



# UML

Lenguaje de Modelado  
Unificado



# UML

Concepto de UML (Unified Modeling language)



## Esquema

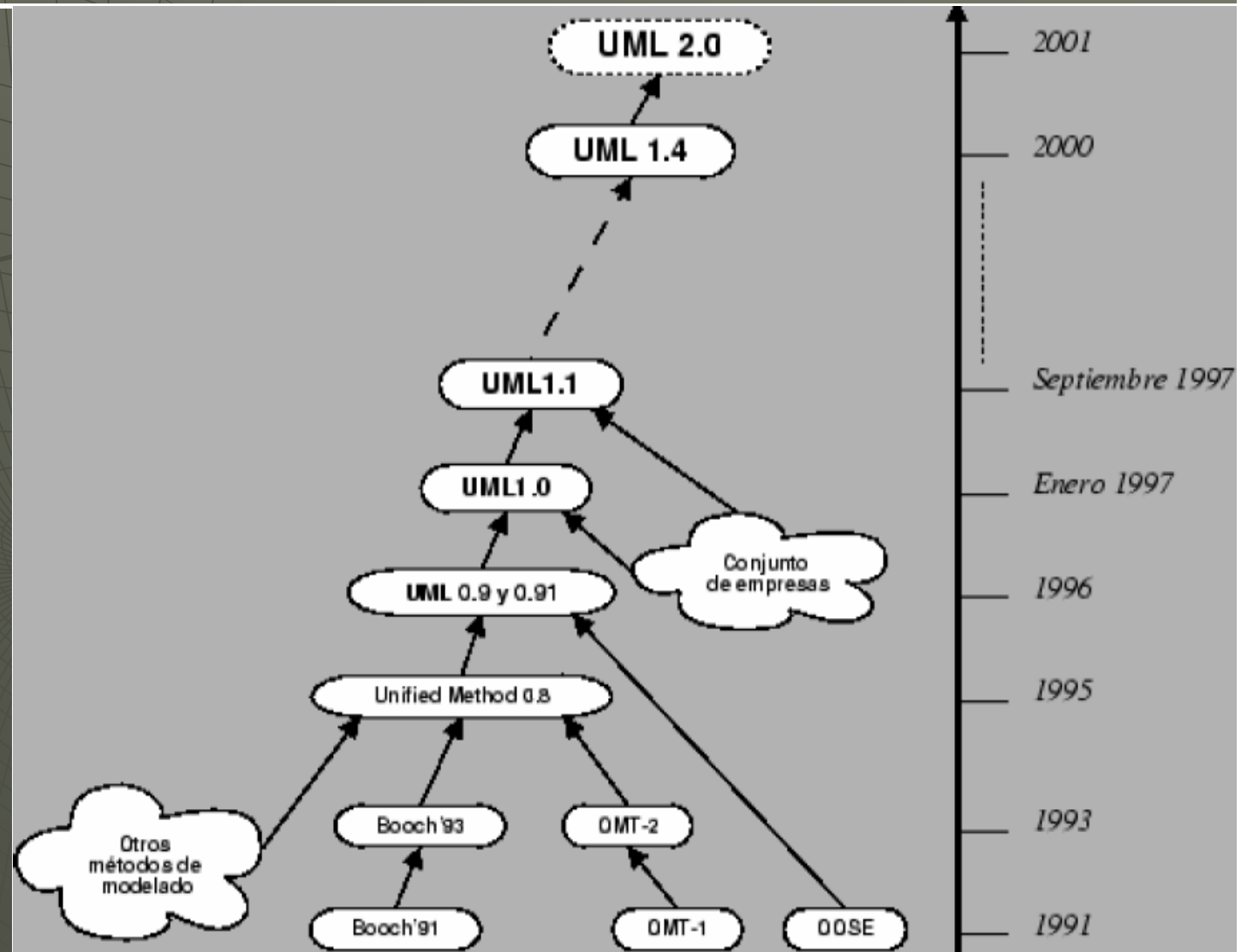
- ◆ **Concepto de UML**
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ Es un lenguaje usado para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software bajo desarrollo, así como para modelado de negocios y almacenamiento de datos.
  - ◆ Los principales factores que motivaron la creación de UML fueron:
    - La necesidad de modelar sistemas.
    - Las tendencias en la industria de software.
    - Unificar los distintos lenguajes y métodos existentes.
    - Innovar los modelos para adaptarse a la arquitectura distribuida.

### Esquema

- ◆ Concepto de UML
  - ◆ **Reseña Histórica**
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ Comenzó como una iniciativa de Grady Booch y Jim Rumbaugh en 1994 para combinar las notaciones visuales de sus conceptos (Booch y OMT).
  - ◆ En 1995 Ivar Jacobson, creador del OOSE se unió al proyecto.
  - ◆ En 1997 la versión 1.0 de UML fue propuesta como metamodelo orientado a objetos de semántica y notación estándares.
  - ◆ En 1997 como estándar por el OMG (Object Management Group).
  - ◆ Para el 99 se hablaba de la versión 1.2 y para el 2002 de UML 1.4 y 2.0, la cual es la ultima versión aprobada

### Esquema

- ◆ Concepto de UML
- ◆ **Reseña Histórica**
- ◆ Características
- ◆ Estándares que conforman UML
- ◆ Diagramas UML
- ◆ Modelo Relacional con UML
- ◆ Ventajas
- ◆ Críticas





### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ **Características**
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ Es una especificación basada en Booch, OMT y OOSE, de allí sus principios.
  - ◆ Divide cada proyecto en un numero de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
  - ◆ Permite describir un sistema en diferentes niveles de abstracción.
  - ◆ Se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del desarrollo de una aplicación, sin definir un modelo de desarrollo.



# UML



## Estándares que Conforman UML

### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ **Estándares que conforman UML**
  - ◆ Diagramas de UML
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ **Superestructura:** Es aquí dónde se definen los diagramas y los elementos que los componen.
  - ◆ **Infraestructura:** Conceptos de bajo nivel. Meta-Modelo da soporte a la superestructura, entre otras.
  - ◆ **OCL:** Lenguaje de restricción. De utilidad para especificar conceptos ambiguos sobre los distintos elementos del diagrama.
  - ◆ **XMI / Intercambio de diagramas:** Permite compartir diagramas entre diferentes herramientas de modelado UML.

### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ **Diagramas UML**
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ **Diagrama de Clases**, modela la estructura estática de las clases en el sistema.
  - ◆ **Diagrama de Componentes**, modela los componentes que componen una aplicación, sistema o empresa.
  - ◆ **Diagrama de Estructura de Composición**, Representa la estructura interna de un clasificador.
  - ◆ **Diagrama de Despliegue Físico**, muestra cómo y dónde se desplegará el sistema.
  - ◆ **Diagrama de Objetos**, modela la estructura estática de los objetos en el sistema.
  - ◆ **Diagrama de Actividades**, modela el comportamiento de los casos de uso, objetos operaciones.



### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ **Diagramas UML**
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ Críticas
- ◆ **Diagrama de Comunicaciones**, modela interacciones entre objetos.
  - ◆ **Diagrama de Secuencias**, representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian.
  - ◆ **Diagrama de Máquinas de Estado**, ilustra como un elemento se puede mover entre estados que clasifican su comportamiento.
  - ◆ **Diagrama de Tiempos**, mostrar los cambios en el estado o la condición de una línea de vida a lo largo del tiempo lineal.
  - ◆ **Diagrama de Casos de Uso**, muestra las relaciones entre los actores y el sujeto (sistema), y los casos de uso.



### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ **Modelo Relacional con UML**
  - ◆ Ventajas
  - ◆ Críticas
- ◆ Los diagramas de Clases de UML se pueden usar para modelar la base de datos relacional en la que un sistema este basado, sin embargo los diagramas tradicionales de modelado capturan mas información sobre la base de datos relacional y son mas adecuados para modelarla.
  - ◆ Como extensión de UML, el diagrama de clases puede ser referenciado en un diagrama de relación de entidad (ER diagram) el cual relaciona entidades que pueden ser modeladas basadas en atributos clave

### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ Modelo Relacional con UML
  - ◆ **Ventajas**
  - ◆ Críticas
- ◆ Diseño y documentación.
  - ◆ Código reutilizable.
  - ◆ Descubrimiento de fallas.
  - ◆ Ahorro de tiempo en el desarrollo del software.
  - ◆ Mucho más fáciles las modificaciones.
  - ◆ Más fácil comunicación entre programadores.

### Esquema

- ◆ Concepto de UML
  - ◆ Reseña Histórica
  - ◆ Características
  - ◆ Estándares que conforman UML
  - ◆ Diagramas UML
  - ◆ Modelo Relacional con UML
  - ◆ Ventajas
  - ◆ **Críticas**
- ◆ UML es excesivamente extenso y complejo.
  - ◆ Contiene diagramas y estructuras que son redundantes o muy poco usadas.
  - ◆ Semántica imprecisa.
  - ◆ Problemas para el aprendizaje.
  - ◆ Solo el código está en sincronización con el código.
  - ◆ Trata de ser todas las cosas para todas las personas.

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

- ◆ Entidad → Clase
- ◆ Atributo → Atributo
- ◆ Relación → asociación de clases
- ◆ Variación en las notaciones para las cardinalidades de las relaciones (asociaciones)





# UML

## Elementos Comunes a Todos los Diagramas



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Clase

es la unidad básica que encapsula toda la información de un objeto. A través de ella podemos modelar el entorno en estudio.

En UML, una clase es representada por un rectángulo que posee tres divisiones.

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

Nombre de la Clase

Lista de atributos

Lista de Métodos u Operaciones

En donde:

Superior: Contiene el nombre de la Clase

Intermedio: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

Inferior: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Atributos:

- ◆ Representa alguna propiedad de la clase
- ◆ pueden representarse mostrando su nombre y su tipo e incluso su valor por defecto.
- ◆ Los atributos definen la estructura de una clase y de sus correspondientes objetos.
- ◆ El atributo define el valor de un dato para todos los objetos pertenecientes a una clase.

Ejemplo: nombre, edad, peso son atributos de la clase persona

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Elementos Derivados:

Un elemento derivado es aquel cuyo valor se puede calcular a partir de otros elementos presentes en el modelo, pero que se incluye en el modelo por motivos de claridad o como decisión de diseño. Se representa con una barra "/" precediendo al nombre del elemento derivado.

```
{edad= fechaAct - fechaNac}
```

Persona
nombre
fechaNac
/edad





# UML

## Elementos Comunes a Todos los Diagramas



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Asociaciones:

Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los más importantes de entre dichos elementos gráficos.

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Nombre de la Asociación y Dirección

El nombre de la asociación es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo negro sólido que indique la dirección en la cual leer el nombre de la asociación.



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Multiplicidad

La multiplicidad es una restricción que se pone a una asociación. Puede expresarse de las siguientes formas:

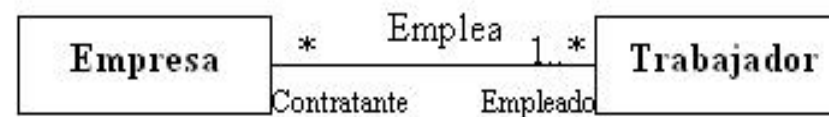
- Con un número fijo: 1.
- Con un intervalo de valores: 2..5.
- Con un rango en el cual uno de los extremos es un asterisco. Significa que es un intervalo abierto. Por ejemplo, 2..\* significa 2 o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7, 15..\*.
- Con un asterisco: \*. En este caso indica que puede tomar cualquier valor (cero o más).

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Roles

Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol.



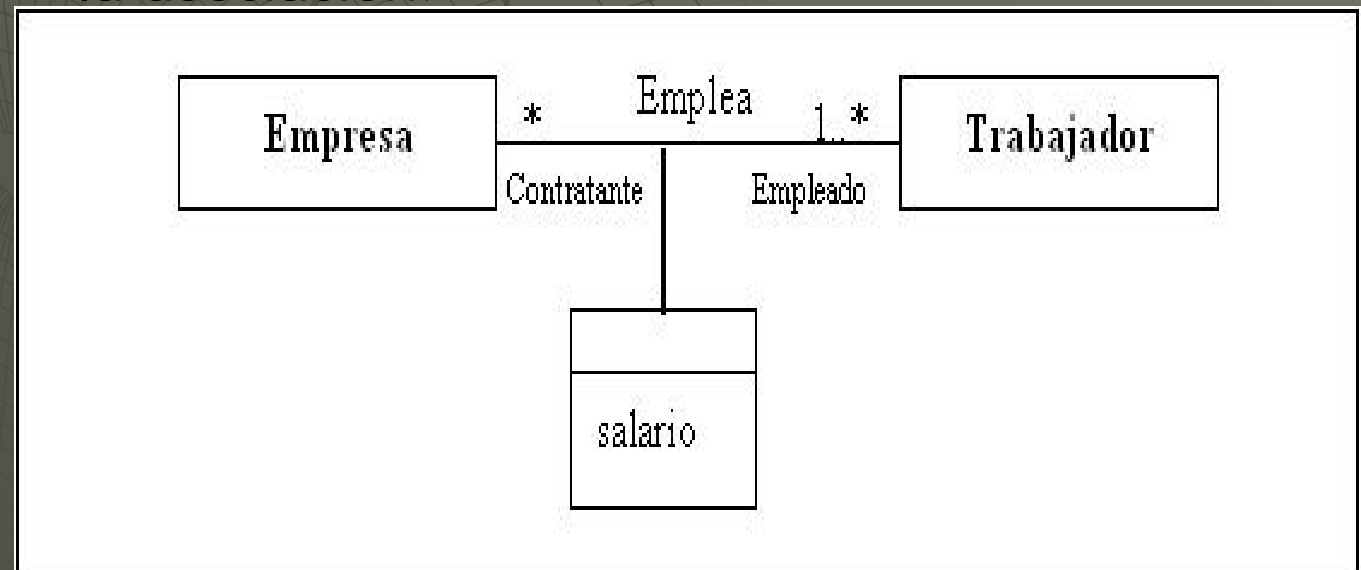


### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Clases Asociación

Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea a trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación. Por tanto ambos tienen el mismo nombre, el de la asociación.

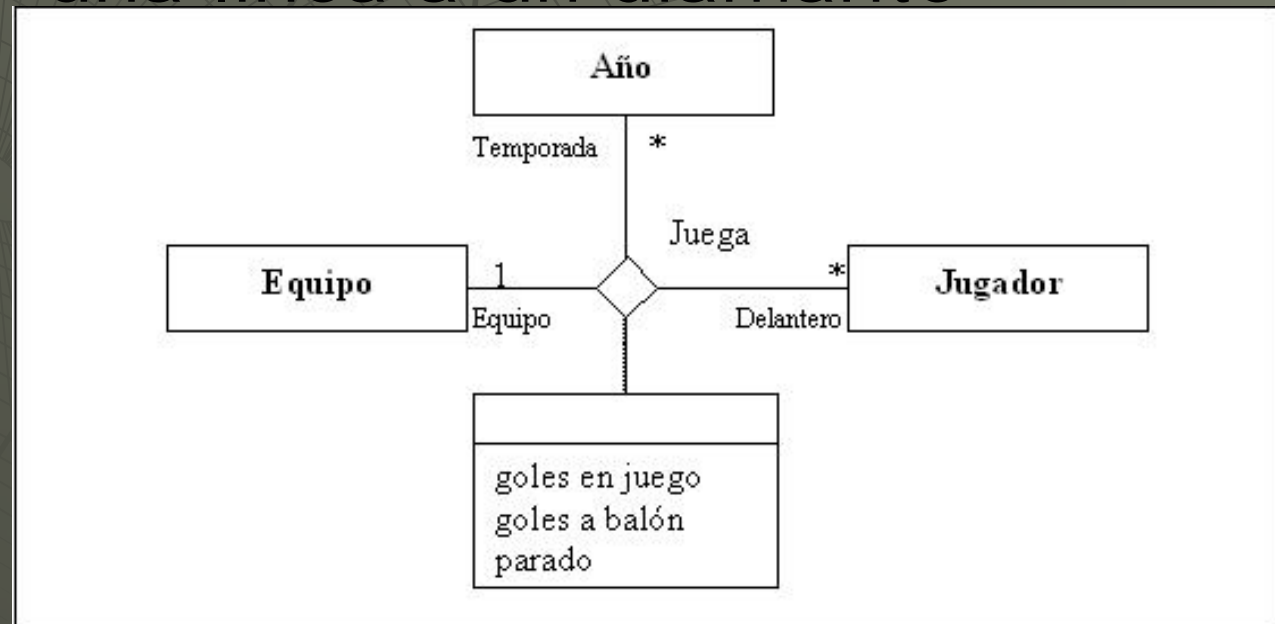


### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
- Clase
- Atributos
- Asociaciones
- Herencia
- Ejemplos

### ◆ Asociaciones N-Arias

En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Ensamblados

Son formas especiales de asociación entre un todo y sus partes, en donde el ensamblado está compuesto por sus componentes.

### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Agregación

El objeto base utiliza al incluido para su funcionamiento. Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye.



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Composición

El Objeto base se construye a partir del objeto incluido. Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye.



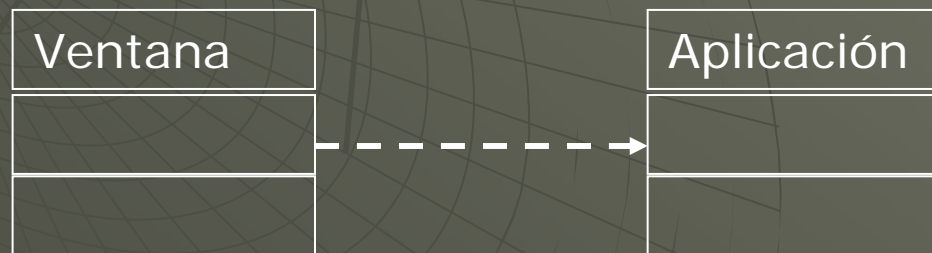
### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### ◆ Dependencia o Instanciación - - - - - →

Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase).

El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra, como por ejemplo una aplicación gráfica que instancia una ventana (la creación del Objeto Ventana esta condicionado a la instanciación proveniente desde el objeto Aplicación):



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Herencia

(especialización/generalización)

- ◆ La herencia es una abstracción importante para compartir similitudes entre clases, donde todos los atributos y operaciones comunes a varias clases se pueden compartir por medio de la superclase.
- ◆ Las clases con atributos y operaciones comunes se pueden organizar de forma jerárquica, mediante la herencia. Como modelo conceptual da buena estructura a las clases, como modelo de implementación es un buen vehículo para no replicar información innecesariamente.



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

### Herencia

(especialización/generalización)

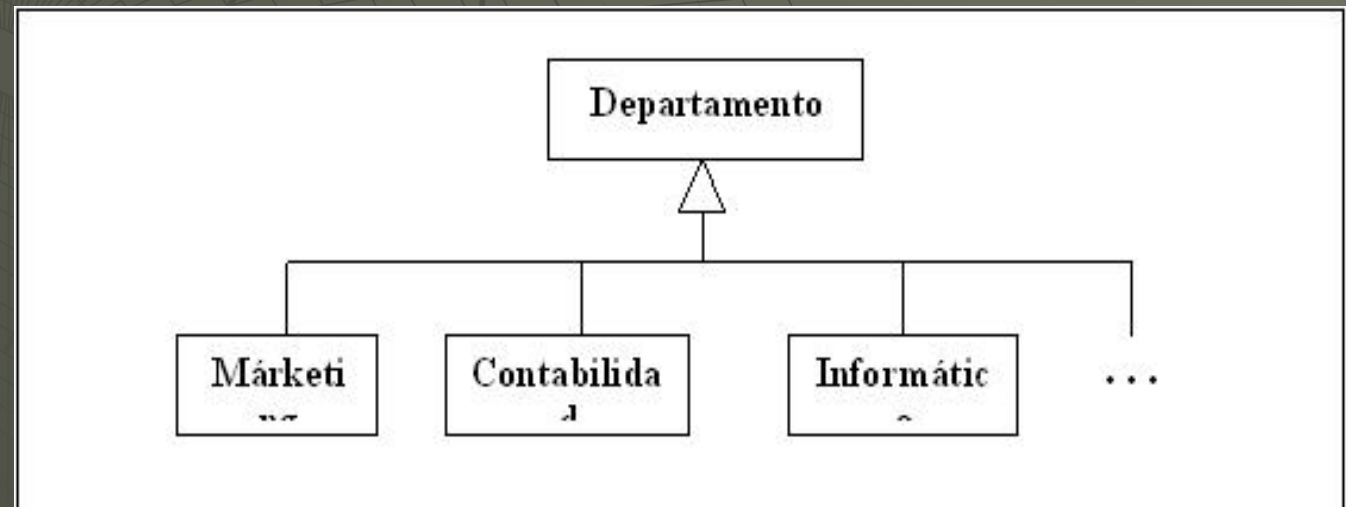
- ◆ La Herencia es útil para el modelo conceptual al igual que para la implementación.
- ◆ La superclase generaliza a sus subclases, y las subclases especializan a la superclase.
- ◆ Cada clase tiene sus propios atributos los cuales se van especializando a medida que las clases son cada vez mas especializadas.



### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos

- ◆ La herencia indica que una subclase hereda los métodos y atributos especificados por una Súper Clase, por ende la sub – clase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la súper clase



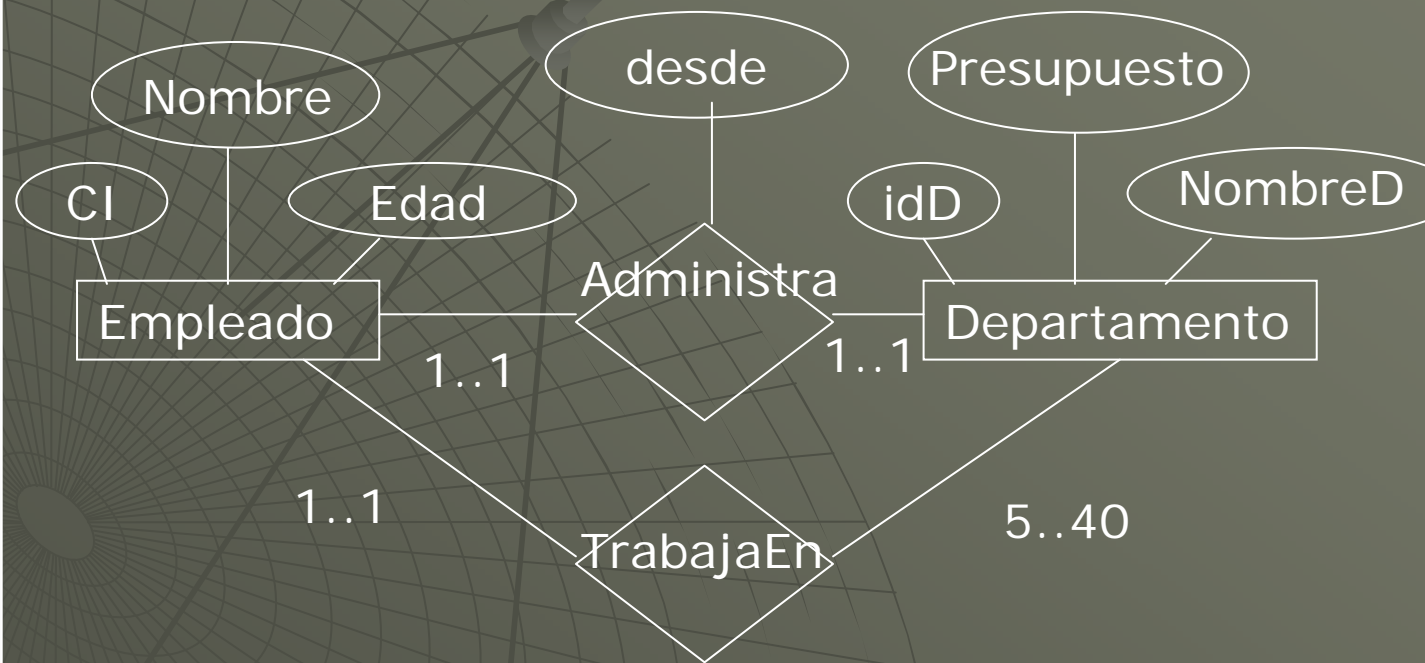
### Esquema

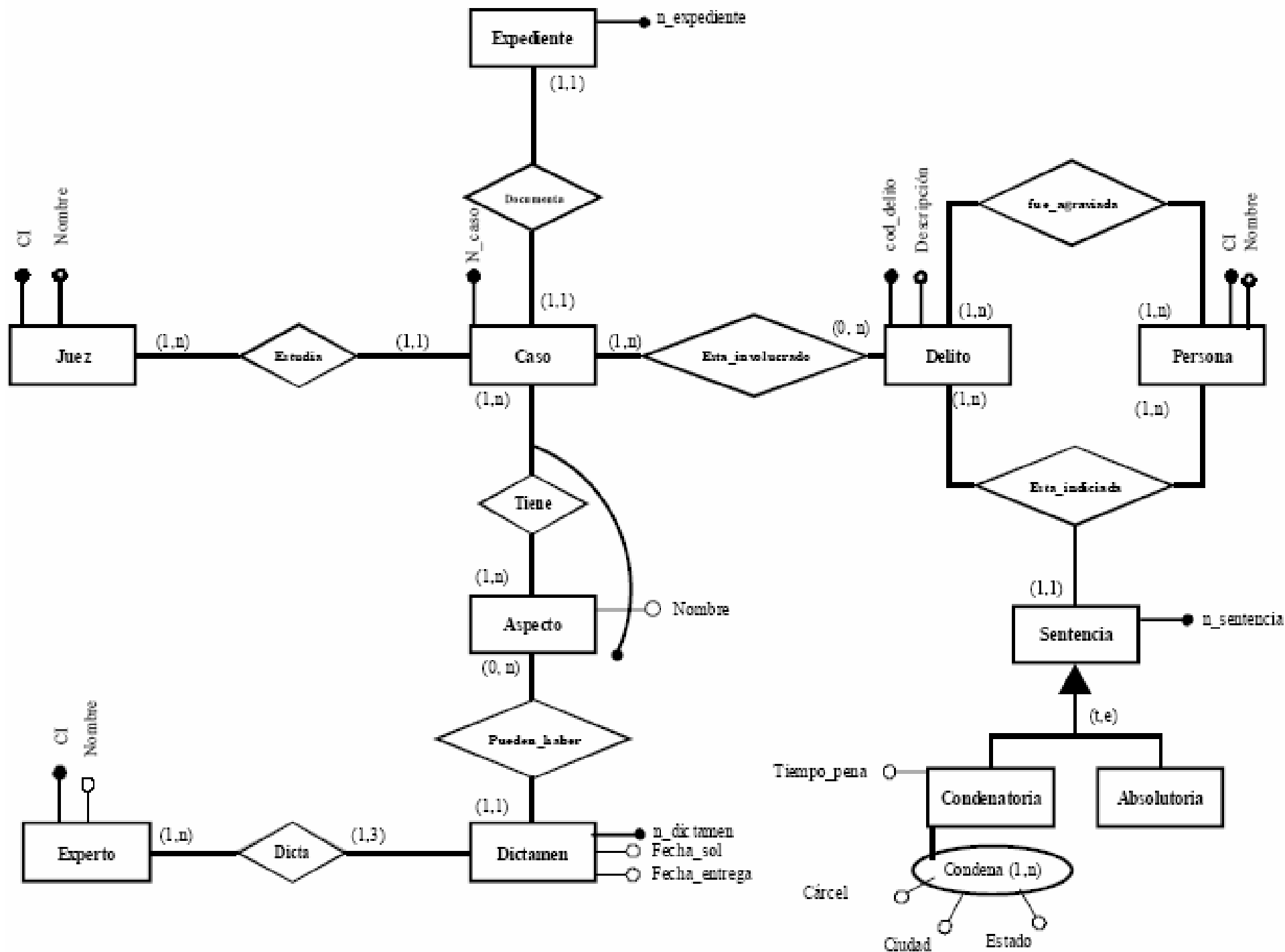
- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos



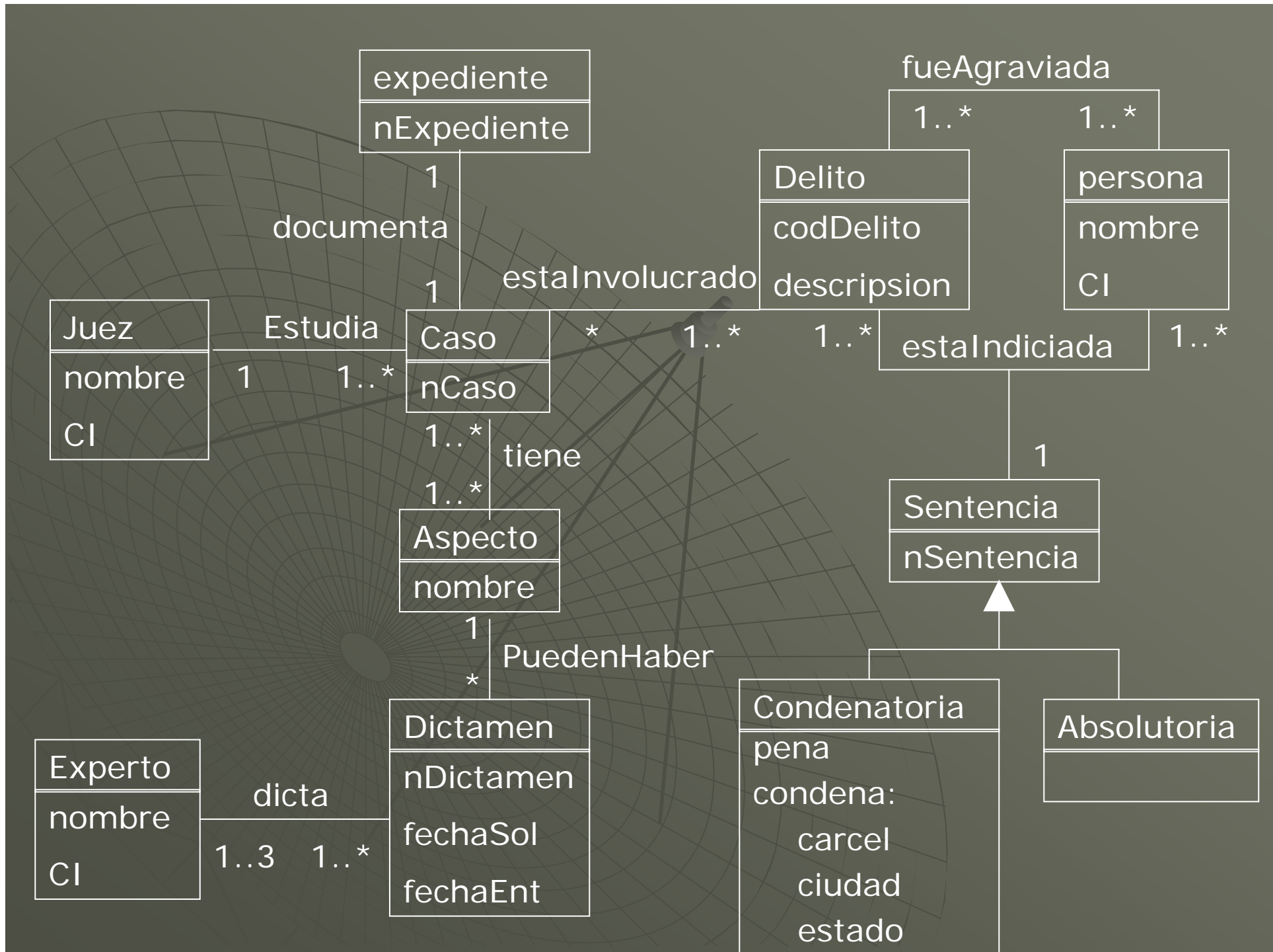
### Esquema

- Correspondencia de términos ER-a-UML
- Elementos Comunes a Todos los Diagramas
  - Clase
  - Atributos
  - Asociaciones
  - Herencia
- Ejemplos









# Referencias Electrónicas

- ◆ <http://www.clikear.com/manuales/uml/>
- ◆ <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/>
- ◆ [http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado#column-one](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado#column-one)
- ◆ [http://www.epidataconsulting.com/tikiwiki/tiki-read\\_article.php?articleId=15](http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15)