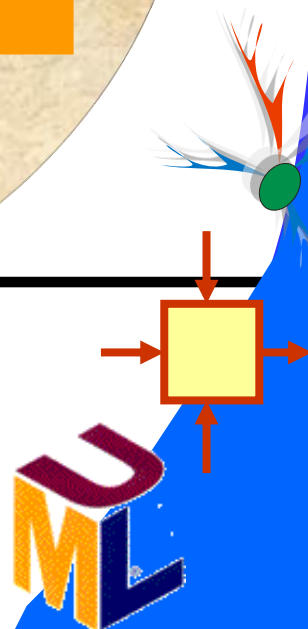




Conceptos básicos

UML

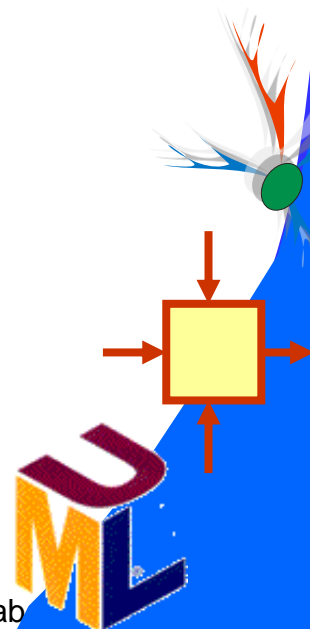


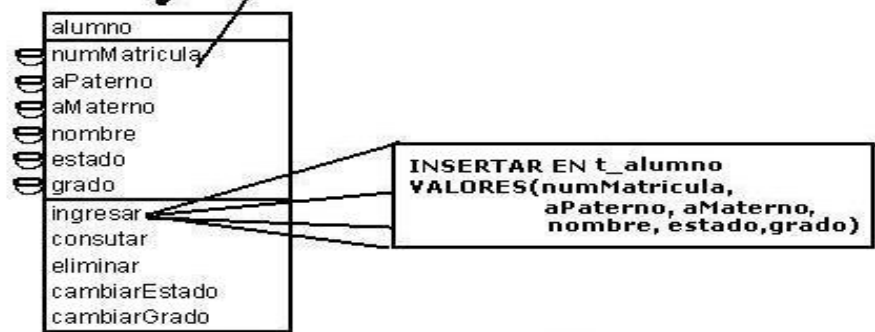
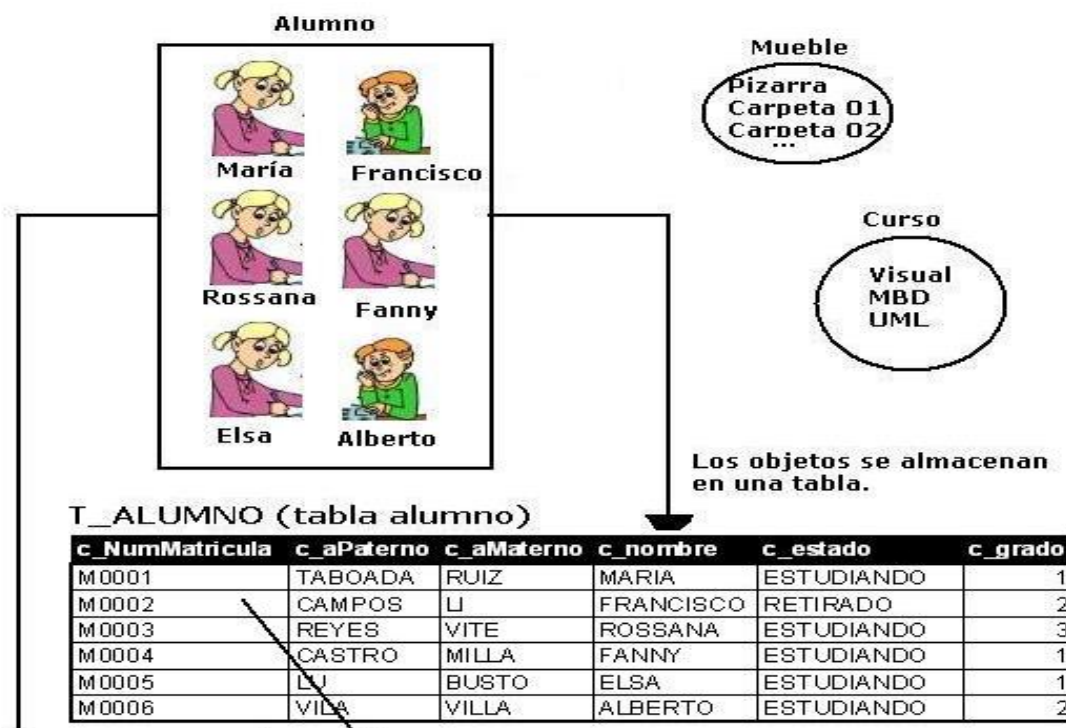


Cómo definimos a una clase.

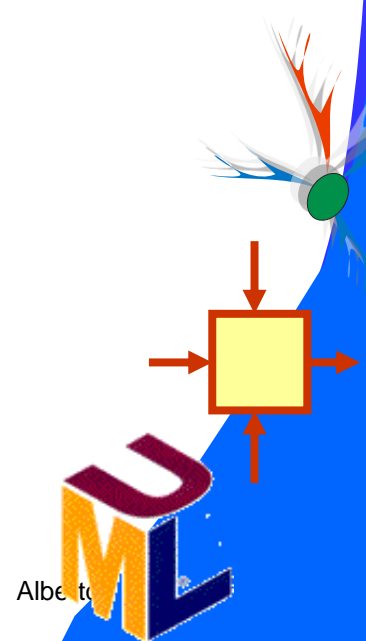
Los objetos que tengan los mismos atributos y comportamiento se agrupan en clases. Una clase es una abstracción que describe (atributos y comportamiento) relevantes para una aplicación determinada, la elección de clases es arbitraria, y depende del dominio del problema. Por lo tanto la clase se puede representar de la siguiente manera:

nombredeClase
atributo1 atributo2 atributo3 atributo4
operacion_1 operacion_2 operacion_3



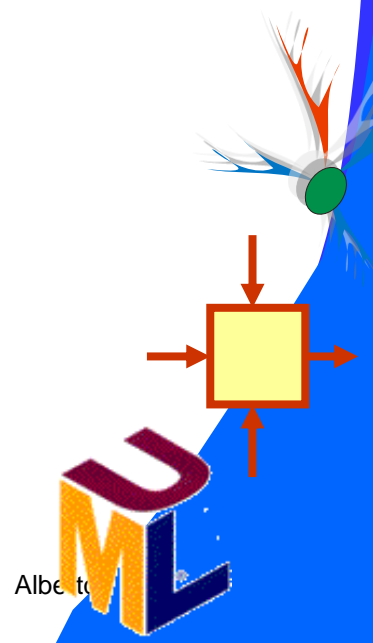
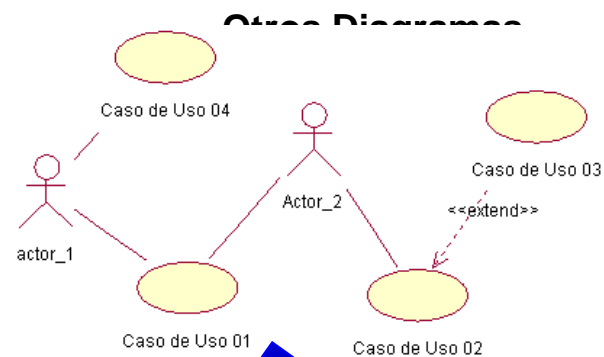
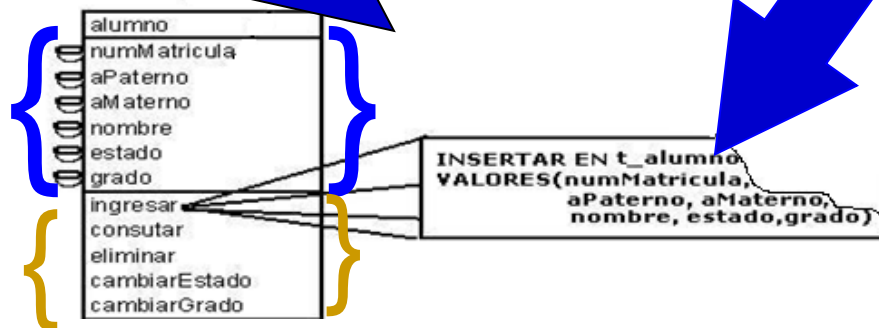
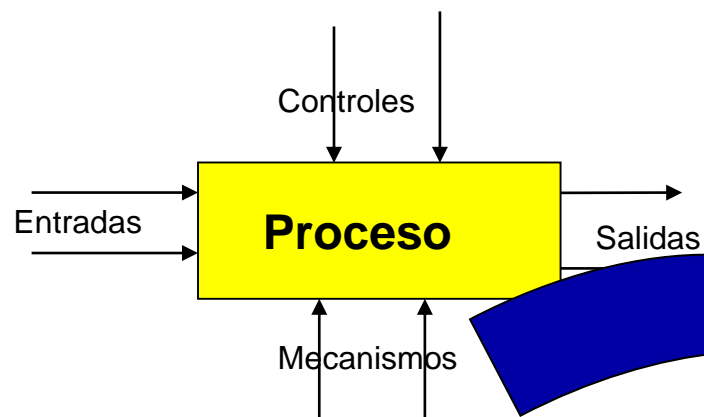


Los atributos de la clase sirven como variables para interactuar con los datos de los objetos. La operaciones tiene líneas de código sobre la lógica del negocio.



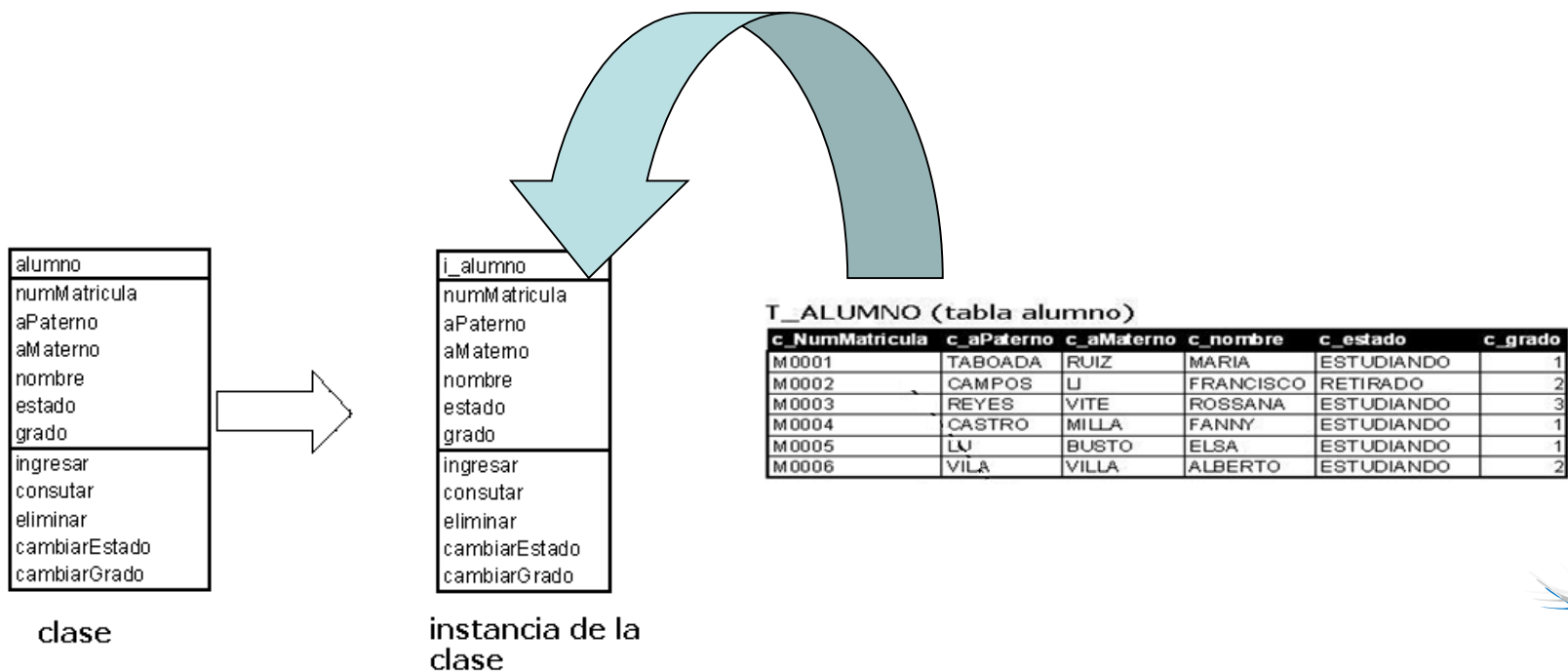


Los Procesos



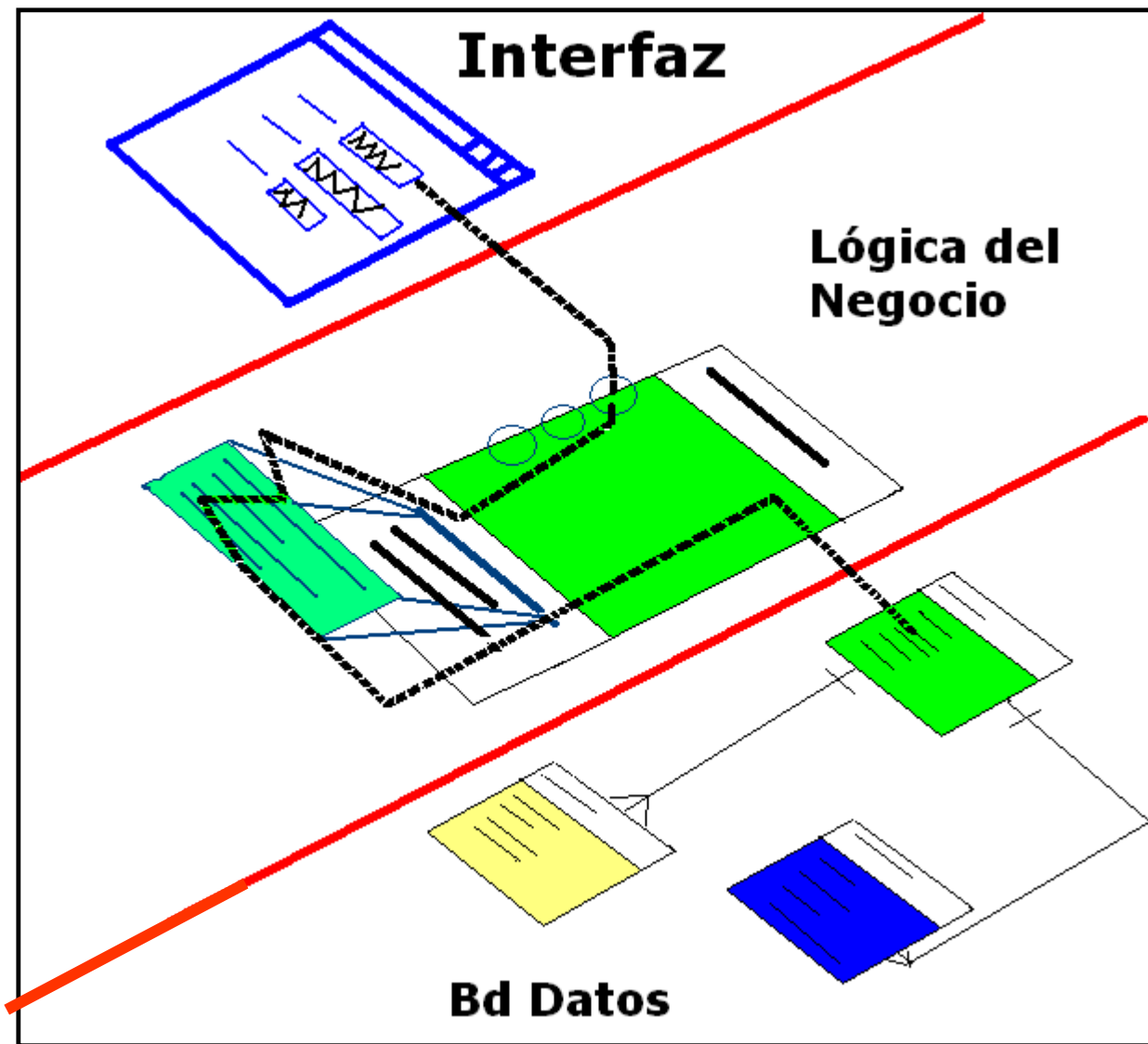


■ Manejo de Datos con la Clase





Cómo Interactúan las Capas





1. Interfaz



C:\carpeta

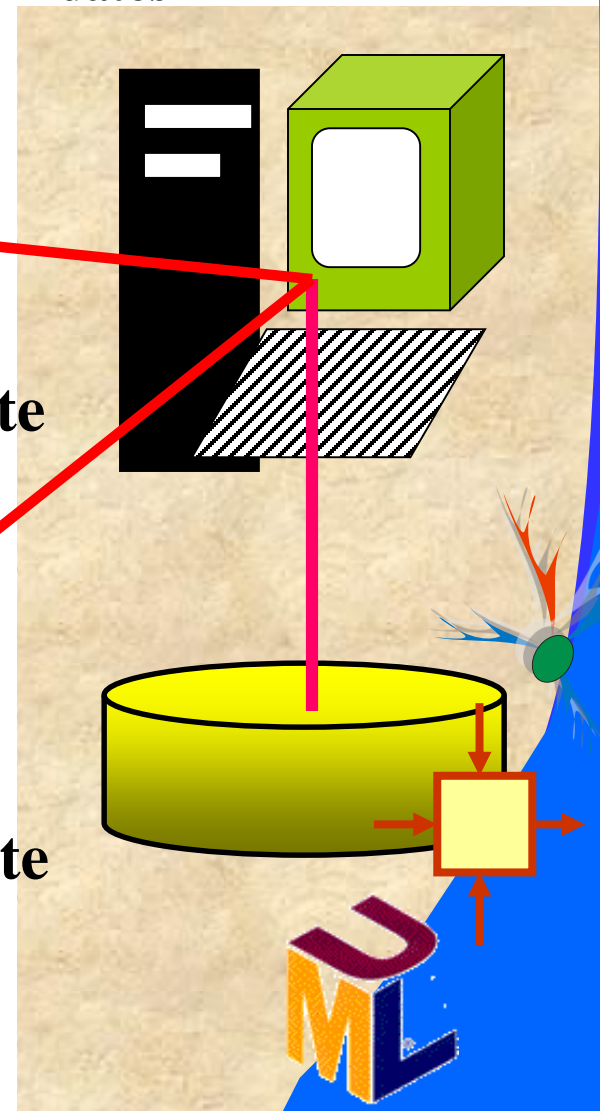
b.dll/paquete

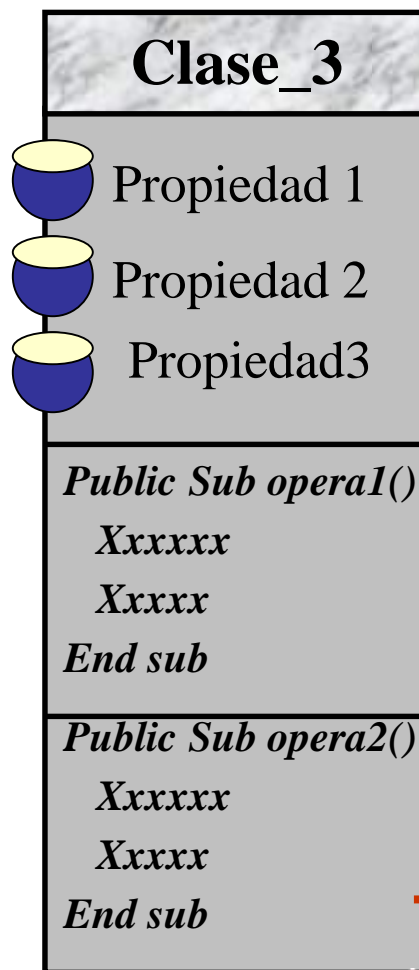
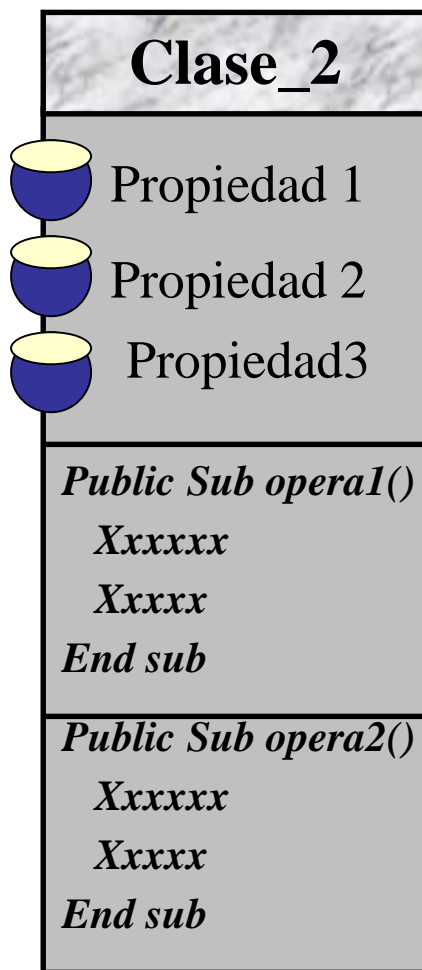
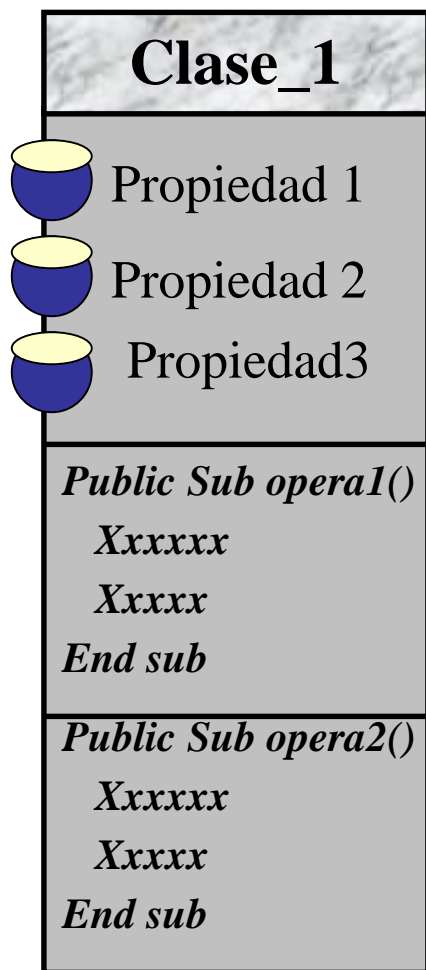
a.dll/paquete

b.dll/paquete

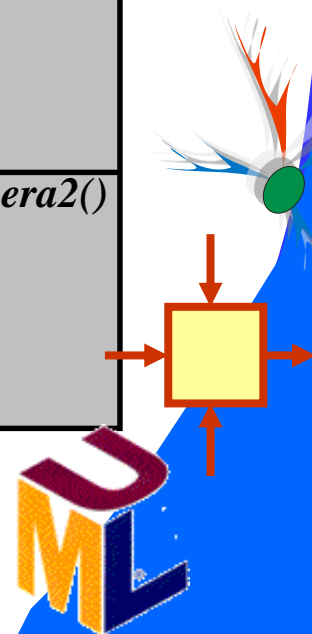
a.dll/Paquete

3. Almacenamiento de datos





Regreso a Componente



Computadora Pentium II

Pedro que paga factura

Impresora

María que emite factura

Computadora Pentium III

Gino que paga boleto

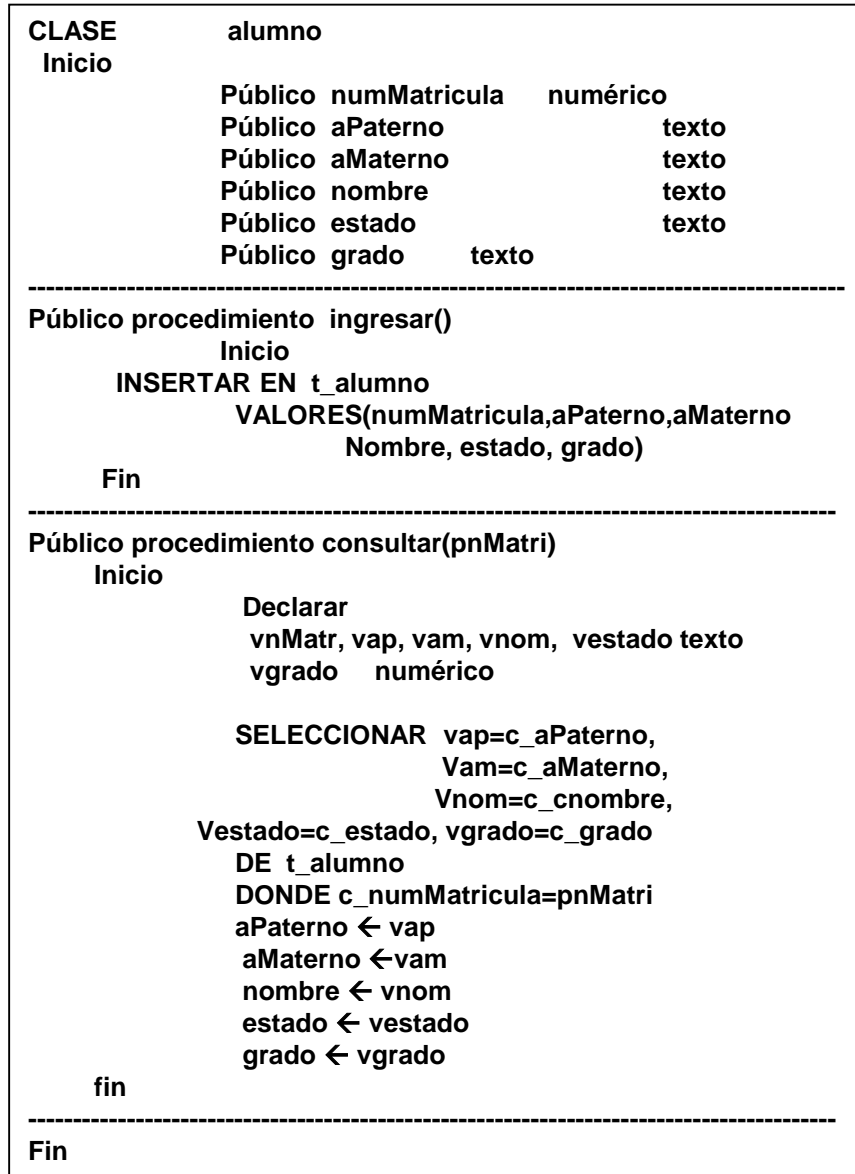
Audífonos

¿Cuántas clases hay?

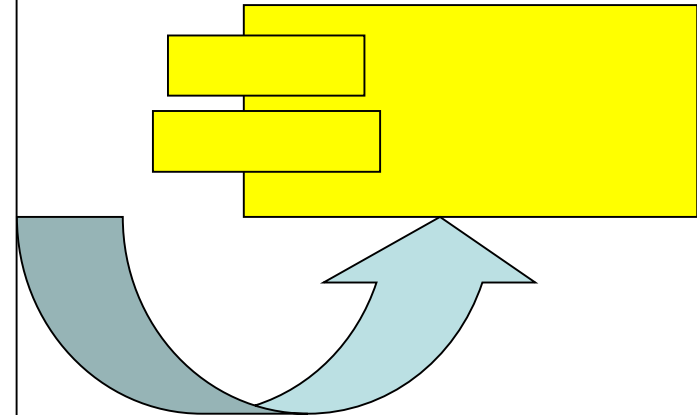
**De que manera manejaría estos objetos en 3
capas**



Implementación de Clases



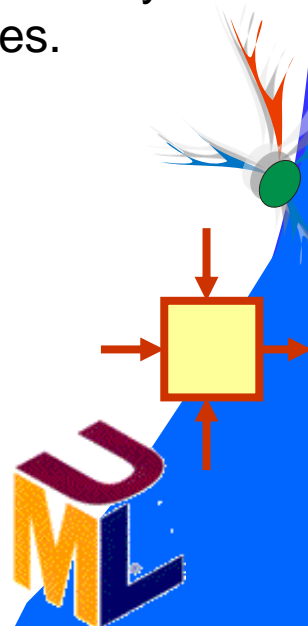
Componente

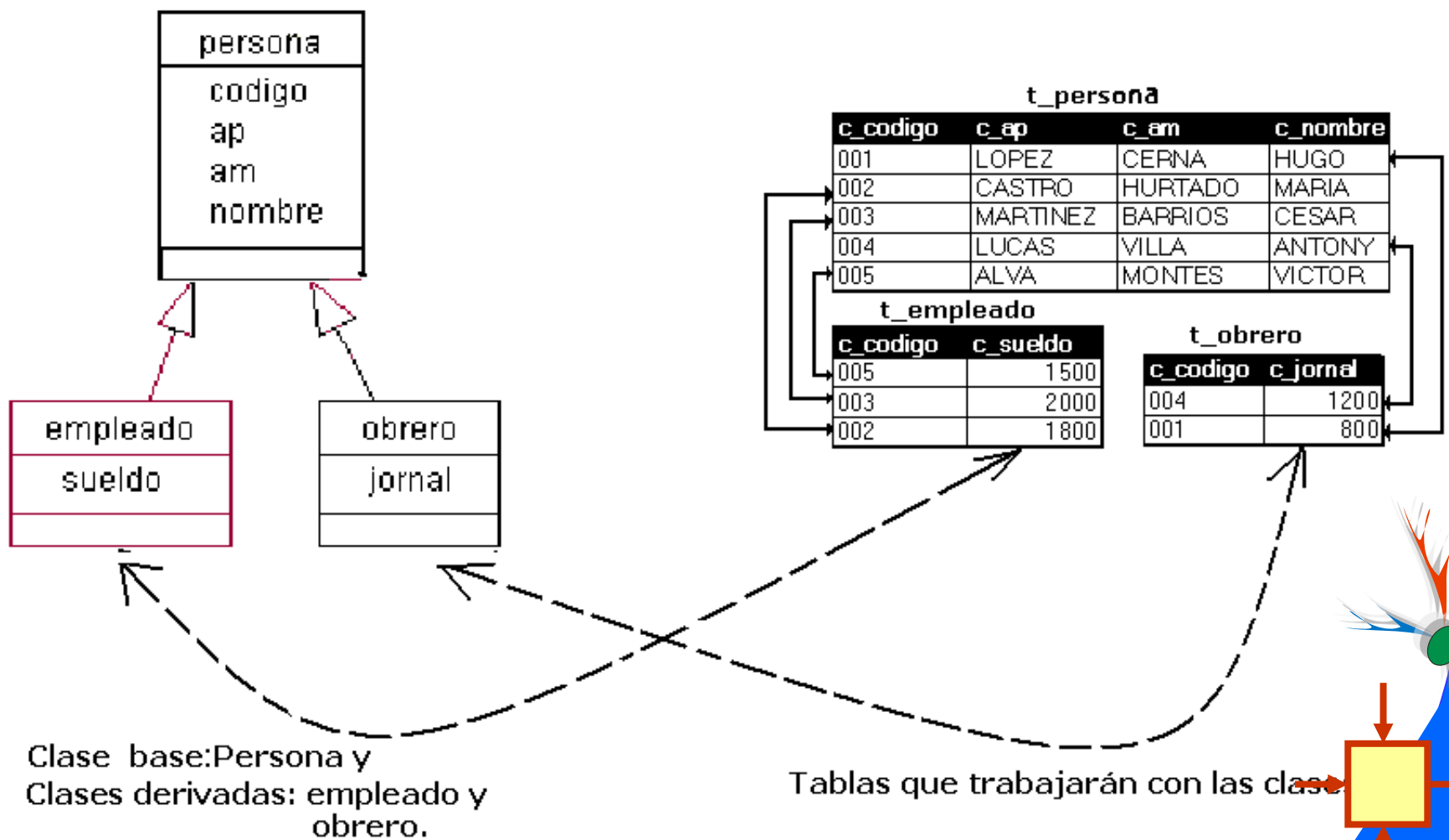




Herencia

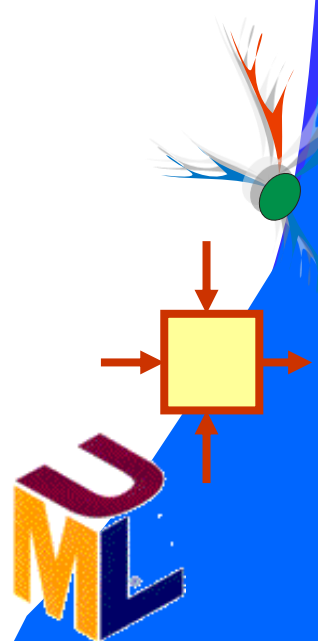
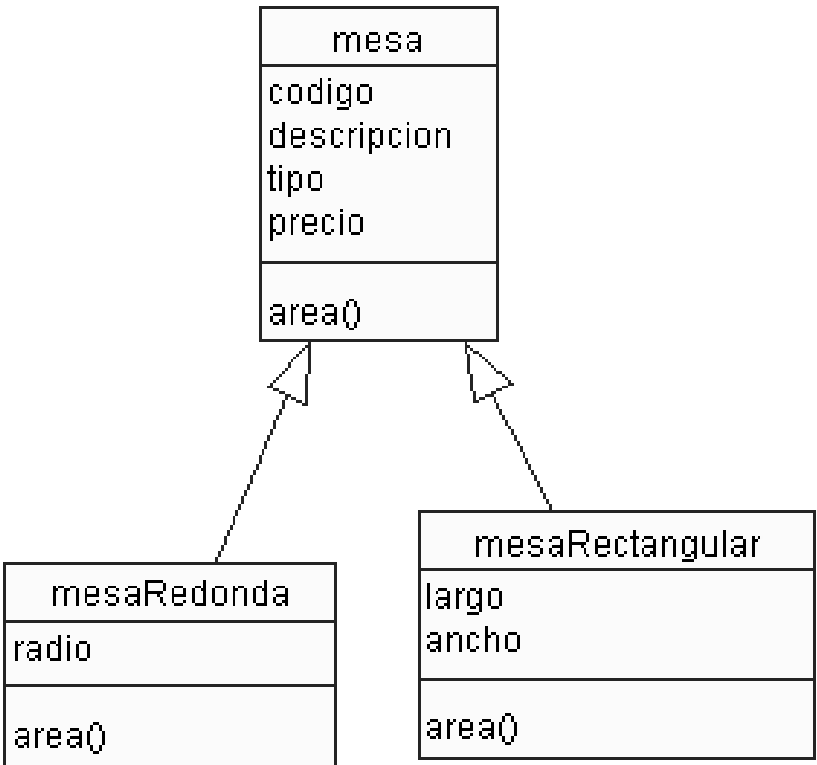
El concepto de herencia se refiere a la compartición de atributos y operaciones basada en una relación jerárquica entre varias clases. Una clase puede definirse de forma general y luego refinarse en sucesivas subclases. Cada clase hereda todas las propiedades (atributos y operaciones) de su superclase y añade sus propiedades particulares.







Ejemplo



Nombre De Clases

- 1 Empleado
- 2 Servicio
- 3 Persona
- 4 Obrero
- 5 Equipo de Oficina
- 6 Servicio de Alquiler Máquinas
- 7 Artefactos
- 8 Útiles
- 9 Servicio de llamadas
- 10 Electrodomésticos
- 11 Servicio de Impresión
- 12 Cliente
- 13 Reglas

Grupos de Atributos

- 1 Código, Marca, Precio Unitario, Stock
- 2 Código, Tipo de Llegada, Tipo de Saldo
- 3 Código, Apellido Paterno, Apellido Materno, Nombre, Edad, Dirección
- 4 Número, Fecha, Tipo de Cancelación (Contado, Crédito)
- 5 Código, Marca, Precio Unitario, Años de Garantía

Atributos

- 1 Sueldo
- 2 Medida
- 3 Jornal
- 4 RUC
- 5 N° Hojas de impresión
- 6 Precio unitario por impresión
- 7 N° de Minutos
- 8 Tipo de llamada
- 9 N° de Horas



Constructor

CLASE mesa

Inicio

Público código texto

Público descripción texto

Público tipo texto

Público precio numérico

Público Procedimiento mesa (pcódigo="", pdescripción="",
ptexto="", pprecio=0)

Inicio

código=pcódigo

descripción=pdéscripción

tipo=ptipo

precio=pprecio

fin

Virtual público función area()

Protegido Destructor_mesa()

Inicio

Destruir instancia

fin

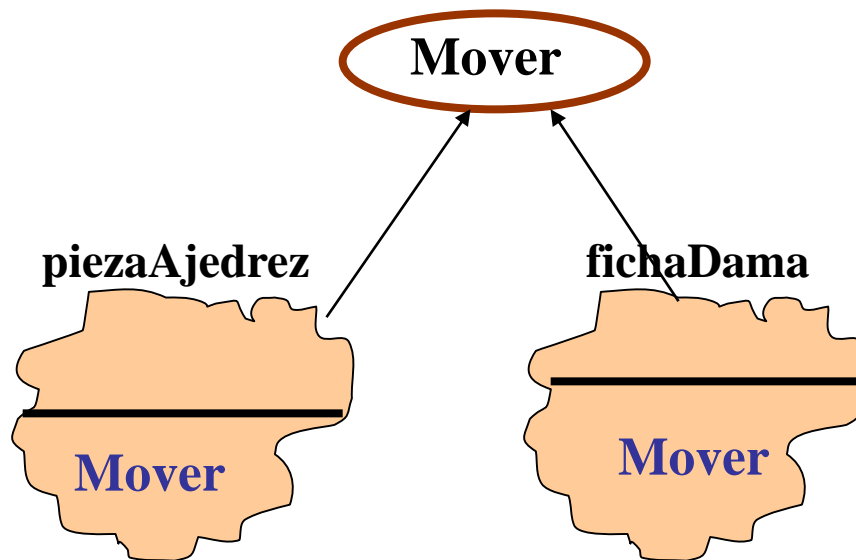
Fin

Destructor

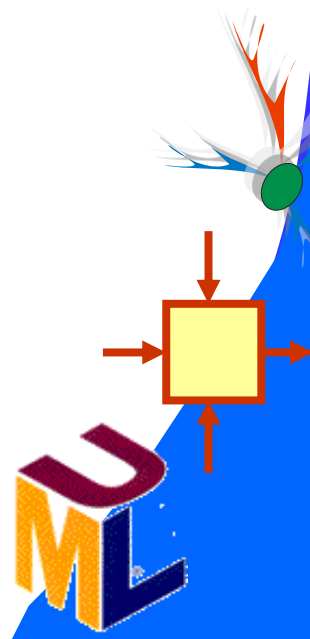
U
ML



Polimorfismo



Por ejemplo, la operación *mover*, es distinta para una pieza de ajedrez que para una ficha de dama, pero ambos objetos pueden ser movidos. Una operación es una acción o transformación que realiza o padece un objeto. La implementación específica de una operación determinada en una clase se denomina *método*.





Clases Derivadas

CLASE mesaRedonda EXTENDIDA mesa

Inicio

Público radio numérico

Público funcion area()

Inicio

Declarar vArea, vpi numérico

$V_{pi} \leftarrow 3.1416$

$vArea = v_{pi} * (radio * radio)$

$area \leftarrow vArea$

fin

Fin

CLASE mesaRectangular EXTENDIDA mesa

Inicio

Declarar

Público largo numérico

Público ancho numérico

Público funcion area()

Inicio

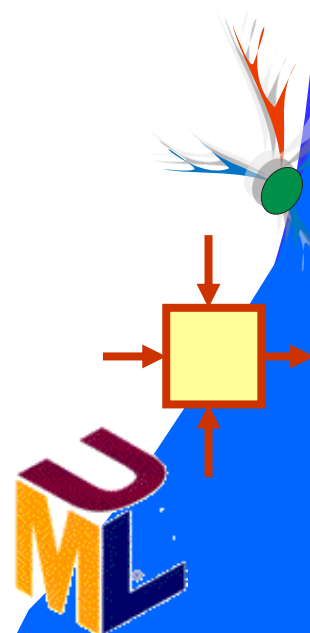
Declarar vArea numérico

$vArea \leftarrow largo * ancho$

$area \leftarrow vArea$

Fin

Fin



inicio

Declarar

vLargo, vAncho, vRadio, vArea numérico

i_mesaRedonda mesaRedonda

i_mesaRectangular mesaRectangular

/* Uso de la Instancia mesa redonda */

Leer "Medida de Radio...:",vRadio

i_mesaRedonda.radio ← vRadio

vArea ← i_mesaRedonda.area

imprimir "El area de la mesa redonda es",vArea

/* Uso de la Instancia mesa rectangular */

Leer "Medida de Largo...:",vLargo

Leer "Medida de Ancho...:",vAncho

i_mesaRectangular.largo ← vLargo

i_mesaRectangular.ancho ← vAncho

vArea ← i_mesaRectangular.area

Imprimir "El área de la mesa rectángular es", vArea

fin

Página Web

Moto

Trabajador

Libro

Avión

Mamífero

Revista

Ave

Auto

Bicicleta.

Publicar()

Comer()

Retroceder()

Editar()

Revisar()

Dormir()

Avanzar()

Detenerse()

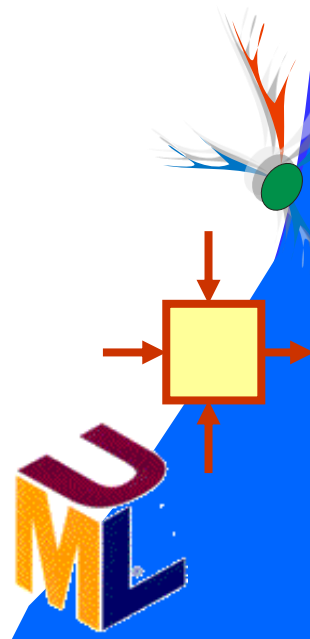
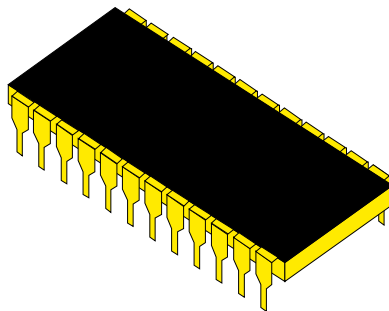
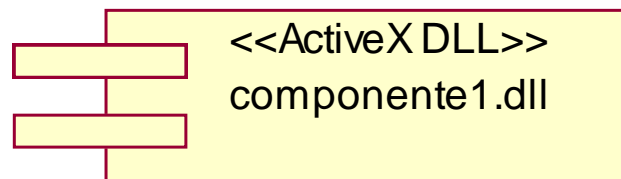
Cambiar de Velocidad()

Caminar()



Encapsulamiento

La esencia del encapsulamiento (o encapsulación), es que cuando un objeto trae consigo funcionalidad, esta última se oculta

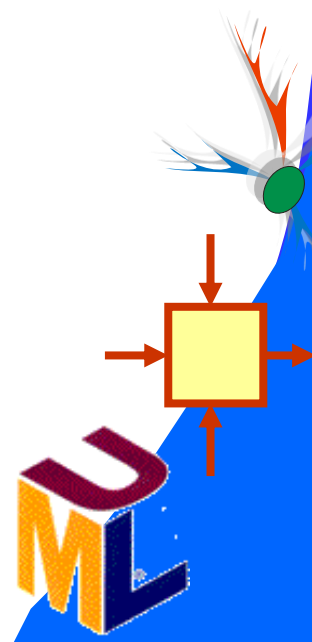
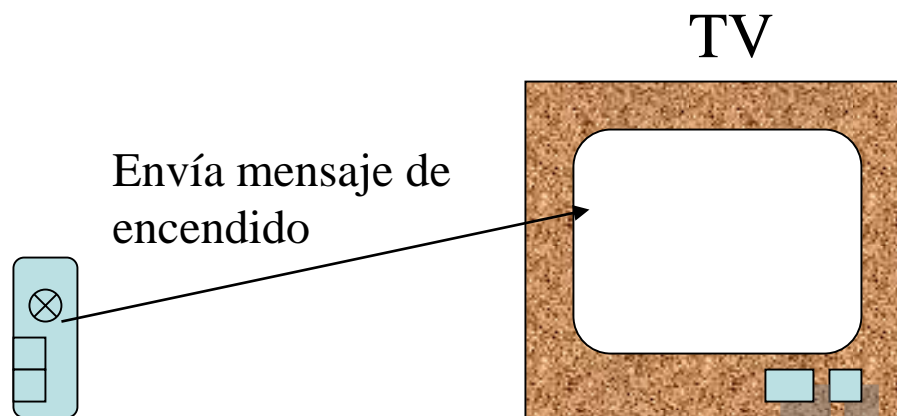


Un receptor (radio)
Caja negra de un avión
Chip
Lapicero
Libro
Reloj
Cerebro
CD
Una bombilla de luz
Maquina de escribir



Envío de Mensajes

Un sistema de Objetos Trabaja en conjunto. Esto se logra mediante el envío de mensajes entre ellos. Un objeto envía a otro un mensaje para realizar una operación, y el objeto receptor ejecutará la operación



Jugador

Pelota

Cocinero

Cocina

Lector

Libro

Chofer

Auto

Profesor

Alumno

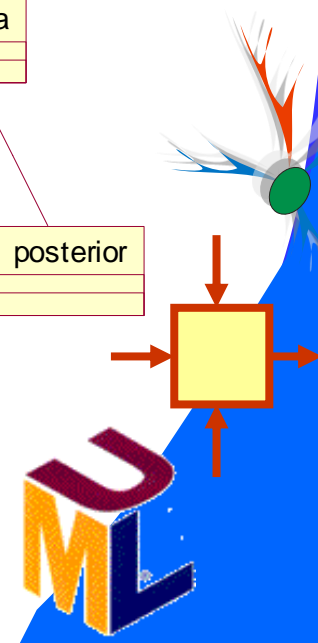
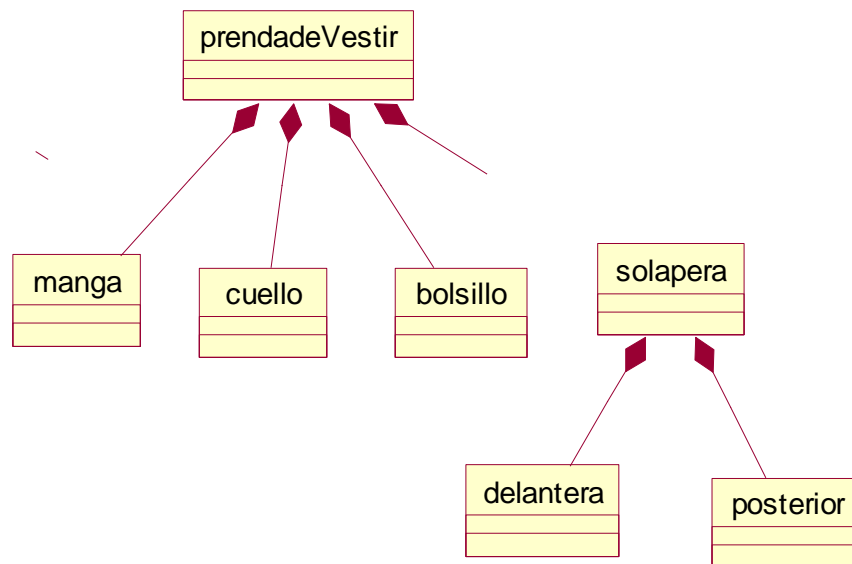
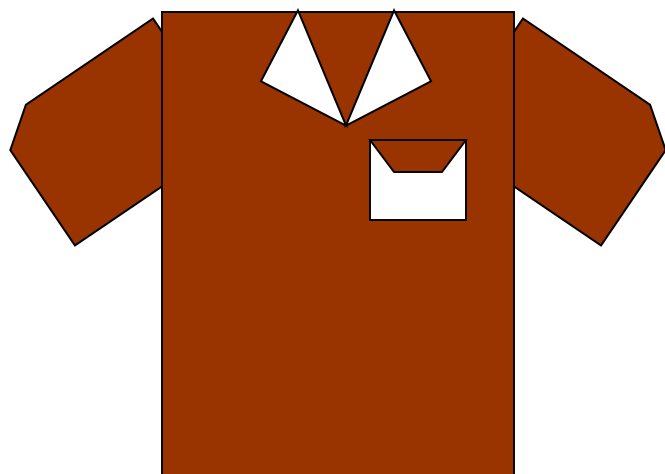
Usuario

Equipo de
Sonido



Composición

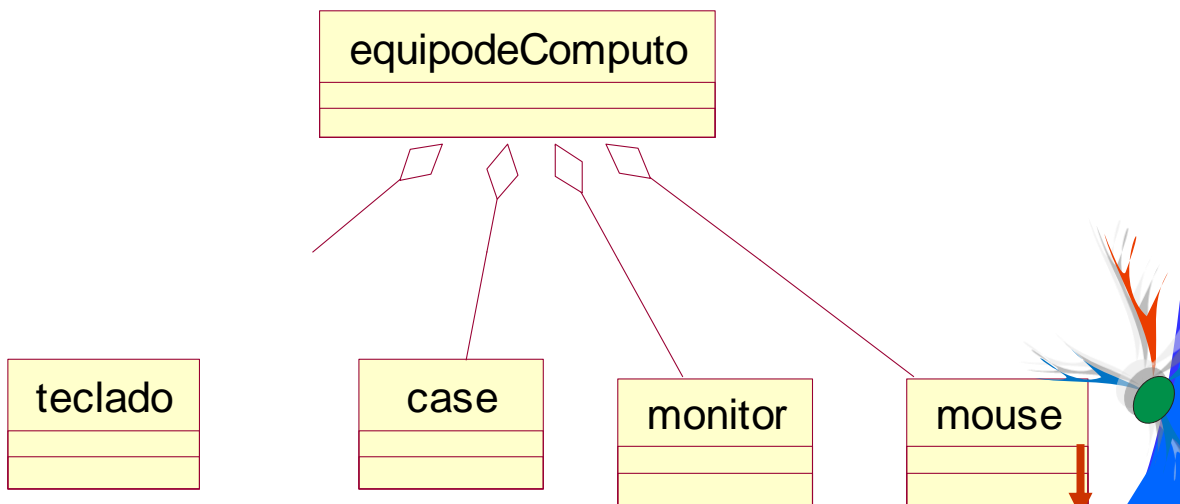
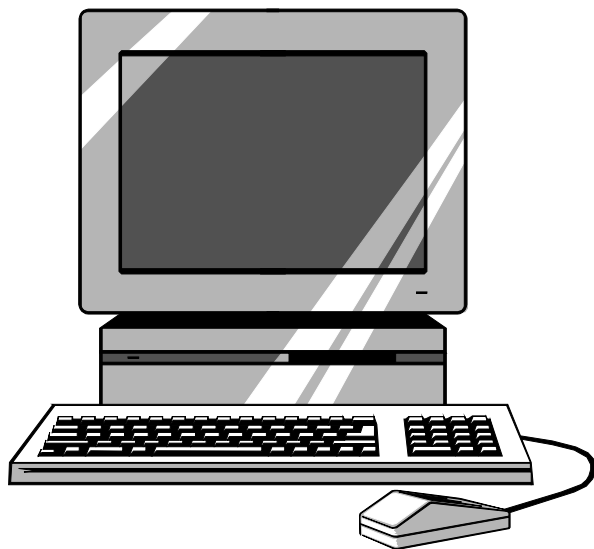
La idea es similar al concepto de agregación, con la diferencia que las partes que conforman el objeto no pueden tener independencia, necesitan estar acoplados





Agregación

Este concepto se aplica cuando las partes que conforman un objeto tienen independendencia para que se puedan acoplar a otro objeto con mucha facilidad.



PC
Carro
Lámpara
Puerta

Mesa
Pantalón

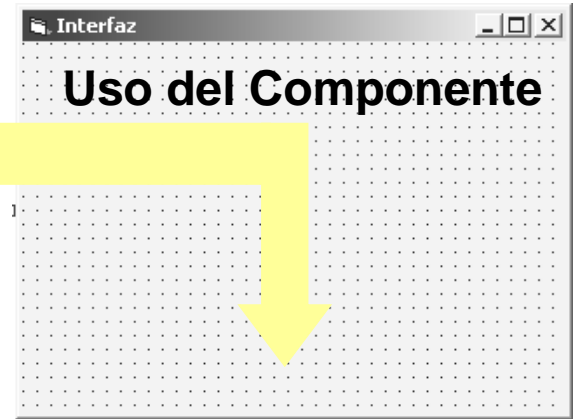
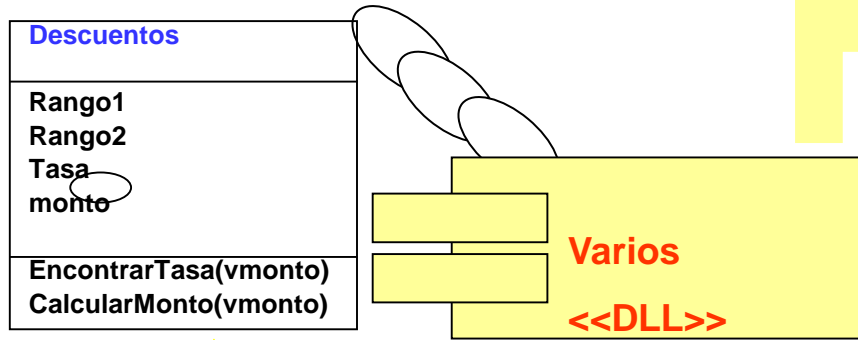
Cabecera de Factura

Detalle

Cliente

Producto

Componentes con Respecto a la Lógica del Negocio



Incluir Varios.dll

Descuentos i_Descuentos

i_Descuentos.EncontrarTasa(200)

Imprimir i_descuentos.tasa

0,05

Interfaz

EncontrarTasa(Vmonto)

Inicio

Numérico Vtasa

Seleccionar Vtasa=tasa de TB_DSTO

Cuando Vmonto>=rango1 y Vmonto<=rango2

Tasa=Vtasa

Fin

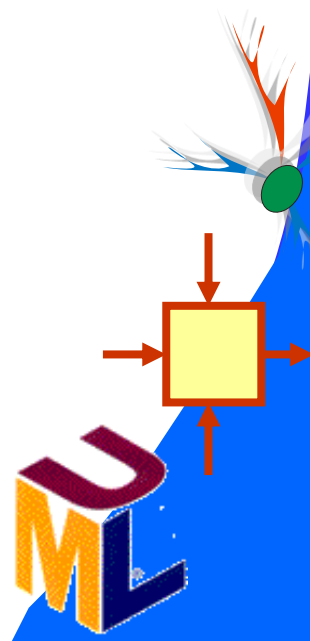
TB_DSTO(Tabla)

i d	rango1	rango2	tasa
1	100	200	0,05
2	201	300	0,06
3	301	400	0,07

Datos



Extensiones UML





■ Extensiones UML

➔ Modelos de Base de Datos Relacionales

■ En la Actualidad Trabajamos con Base de Datos Relacionales, tal como SQL, Oracle etc.

■ Diagramas de Clases vs Modelo E-R

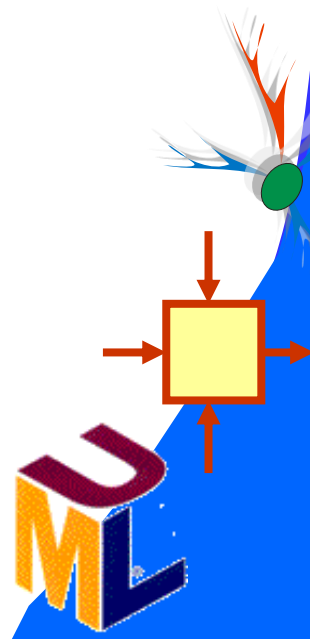




Diagrama Relacional

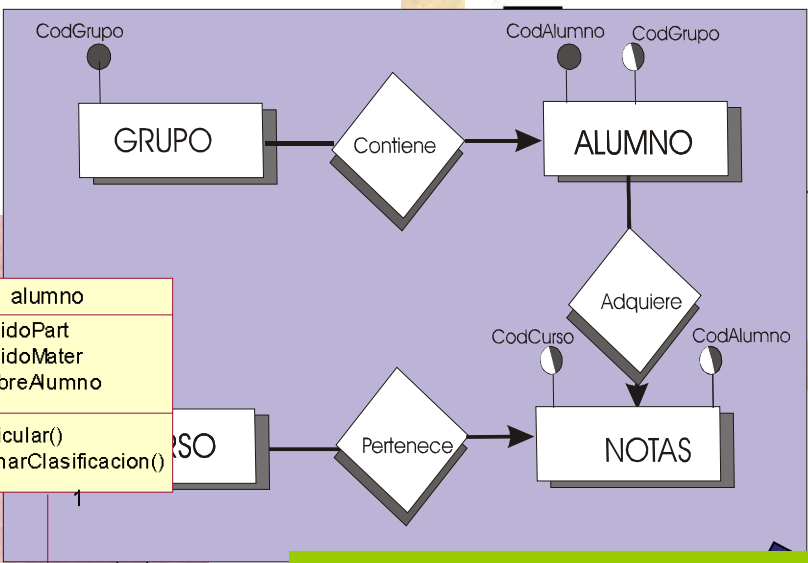
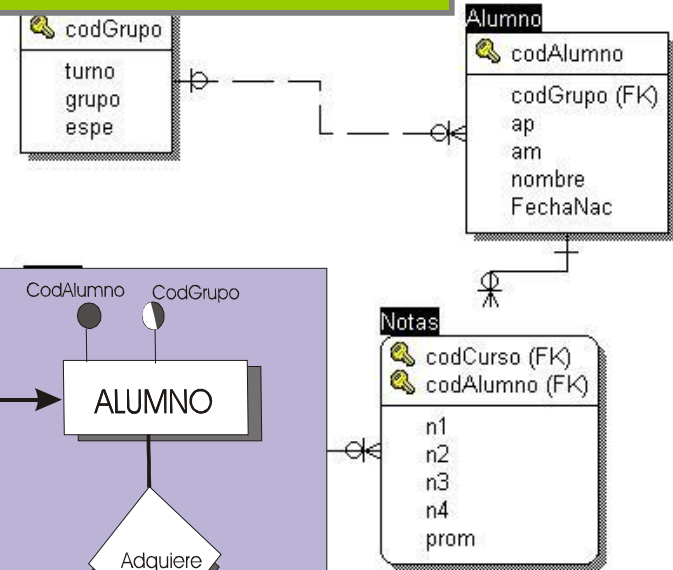


Diagrama E-R

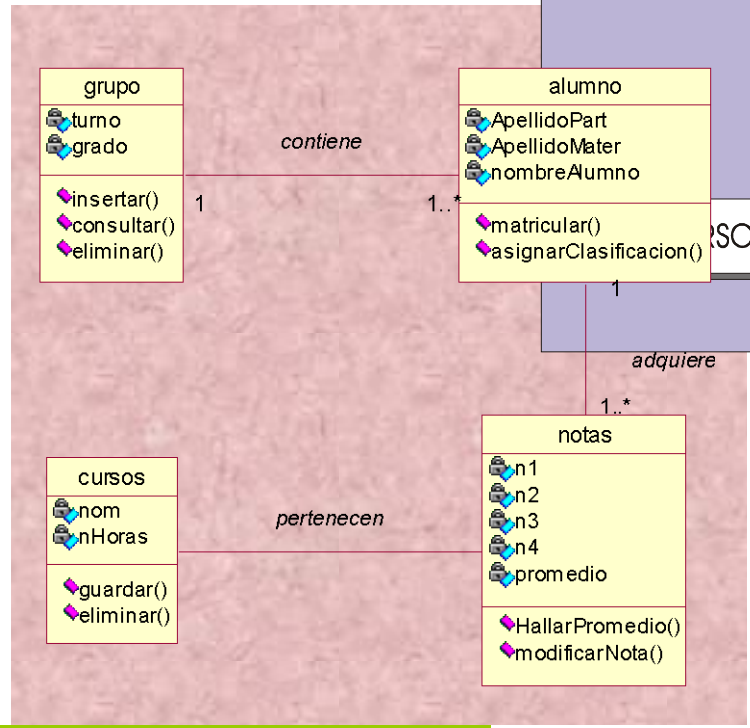
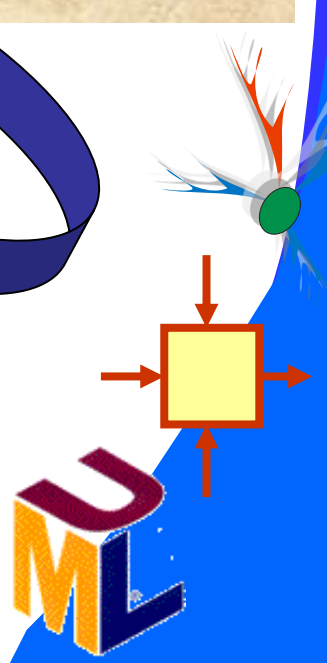


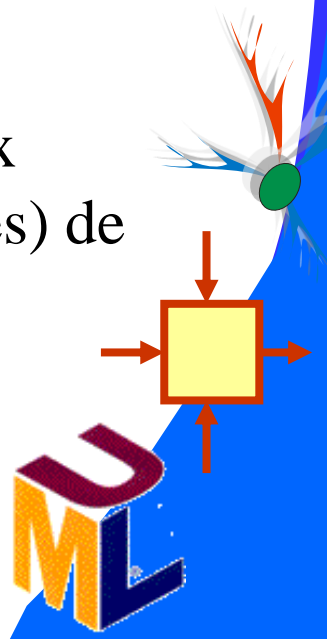
Diagrama de Clases

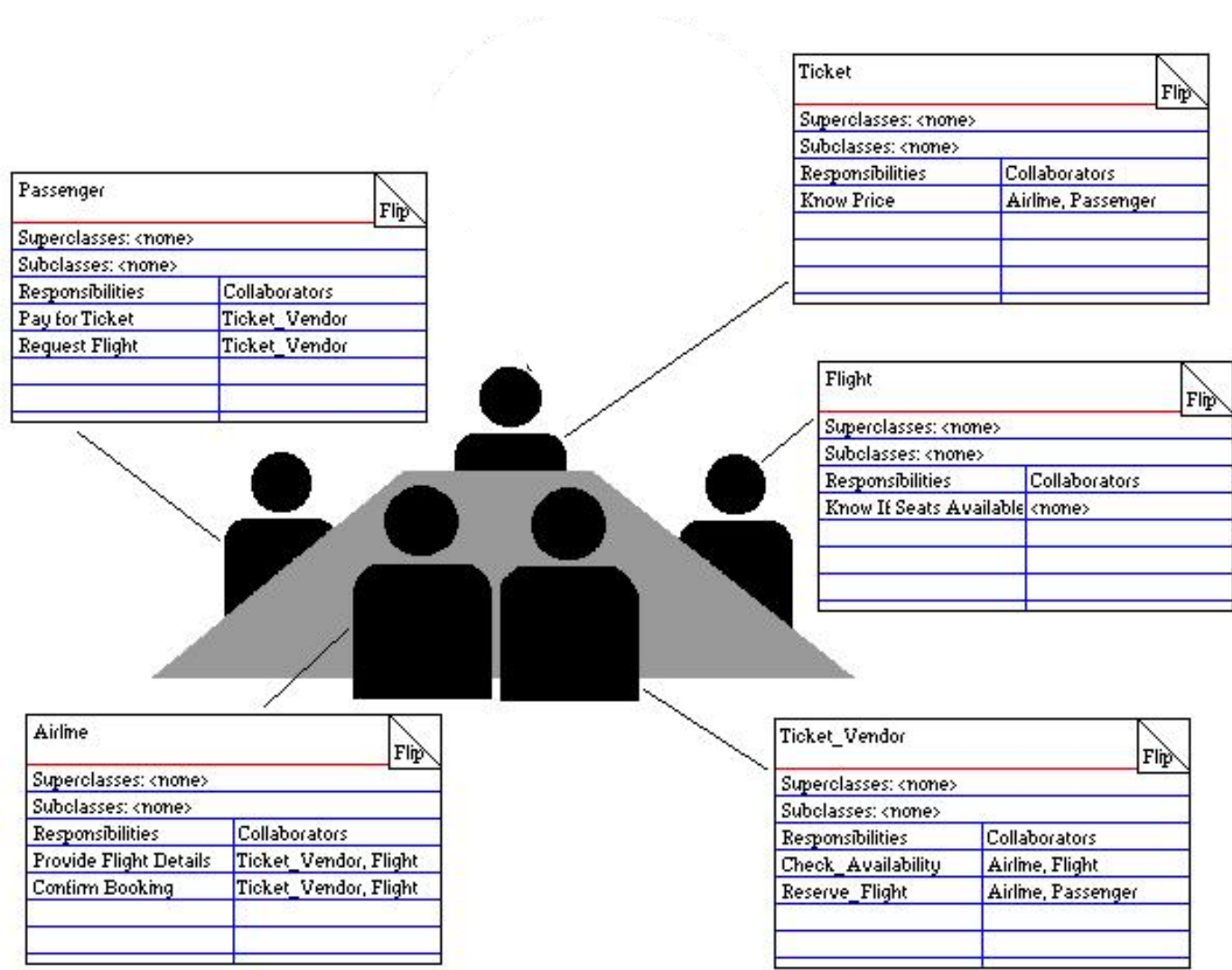




➡ **Tarjetas CRC**

- Se Utiliza para análisis guiados por responsabilidades
- La definición de las clases son refinadas basándose en las responsabilidades de clase y en otras clases que colaboran para completar su responsabilidad
- Cada clase se representa en una tarjeta índice (index card), y los diseñadores establecen los papeles (roles) de las clases en el sistema para definir su trabajo,





Extensión informal de UML -- Tarjetas CRC para análisis guiados por la responsabilidad.

