sistema. El diagrama de eventos puede representar las operaciones que realizan un sistema. El diagrama de evento contiene los procesos que se realizan; los mensajes que se envían de evento a evento; regla de activación, que permite realizar un proceso o tarea determinada por el objeto o por el sistema.


**Fig. 22 Diagrama de Eventos**
**CAPAS - NIVELES**

Las capas representan los niveles en que se subdivide un proceso o tarea determinada en un evento. Podríamos decir que un nivel representa un proceso mayor y el siguiente nivel representa una descomposición del proceso en otros procesos más detallados, que explique con precisión el proceso de nivel superior.



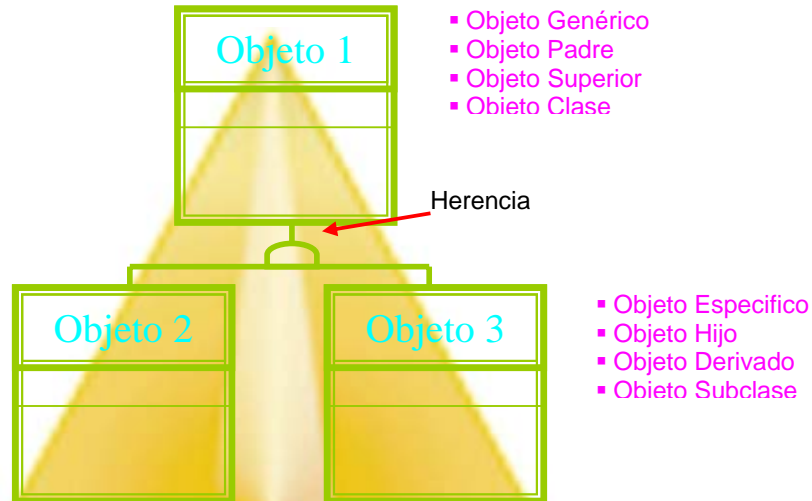
**Fig. 23** Diagrama de Eventos por Niveles

## 6.0 ESTRUCTURA DE LA CLASIFICACION DE OBJETOS

### 6.1 Estructura de Generalización - Especialización

**El Objeto Genérico:** Es el objeto que contiene todos los atributos comunes que tiene todos los demás objetos del sistema; en este objeto se encontrará todas las características comunes que tiene los objetos de especialización.

**El Objeto de Especialización:** Es el objeto se diferencia del objeto del objeto genérico y de los demás objetos por tener sus propias características y haber heredado las características del objeto genérico.



### 6.2 Tipos de Estructuras de Generalización

La estructura de generalización, puede extenderse a más niveles de relación de objetos; estos niveles no deben exceder del cuarto nivel como máximo. Las estructura de generalización pueden simples o compuestas; una estructura de generalización simple se compone de un objeto clase y de un objeto subclase. Mientras una estructura de generalización compuesta, esta formado por una clase y varias subclases o objetos derivados.

#### Clase de Objeto

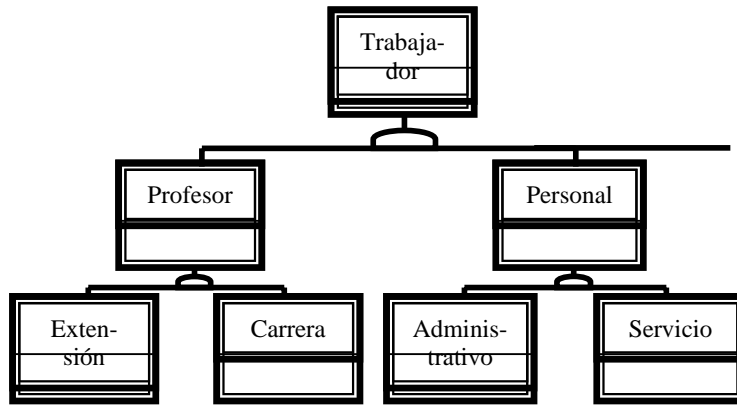


#### Subclase de Objeto

#### Clase de Objeto



#### Subclase de Objeto

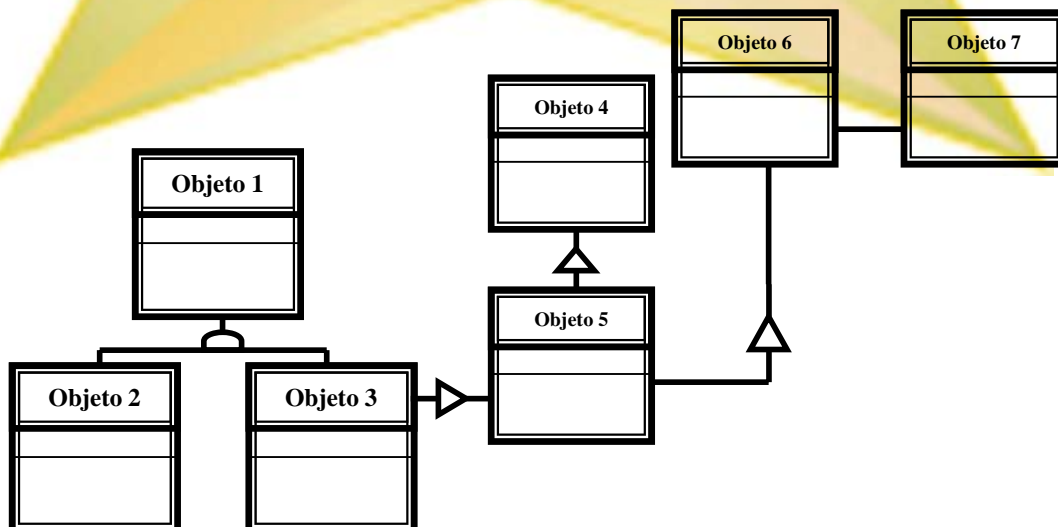


Estructura de Generalización de varios niveles (Superclase, Clase, Subclase)

### 6.3 Estructura todo Parte - Tipos



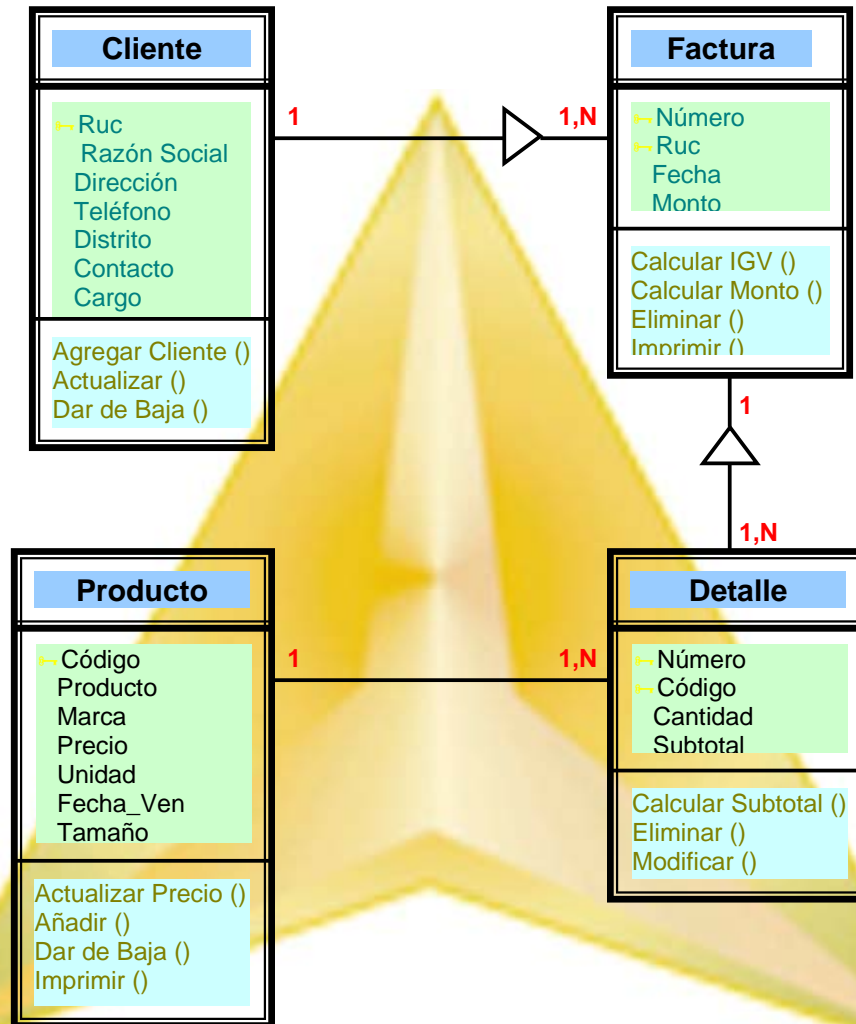
### 6.4 Relaciones Entre los Objetos





### 6.5 Cardinalidad entre los Objetos

La Cardinalidad entre los objetos determina el grado de intercambio de datos entre los objetos del sistema; al igual que las entidades los objetos intercambian datos en áreas comunes, y estas áreas determinan la cantidad de datos que se deberán intercambiar; y tenemos así las cardinalidades siguientes: de uno a ninguno, de uno a uno, de uno a varios, de varios a varios y de varios a ninguno. También la Cardinalidad entre objetos se presenta las restricciones de la Cardinalidad como en las entidades.





## 7.0 DEFINICIONES DE LA ESTRUCTURA DEL OBJETO

### 7.1 Atributos del Modelo de Objetos

Los atributos describen el estado del objeto. Un atributo consta de dos partes: un nombre de atributo y un valor de atributo. Los atributos de un objeto pueden ser de tres tipos:

**Atributo público:** Los atributos públicos son accesibles por cualquier usuario de la clase. Estos atributos se pueden acceder desde cualquier parte del objeto, como también se utilizan para poder comunicarse entre las clases. Tales atributos representan la interfaz de usuario de las clases.

**Atributo protegido:** Los atributos protegidos tienen un nivel de ocultamiento de los datos entre los métodos públicos y privados. Normalmente se utilizan cuando se deriva una nueva clase de una clase predefinido. Son privadas para el mundo exterior y son accesibles dentro de la clase en que están definidos y en cualquier clase derivada de la clase original.

**Atributo privado:** Los atributos privados, son atributos que representa a los datos locales de la clase; es decir, sólo se puede acceder a un dato privado mediante otro método privado de la clase. Al igual una variable local solo está disponible en el interior de los métodos que esta definida.

**Clasificación de los Atributos:** Los atributos se clasifican en:

Atributo clave: es un atributo que contiene datos únicos e identifica a la clase

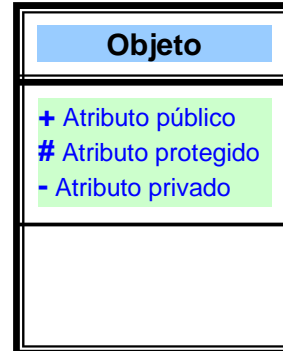
Atributo simple: es un atributo común dentro de la clase.

Atributo multievaluados: es un atributo que puede asumir más de un valor

Atributo monoevaluados: es un atributo que puede asumir un solo valor

Atributo compuesto: es un atributo se compone de atributos simples

Atributo derivado: es un atributo que se deriva de otro atributo.



### 7.2 Métodos del modelo de objetos

Los métodos (operaciones o servicios) describen el comportamiento asociado a un objeto. Representan las acciones que pueden realizarse por un objeto o sobre un objeto. La ejecución de un método puede conducir a cambiar el estado del objeto o dato local del objeto.

Cada método tiene un nombre y un cuerpo que realiza la acción o comportamiento asociado con el nombre del método. Todos los métodos que alteran o acceden a los datos de un objeto se definen dentro del objeto. Un objeto puede modificar directamente o acceder a los datos de otros objetos.

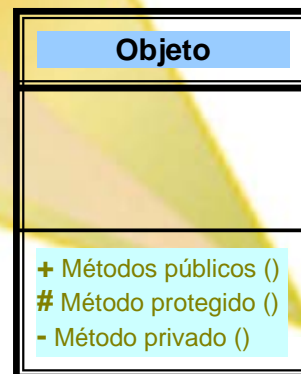
Un método dentro de un objeto se activa por un mensaje que se envía por otro objeto que contiene el método. De otro modo alternativo, se puede llamar al método del mismo objeto por un mensaje local enviado de un método a otro, dentro del objeto.

Los métodos pueden ser de tres tipos:

**Método público:** Los métodos públicos, nos permiten que el usuario pueda usarlo como una interfaces entre el objeto y el usuario. Además, a través de los métodos públicos los objetos pueden comunicarse con los demás métodos del objeto y con otros objetos.

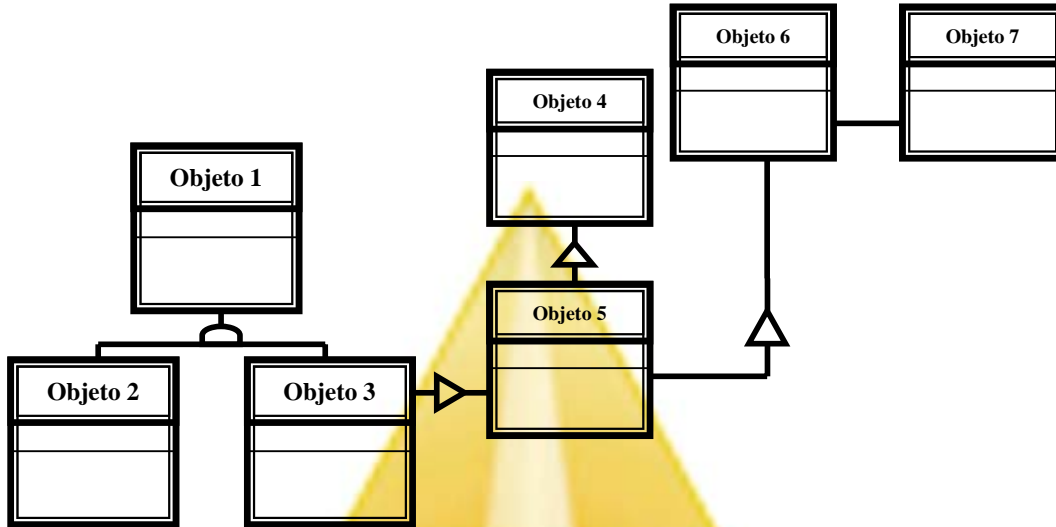
**Método protegido:** Los métodos protegidos pueden ser utilizado por métodos y funciones amigas del objeto que se declare. Y los por métodos de objetos derivados de este objeto.

**Método privado:** Los métodos privado se refieren a métodos que pertenece a una sección del objeto en el interior del objeto. Sólo se puede acceder por métodos o funciones del objeto que están declaradas.



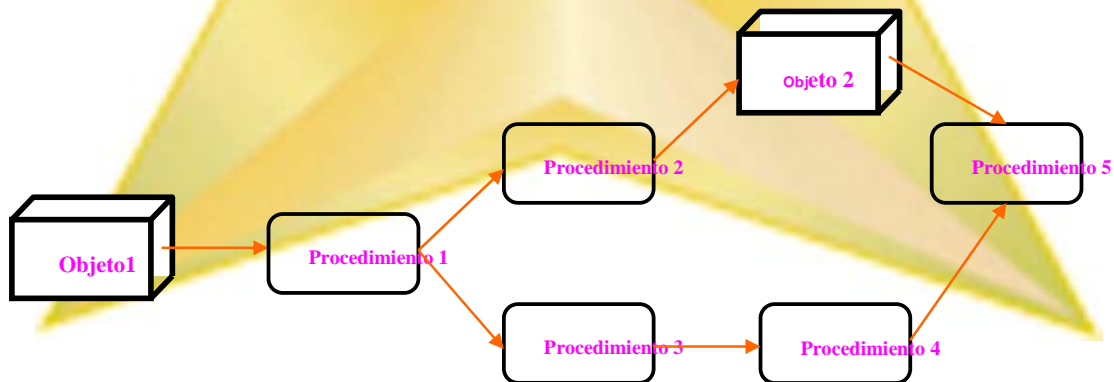
### 7.3 Diagrama de Relaciones del Objeto

Un tipo de modelo que se emplea prácticamente en todas las metodologías OO es el diagrama de relación de objetos. Los diagramas representan las estructuras de los objetos. Contiene los conceptos de tipos, subtipos, y supertipos de objetos, así como las asociaciones entre tipos de objetos.



### 7.4 Diagrama de Eventos del Modelo de Objetos

Los diagramas de eventos expresan los escenarios en forma de flujo de trabajo. Los diagramas de eventos, entonces, son diagramas de flujo de trabajo orientado a objetos. Los diagramas de eventos se representan utilizando cuatro conceptos básicos relacionados con los procesos: operación, eventos, activadores y condiciones de control.





## 8.0 DICCIONARIO DE DATOS DEL SISTEMA

Los objetos del sistema se dividen en dos partes; la parte de los atributos del objeto, son los datos que el objeto contiene y así poder conocer y clasificar el objeto del sistema. Estos datos son transportados entre los métodos del objeto, como también entre los otros objetos del sistema; cuando esto sucede, se debe determinar la cantidad de datos que fluyen en los mensajes que se envían los objetos y que tipos de datos son los que se envían y que tamaño ocupan en el flujo de los mensajes. Para ellos se debe describir a los datos, determinar su estructura y los valores que estos datos deben aceptar y ello se realiza en el diccionario de datos del sistema.

### 8.1 Definición del Diccionario de Datos

El diccionario de datos es un listado de todos los elementos organizados de los datos a ser incluido en un flujo de datos de los mensajes que se envían los objetos entre sí o hacia otros objetos, pertenecientes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tenga un entendimiento común de todas las entradas y salidas, componentes de almacenes y cálculo intermedios. El diccionario de datos del sistema se desarrolla durante el análisis de los flujos de datos y ayuda al analista en la determinación de los requerimientos del sistema. El diccionario de datos define los datos haciendo lo siguiente:

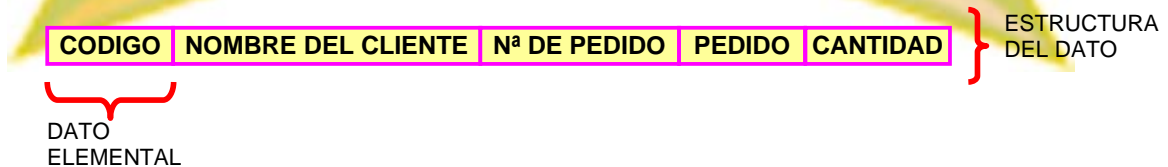
- Describe el significado de los flujos y atributos que contiene los objetos y clases en el diagrama de relación de objetos.
- Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, es decir, paquetes complejos. (Por ejemplo el domicilio del cliente), que pueden descomponerse en unidades más elementales (como ciudad, estado y código postal).
- Describe la composición de los atributos de datos en los objetos del sistema.
- Especifica los valores y unidades relevantes de piezas elementales en información en los flujos de datos y en los atributos de los objetos.
- Describe los detalles de las relaciones y asociaciones entre los objetos que se enfatizan en un diagrama de relación de objetos.

### 8.2 Estructura del dato

**Dato elemental:** Es el nivel de datos esencial, elemental. También se le puede llamar por otros nombres: campo, dato o parte elemental. No hay ninguna unidad menor que tenga significado para el usuario del sistema.

Los elementos datos son los bloques que construye todos los otros datos del sistema y por sí mismo no proporciona suficiente información a ningún usuario.

**Estructura de datos:** Es el conjunto de datos que están relacionados entre sí y que describen en forma colectiva un componente del sistema. Tanto el flujo de datos como los atributos de los objetos del sistema, son estructura de datos. Consiste en elemento relevante que describen la actividad o entidad que esta estudiando.



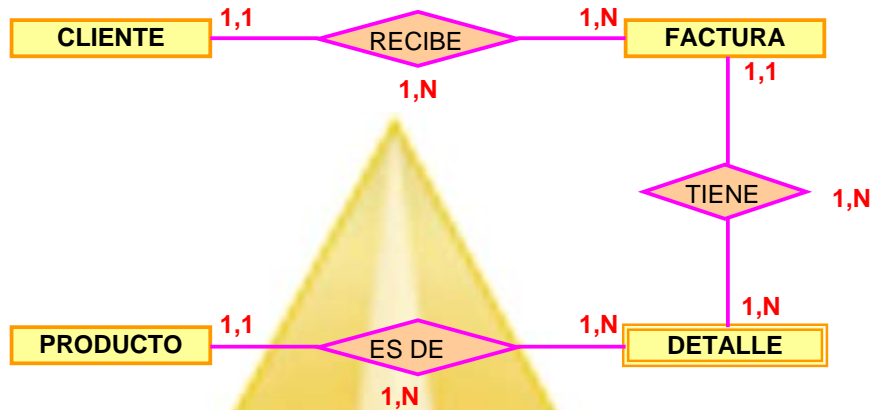
DATO SIMPLE → CODIGO



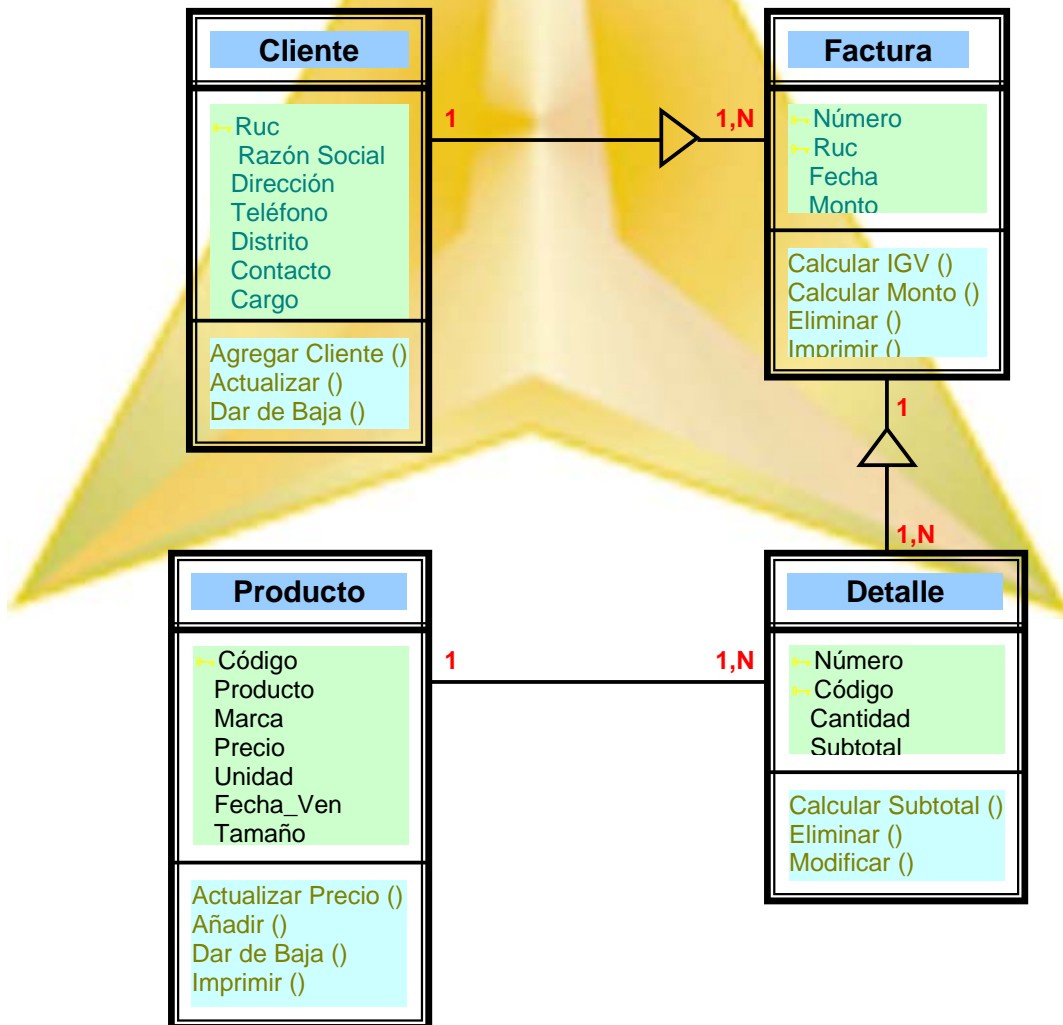
### 8.3 Conversión del Modelo de Objetos a un Modelo de Datos

Los atributos de los objetos se describen en la estructura de datos que tiene el objeto; esta estructura deberá ser almacenada en dispositivos de almacenamiento que viene hacer las bases de datos y para lo cual el modelo de objeto deberá ser convertido a un modelo de datos y así poder guardar los datos en los dispositivos de almacenamiento.

#### MODELO DE DATOS



#### MODELO DE OBJETOS





## 9.0 ANALISIS DE OBJETOS CON UML

El UML es Unified Modeling Language; es un lenguaje de propósito general para el modelado de orientado a objetos. UML combina notaciones provenientes desde:

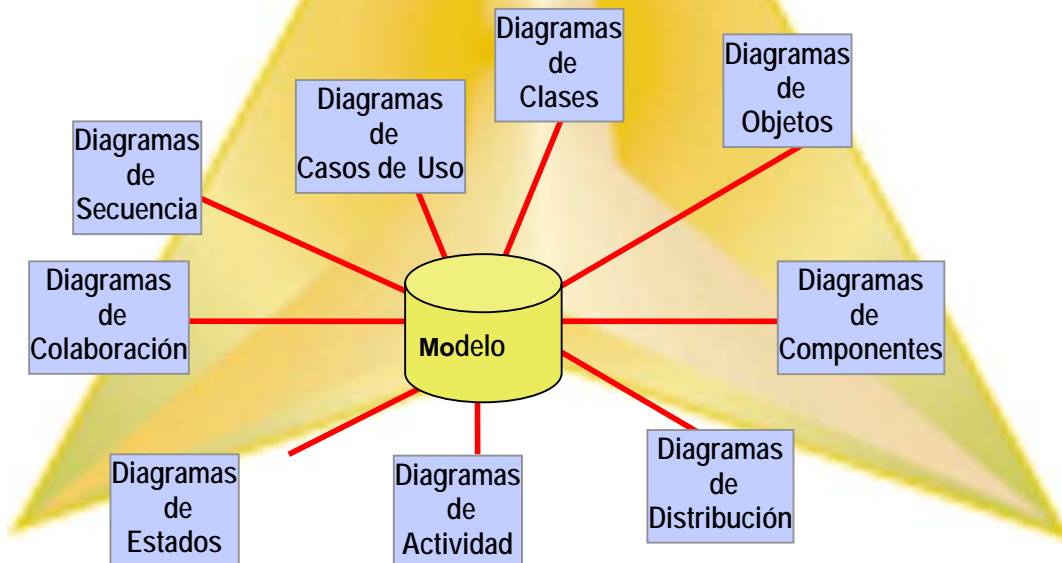
- Modelado Orientado a Objetos
- Modelado de Datos
- Modelado de Componentes
- Modelado de Flujos de Trabajo (Workflows)

El UML es diversos métodos y técnica OO, con muchos aspectos en común pero utilizando distintas notaciones.

El UML comenzó como el "Método Unificado", con la participación de Grady Booch y Jim Rumbaugh. Se presentó en el OOPSLA'95. El mismo año unió Ivar Jacobson. Los "Tres Amigos" son socios en la compañía Rational Software. Herramienta CASE Rational Rose.

Los diagramas que comprende el UML son los siguientes:

- Diagrama de Casos de Uso
- Diagrama de Clase (incluyendo Diagrama de Objetos)
- Diagrama de Comportamiento
  - Diagrama de Estados
  - Diagrama de Actividad
  - Diagrama de Interacción
    - Diagrama de Secuencia
    - Diagrama de Colaboración
- Diagramas de Implementación
  - Diagrama de Componentes
  - Diagrama de Despliegue



### 9.1 Diagrama de Caso de Uso

Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja actualmente, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

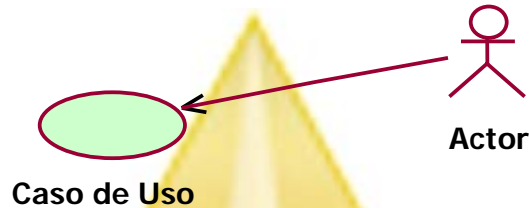
Los Casos de Usos (Ivar Jacobson) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario; permite definir los límites del sistema y las relaciones entre el sistema y el entorno; los Casos de Uso son descripciones de la funcionalidad del sistema independiente de la implementación. Comparación con respecto a

los Diagramas de Flujo de Datos del enfoque estructurado. Los Casos de Usos cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos. Los Casos de Uso particiona el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo, esta basado en el lenguaje natural, es decir, es accesible por los usuarios. Los componentes de los Casos de Usos son: los actores y estos pueden ser; actores principales, personas que utilizan el sistema; secundarios, personas que mantienen o administran el sistema; material externo, dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados; otros sistemas, sistema con los que el sistema interactúa.

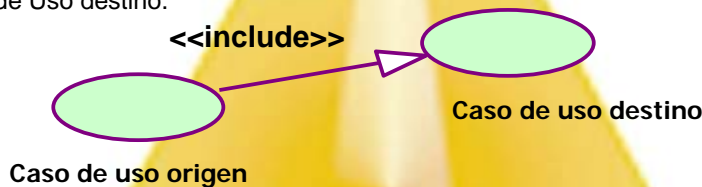
**LAS REALACIONES DE LOS CASOS DE USO:**

UML define cuatro tipos de relaciones en los Diagramas de Casos de Uso:

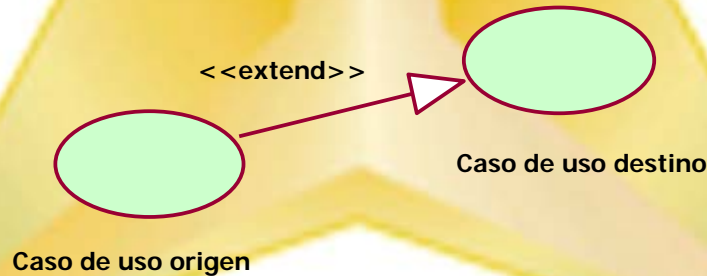
**COMUNICACIÓN:** es la relación que se establece entre el actor un caso de uso; o también entre un caso de uso 1 y un caso de uso 2.



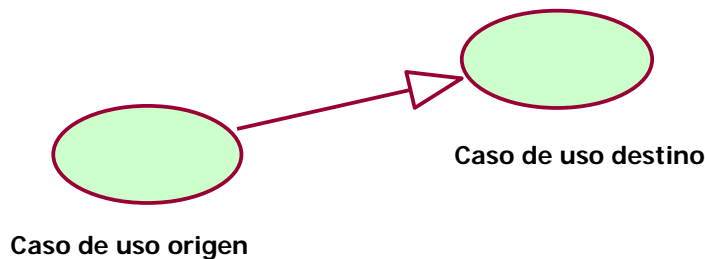
**INCLUSION:** una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino.



**EXTENSION:** el Caso de Uso origen extiende el comportamiento del Caso de Uso destino.

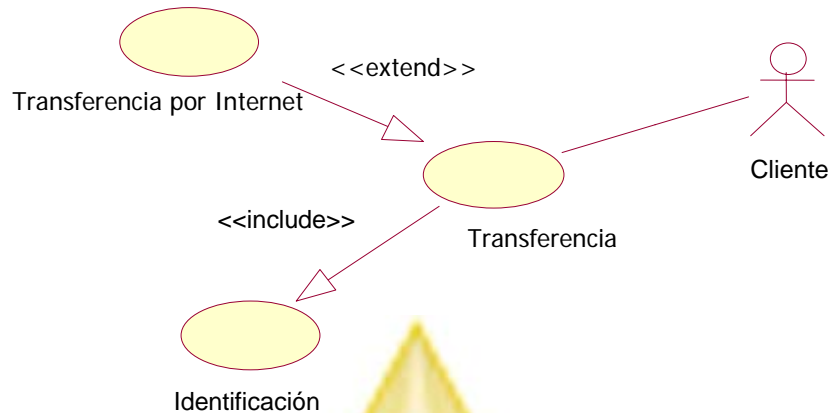


**HERENCIA:** el Caso de Uso origen hereda las especificaciones del Caso de Uso destino y posiblemente la modifica y/o amplía.





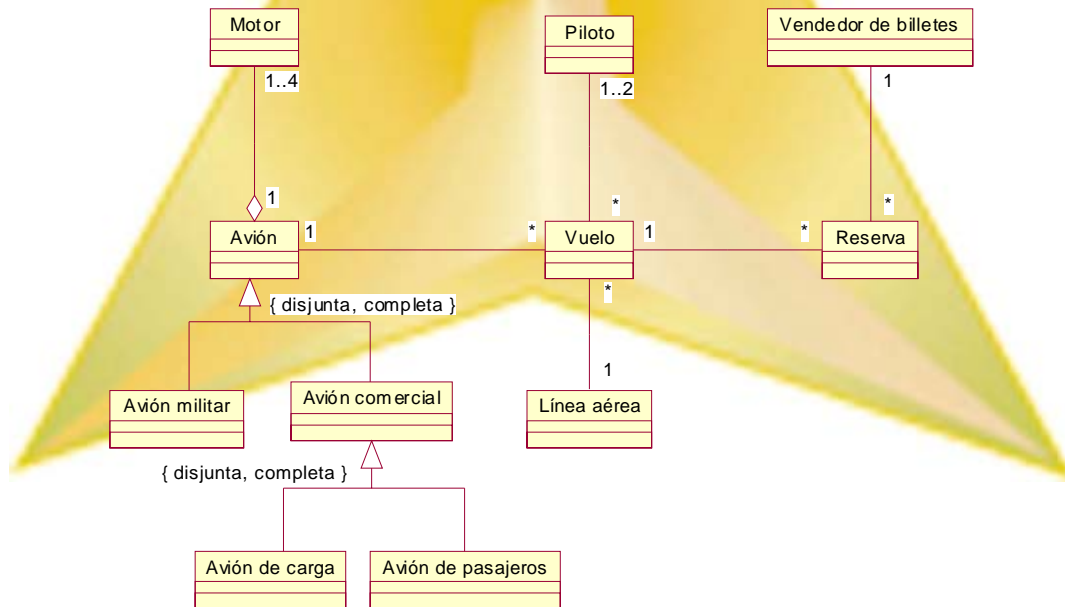
Ejemplo:



9.2 Diagrama de Clases y de Objetos

Los Diagramas de Clases y los Diagramas de Objetos pertenecen a dos vistas complementarias del modelo. Un diagrama de clases muestra la abstracción de una parte del dominio; mientras que un diagrama de objetos representa una situación concreta del dominio; cada objeto es instancia de una clase; ciertas clases (clases abstractas o diferidas) no pueden ser instancias.

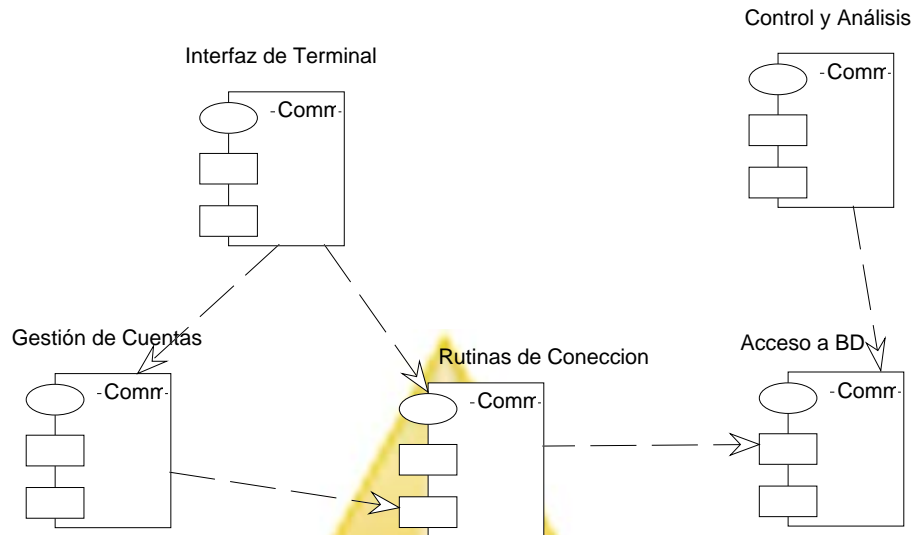
El Diagrama de Clases es el diagrama principal para el análisis y diseño; el diagrama de clases presenta las clases y objetos del sistema con sus relaciones estructurales y de herencia. La definición de clase u objeto incluye definiciones para atributos y operaciones.



9.3 Diagrama de Componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones; muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas

cargadas dinámicamente, etc. Cada clase del modelo lógico se realiza en dos componentes: la especificación y el cuerpo



## 9.4 Diagrama de Estado

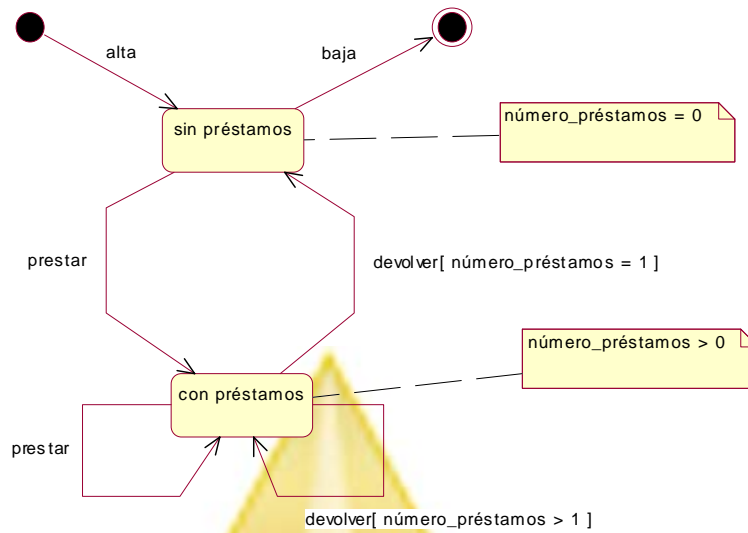
Un Diagrama de Estados muestra el comportamiento de la secuencia de estados por los que pasa bien un caso de uso, bien un objeto a lo largo de su vida, o bien todo el sistema. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino. La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma entrada/acción\_asociada donde acción\_asociada es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta. Una acción de salida aparece en la forma salida/acción\_asociada. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.

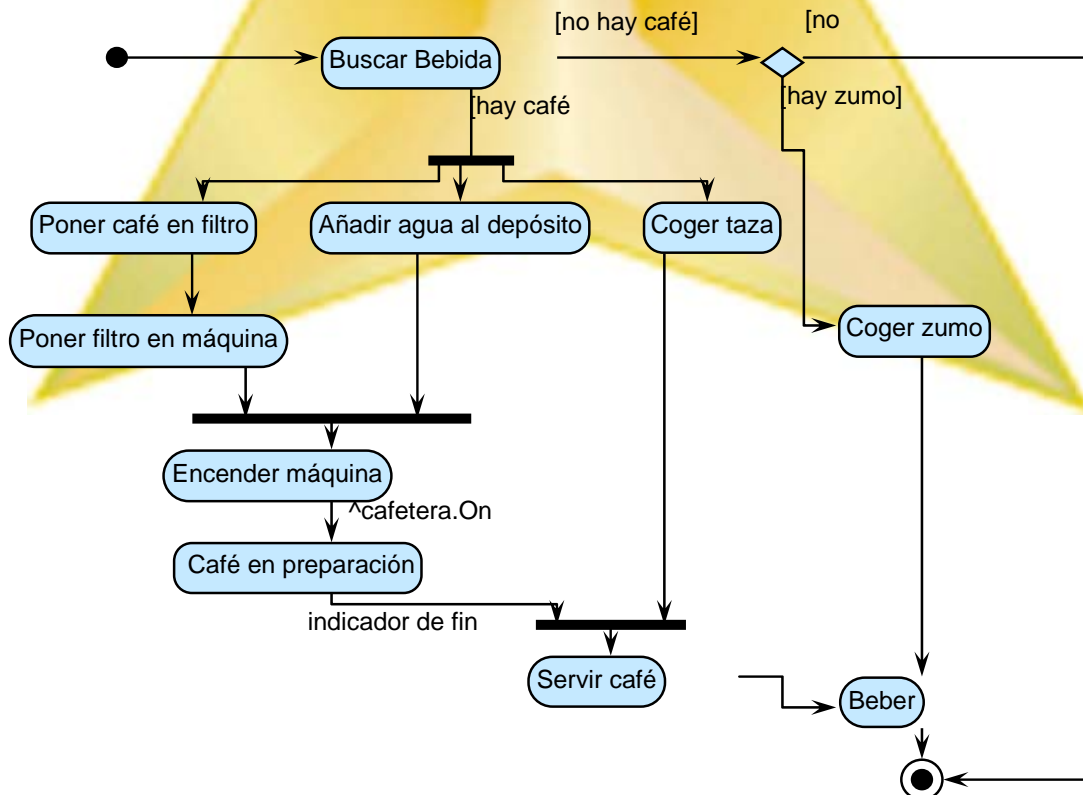
Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma nombre\_de\_evento/acción\_asociada.

Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (finalización del caso de uso o destrucción del objeto). El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad, los estados inicial y final son pseudoestados, pues un objeto no puede “estar” en esos estados, pero nos sirven para saber cuáles son las transiciones inicial(es) y final(es).



### 9.5 Diagrama de Actividad

Los Diagramas de Estados representan autómatas de estados finitos, desde el punto de vista de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. El resto de los objetos se puede considerar que tienen un único estado. El formalismo utilizado proviene de los *Statecharts* (Harel). Cada objeto está en un estado en cierto instante, el estado está **caracterizado parcialmente por los valores de los atributos del objeto**; el estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase. Los Diagrama de Estados y escenarios son complementarios.





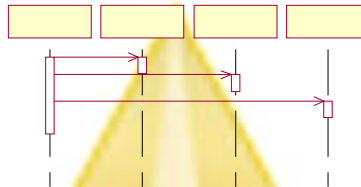
## 9.6 Diagrama de Secuencia

Muestra la secuencia de mensajes entre objetos durante un escenario concreto. Cada objeto viene dado por una barra vertical. El tiempo transcurre de arriba abajo. Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua

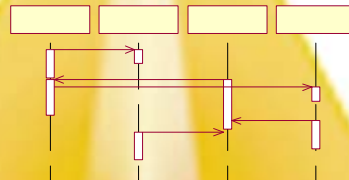
Un objeto puede enviarse a sí mismo un mensaje. Gráficamente también se puede indicar cuándo el mensaje es para crear el objeto (va dirigido al rectángulo del objeto o etiquetado con *new*) o para destruirlo (va dirigido a la línea del objeto pero el final de la flecha es una cruz).

Normalmente no es necesario indicar el retorno del control. En el caso asíncrono el retorno, si existe, se debe representar.

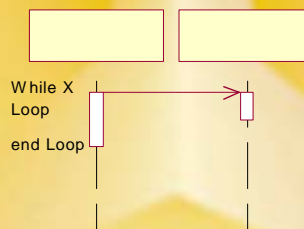
**Tipos de Control:** El Diagrama de Secuencia refleja de manera indirecta las opciones de control. Un control centralizado tiene una forma como esta:



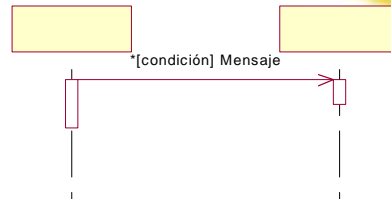
Un control descentralizado tiene una forma como esta:



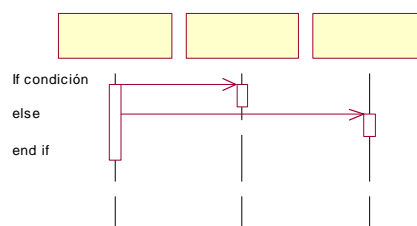
Podemos representar iteraciones en el envío de mensajes, punto de vista, mientras se cumpla una condición:

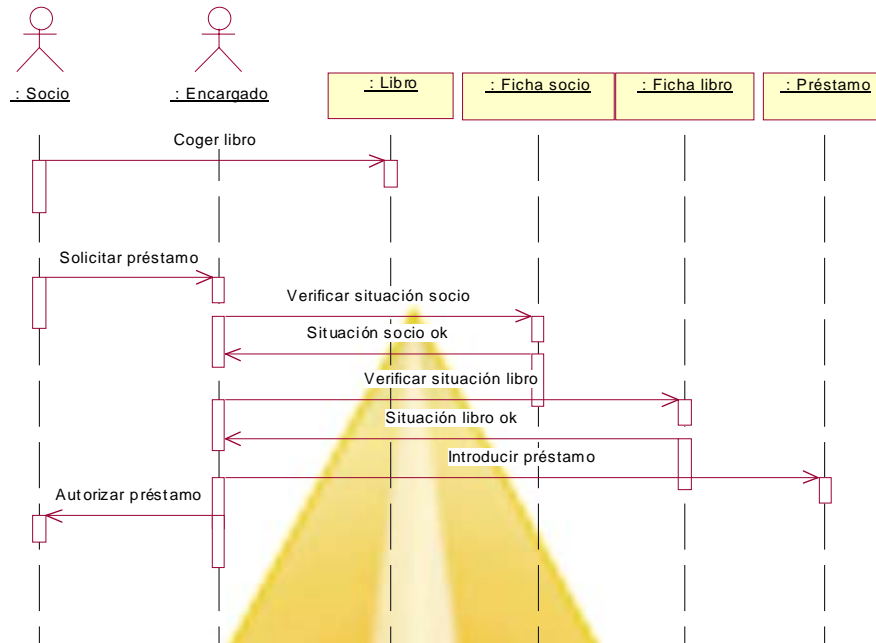


La iteración puede expresarse también como parte del mensaje



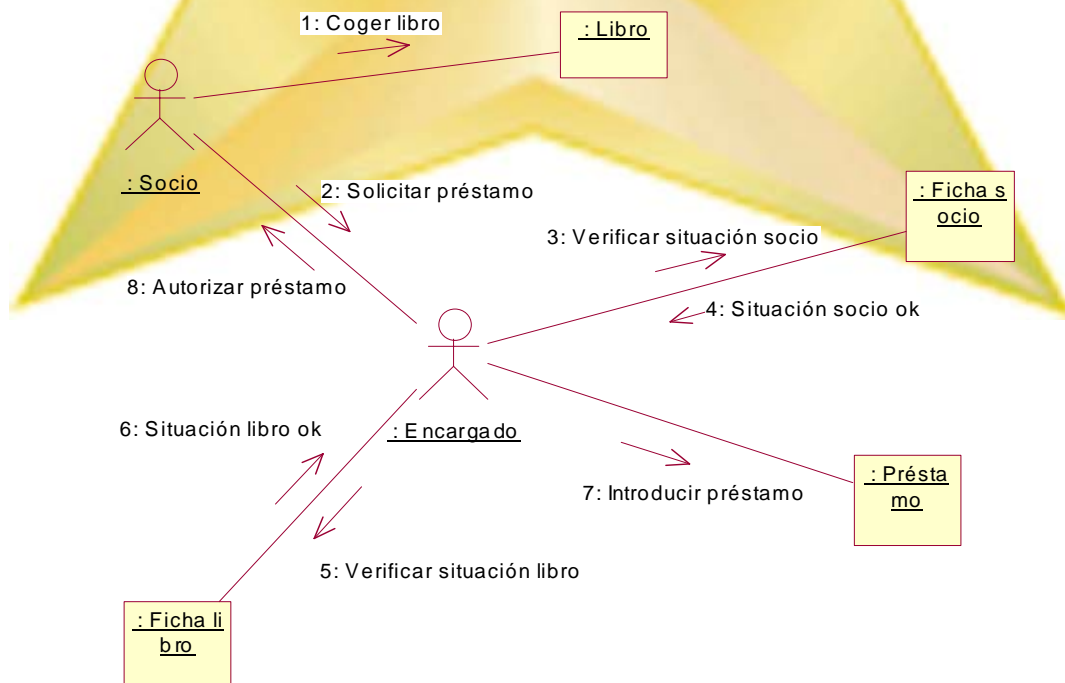
Las bifurcaciones condicionales pueden representarse de esta forma





### 9.7 Diagrama de Colaboración

Son útiles en la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás. La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.



**BIBLIOGRAFIA:**

- (1) DeMarco Ton, *Structured Analysis and System Specification*, editorial Prentice-Hall, New York, 1979
- (2) Hawiyszkiewicz, 1. T, *Análisis y Diseño de Sistemas*, editorial Anaya, Madrid, 1990
- (3) Kendall Kenneth y Kendall Julie, *Análisis y Diseño de Sistemas*, editorial Prentice-Hall Hispanoamericana, México 1991
- (4) Pressman Roger S, *Ingeniería del Software: Un Enfoque Práctico*, editorial McGraw-Hill, Madrid, 1993, tercera edición.
- (5) Senn James, *Análisis y Diseño de Sistemas de Información*, segunda edición, editorial McGraw-Hill, México, 1991
- (6) Coad Peter y Yourdon Edward, *Object Oriented Analysis*, Prentice Hall, New Jersey, 1991
- (7) MartIn James y Odeil James, *Análisis y Diseño Orientado a Objetos*, Prentice Hall Hispanoamérica S.A., México, 1994
- (8) Rumbaugh James, Blaha Michael, Premertani William, Eddy Fredenck y Lorensen William, *Object-Oriented Modelling and Design*, Prentice Hall, New Jersey, 1991
- (9) Yourdon Edward, *Object-Oriented Systems Design: An Integrated Approach*, Prentice-Hall, New Jersey, 1994

