

## **Capítulo 3**

# **Modelado de datos con herramientas CASE**

**Al finalizar el capítulo, el alumno podrá:**

- Identificar las bondades de una herramienta CASE.
- Construir modelos bajo casuísticas reales.

**Temas:**

1. Herramientas CASE
2. Tipos de herramienta CASE
3. Ventajas y desventajas
4. ER/One Data Modeler Tool

## 1. Herramientas CASE

## Herramientas CASE

**A nivel conceptual:**

- En los 70s, un proyecto llamado ISDOS analizaba la relación entre los requisitos de un problema y las necesidades que este generaba, ayudando a encontrar soluciones a los diseñadores.



**A nivel técnico:**

- En 1984 salió el primer CASE con el nombre Excelerator para PC.
- En los 90s, IBM consiguió una alianza con AD/Cycle y juntos desarrollaron un CASE que permitía supervisar todo el ciclo de vida del desarrollo del software.

3 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



### 1.1 Definición de herramienta CASE

En la actualidad, se manifiesta una tendencia hacia un mundo heterogéneo en el cual puedan convivir diversos productos que se complementen y en ese contexto, contar con herramientas de desarrollo abiertas, con conectividad a diversas plataformas, basadas en tecnología, orientada a objetos y que permitan la reutilización del software. De este modo, la mayoría de las empresas se han extendido a la adquisición de herramientas **CASE** (Computer Aided Software Engineering), con el fin de automatizar los aspectos clave de todo lo que implica el proceso de desarrollo de un sistema e incrementar su posición en el mercado competitivo. Sin embargo, en ocasiones, se obtienen elevados costos tanto en la adquisición de herramientas y entrenamiento del personal, como en la falta de adaptación de tal herramienta a la arquitectura de la información y a metodologías de desarrollo, utilizadas por la organización.

Por otra parte, algunas herramientas CASE no ofrecen o evalúan soluciones potenciales para los problemas relacionados con sistemas o virtualmente, no llevan a cabo ningún análisis de los requerimientos de la aplicación. Sin embargo, CASE proporciona un conjunto de herramientas semi-automatizadas y automatizadas que están desarrollando una cultura de ingeniería nueva para muchas empresas. Uno de los objetivos a largo plazo, más importante del CASE es conseguir la generación automática de programas desde una especificación al nivel de diseño.

Existe libertad de seleccionar herramientas de desarrollo que están disponibles en el mercado, de acuerdo con gustos y necesidades; por ejemplo: Cool Stuf, Erwin, Informix NewEra, Lotus Notes, Domino, Power Builder, Opal, Java, Oracle Developer 2000, Visual Basic .NET y Visual Age de IBM. Si bien cada una de ellas cuenta con un valor diferenciador, en todas está presente la tecnología abierta, una búsqueda de conectividad cada vez mayor, la simplicidad en su uso y sin lugar a dudas, la velocidad para desarrollar. La mayoría está basada en tecnología de la

programación orientada a objetos o soporta algunas de las reglas de su creador, Edgar F. Codd. Además, generan aplicaciones para tecnologías que en la actualidad marcan la pauta a seguir, como Sistemas distribuidos, Cliente/Servidor, aplicaciones WEB, entre otras.

Los ingenieros del software reconocen que necesitan más herramientas variadas (las herramientas manuales no satisfacen las demandas actuales de los sistemas basados en computadoras): además, requieren un taller organizado y eficiente, en el cual puedan ubicar sus herramientas. El taller de ingeniería del software se establece como un entorno de apoyo de proyectos integrados y el conjunto de herramientas que llena ese taller se denomina **ingeniería del software asistida por computadora (CASE)**.

## 1.2 Herramientas CASE orientadas a bases de datos: ER/One Data Modeler Tool

**ER/One Data Modeler Tool** permite a los desarrolladores diseñar bases de datos gráficas mediante el uso de modelos físicos y diagramas de relación de entidades para generar automáticamente las bases de datos SQL más populares. También, permite a los desarrolladores implementar más fácilmente las bases de datos al ofrecer un entorno sofisticado de modelado de datos visuales.

El ER/One Data Modeler Tool reduce los errores en el desarrollo de bases de datos a la vez que mejora la productividad y simplifica el modelado de datos. Los desarrolladores pueden visualizar estructuras de modelos físicos para comprender sus bases de datos, crear nuevas bases de datos, modificar, analizar y optimizar la solución.

El ER/One Data Modeler Tool, también, permite:

- Generación de modelos de datos físicos.
- Administrar un número ilimitado de tablas.
- Creación de índices, claves, disparadores, procedimientos almacenados, vistas y tipos de datos definidos por el usuario.
- Generación de un script para los motores de bases de datos Oracle, MS SQL Server, PostgreSQL, MySQL, Interbase y Firebird.

Las características principales del ER/One Data Modeler Tool son:

- Modelos de datos específicos de DBMS
- Gran gestión de modelos
- Definir entidades, atributos, dominios, índices y relaciones
- Modelado de datos visuales
- Navegador de objetos (vista de árbol)
- Verificación de modelo
- Generación de script DDL
- Gestión de opciones
- Disparadores
- Definir vistas y procedimientos
- Exportar el modelo como una imagen

## 2. Tipos de herramienta CASE


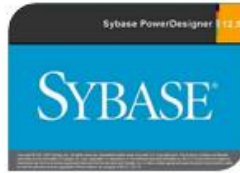
### Tipos de herramientas CASE


**Cases Upper:**

- Apoya durante el proceso de planificación, análisis de requisitos y estrategia del desarrollo de aplicaciones.

**Cases Middle:**

- Decide si la información de un fenómeno o proceso del mundo real debe almacenarse en una base de datos, se le conoce como: campo o atributo.

3 - 6
Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.


En los años 70s un proyecto llamado **ISDOS** diseñó un lenguaje que analizaba la relación existente entre los requisitos de un problema y las necesidades que estos generaban, el lenguaje en cuestión se denominaba **PSL (Problem Statement Language)** y la aplicación que ayudaba a buscar las necesidades de los diseñadores, **PSA (Problem Statement Analyzer)**. Aunque esos son los inicios de las herramientas informáticas que ayudan a crear nuevos proyectos informáticos, la primera herramienta CASE fue Excelerator, que salió a la luz en el año 1984 y trabajaba bajo una plataforma PC.


Las herramientas CASE alcanzaron su techo a principios de los años 90. En la época en la que IBM había conseguido una alianza con la empresa de software **AD/Cycle** para trabajar con sus mainframes. Pero poco a poco, los mainframes han sido menos utilizados y actualmente, el mercado de las Big CASE ha muerto completamente, abriendo el mercado a diversas herramientas más específicas, para cada fase del ciclo de vida del software. Dichas herramientas son descritas a continuación.

- **Upper CASE (U-CASE)**, herramientas que ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando diagramas UML, entre otros.
- **Middle CASE (M-CASE)**, herramientas para automatizar tareas en el análisis y diseño de la aplicación.
- **Lower CASE (L-CASE)**, herramientas que semi-automatizan la generación de código, crean programas de detección de errores, soportan la depuración de programas y pruebas. Además, automatizan la documentación completa de la aplicación, pueden incluirse las herramientas de desarrollo rápido de aplicaciones.

### 3. Ventajas y desventajas

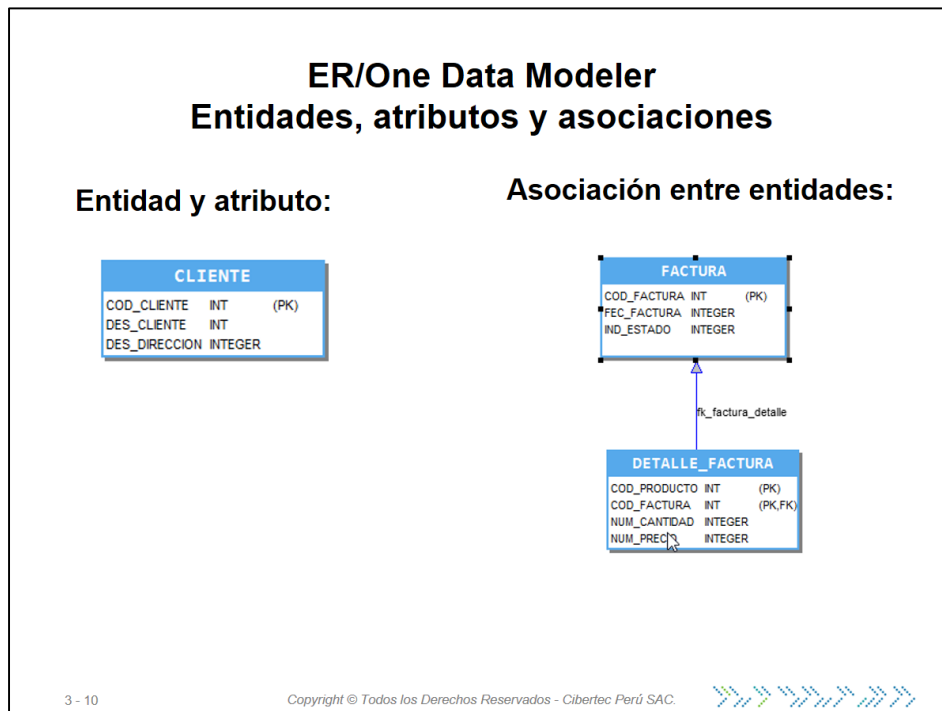
#### Ventajas y desventajas

<b>Ventajas:</b>	<b>Desventaja:</b>
<ul style="list-style-type: none"><li>• Mejora la productividad.</li><li>• Aumenta la calidad y reduce los costos y tiempos.</li><li>• Mejora la planificación del proyecto.</li><li>• Automatiza las fases del ciclo de vida del software.</li><li>• Portable y estandariza el trabajo entre proyectos.</li></ul>	<ul style="list-style-type: none"><li>• Incompatibilidad entre cases.</li><li>• Entrenamiento del personal.</li><li>• Funcionalidad limitada entre cases, no abarca todas las fases.</li><li>• Alto costo de licenciamiento.</li><li>• Hardware.</li><li>• Tiempo de aprendizaje.</li></ul>

3 - 9Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

- Mejora la productividad del desarrollo y mantenimiento del software, aumentando su calidad, y reduciendo los costos y tiempos para la implementación del proyecto.
- Mejora la planificación del proyecto dándole soporte a todas las fases del ciclo de vida del software.
- Reutilización del software, portabilidad y estandarización entre los proyectos.
- Incompatibilidad entre herramientas CASE.
- Entrenamiento del personal especializado.
- Funcionalidad limitada de algunos CASE.
- Costo de licenciamiento.
- Requerimientos de hardware.
- Tiempo de aprendizaje.

#### 4. ER/One Data Modeler Tool

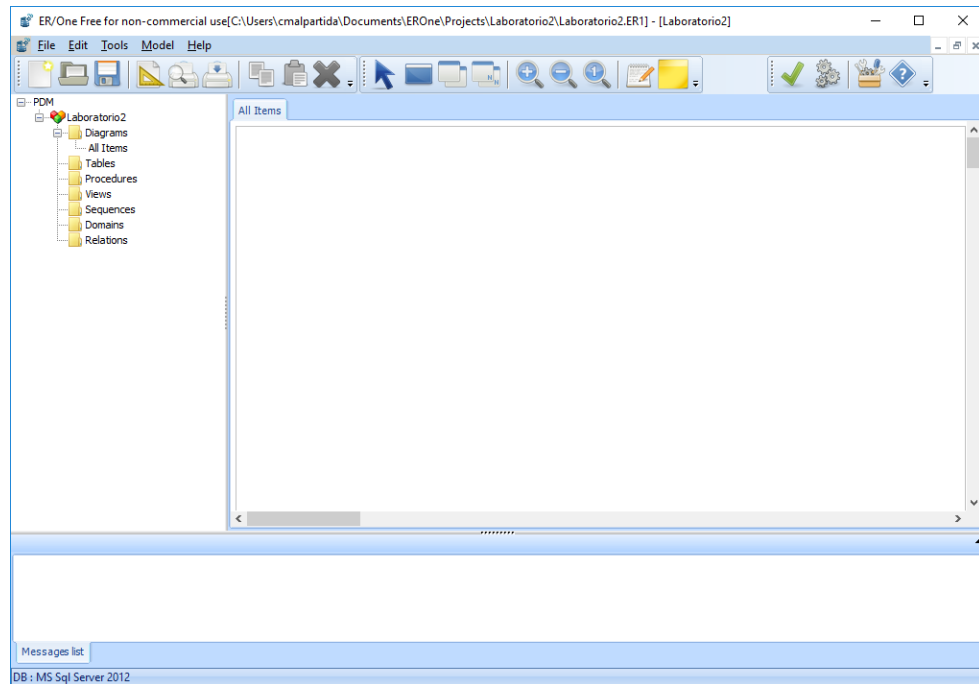


## PASO A: Iniciando el programa.

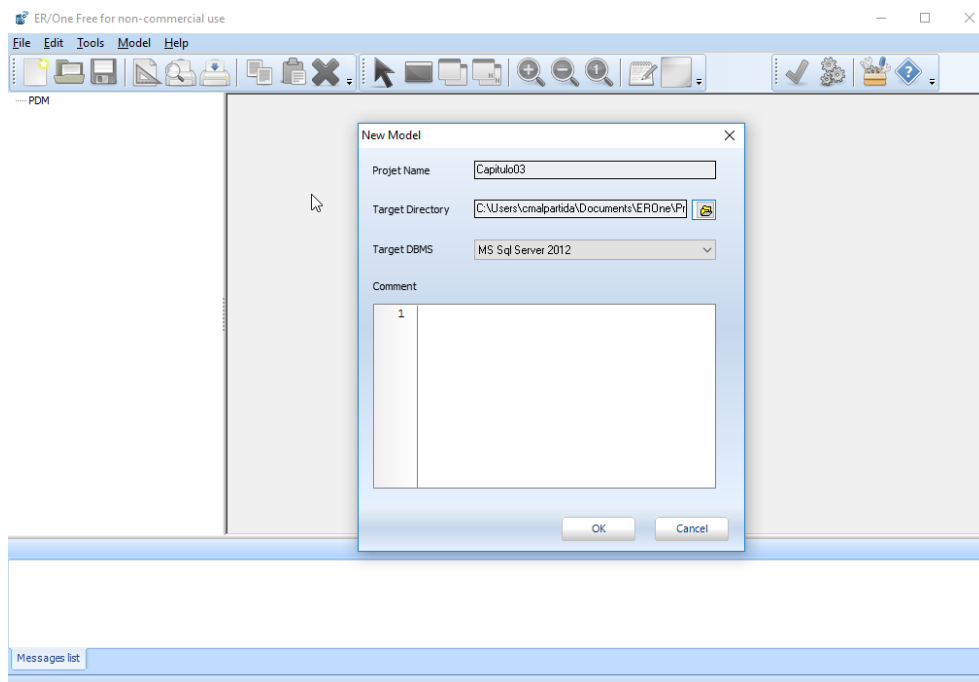
10. En escritorio de Windows, buscar el ícono correspondiente a **ER/One Data Modeler Tool**.



- 2º. Al ingresar, aparecerá la siguiente pantalla. Seleccionar en la barra de **Menú, File, New Project**, como se muestra en la imagen.



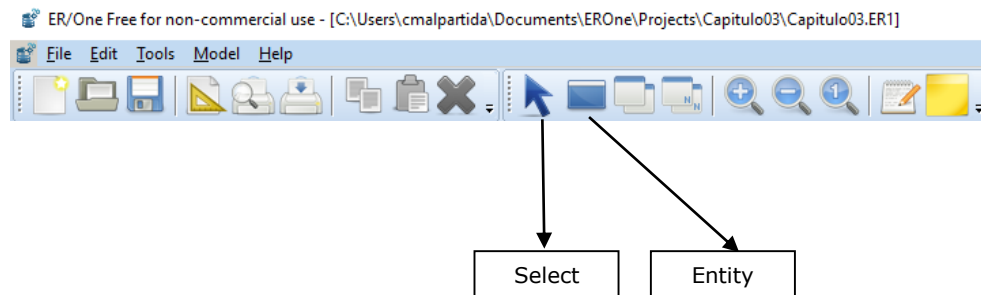
- 3º. Luego, aparecerá una nueva ventana, en ella, se debe ingresar el nombre del proyecto: luego, escoger la ruta en donde se va almacenar el proyecto y finalmente seleccionar el motor de base de datos, en este caso, **SQL Server**.



## PASO B: Definición de una nueva entidad

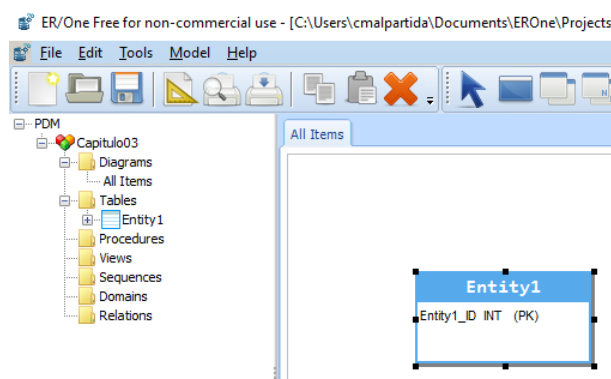
- **Creación de la entidad**

1. En **ER/One Toolbox**, hacer clic sobre el botón **Add Entity** y luego, clic sobre la posición del diagrama en la que se colocará la nueva entidad.

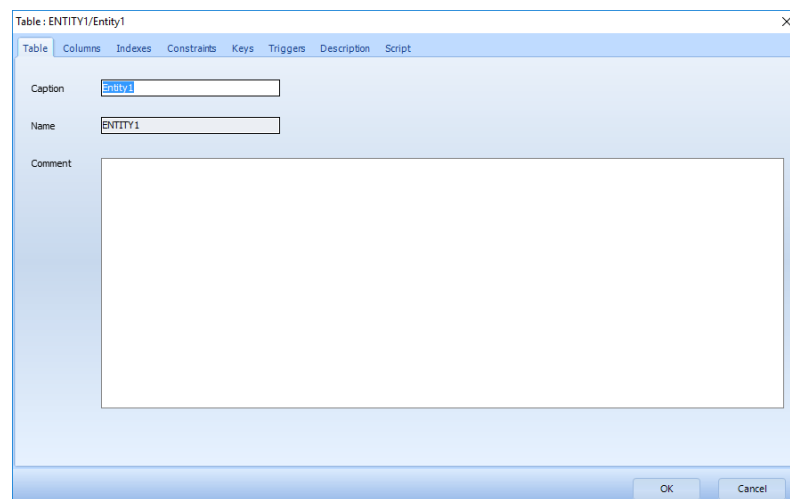


- **Definición del nombre de la entidad**

1. Al insertar la nueva entidad, aparece automáticamente el nombre de la entidad como **Entity1** y una llave primaria **Entity1\_ID**, para modificarlo, darle doble clic en la entidad.

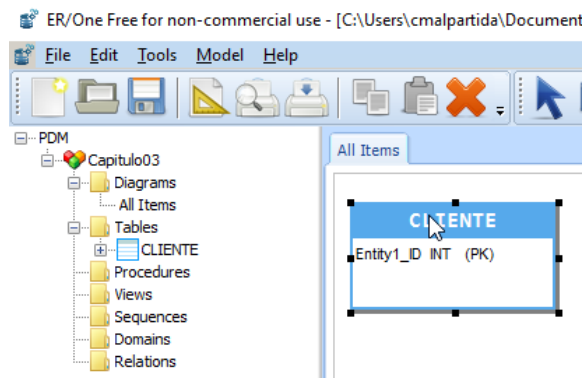


2. Al dar doble clic, aparece un cuadro de diálogo en donde se permite modificar el nombre de la Entidad, en este caso en la etiqueta **Caption**: digitar el nombre de la nueva entidad. Por ejemplo, CLIENTE.



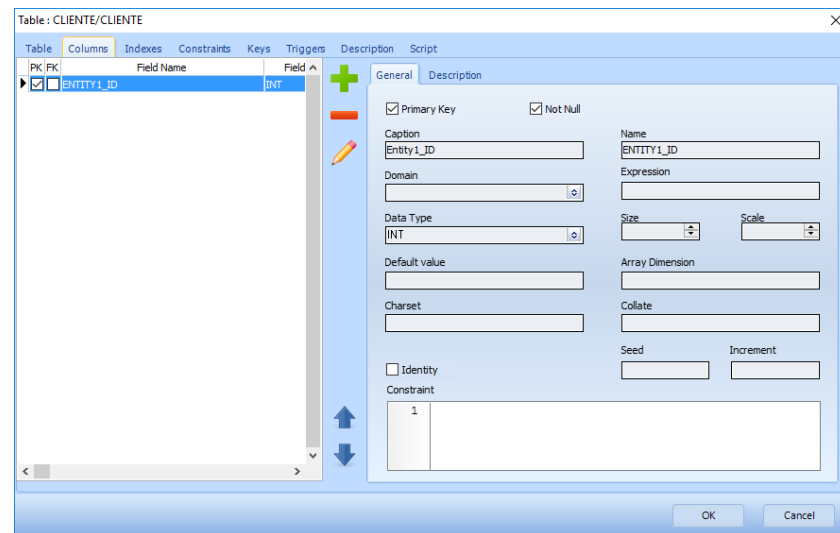


3. Hacer clic en el botón **OK**.





- **Definición de los atributos de la entidad**



1. Hacer doble clic sobre la entidad y en el cuadro de diálogo seleccionar la ficha **Columns**.

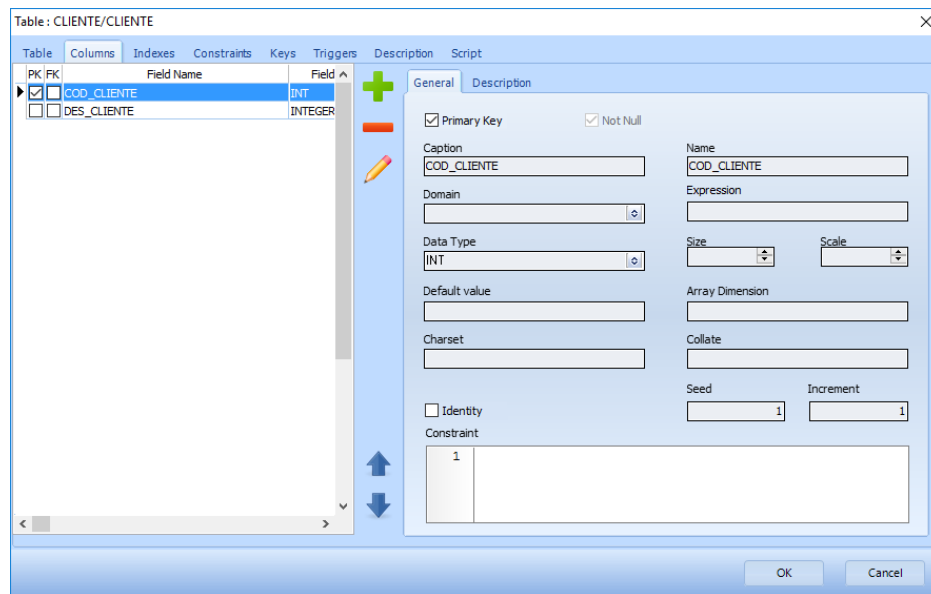


- **Modificar el nombre del atributo de la llave primaria**

1. Hacer clic en el ícono del lápiz  para poder editar el atributo.
2. En la ficha **General**, en el control **Caption**, modificar el nombre del atributo que será llave primaria. Por ejemplo, `cod_cliente`.
3. Dar clic en el ícono check , para confirmar el cambio.

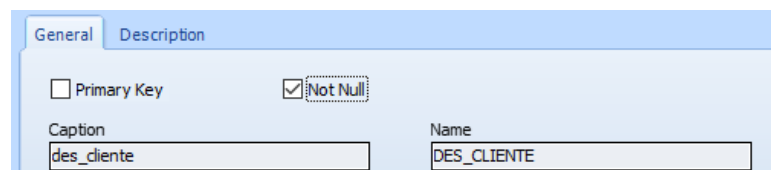
- **Para definir un nuevo atributo.**

1. Hacer clic en el botón  del diálogo **Table Editor**.
2. En la ficha **General**, en el control **Caption**, digitar el nombre del nuevo atributo. Por ejemplo, `des_cliente`.
3. Dar clic en el ícono check .



• **Para definir el tipo de dato de un atributo:**

1. Si el valor del atributo (dato) es el **Primary Key**, marcar en el recuadro respectivo, dicha opción.
2. Si el valor del atributo (dato) no va a recibir valores nulos, marcar en el recuadro respectivo de dicha opción.




## Definición de las relaciones

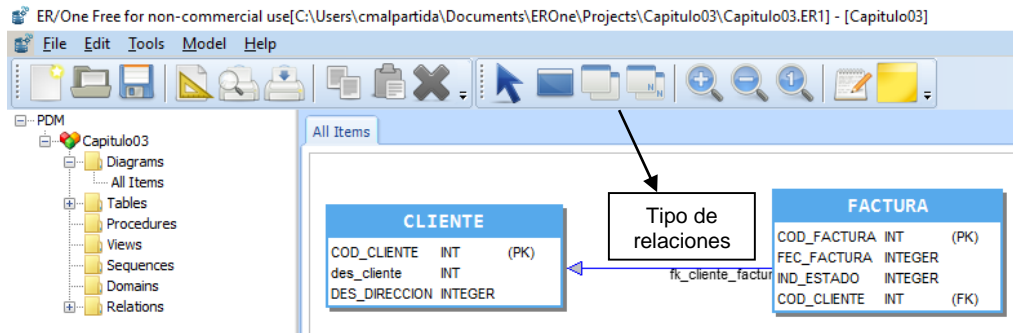
ER/One Data Modeler Tools permite especificar dos tipos de relaciones:

- Relación de identificación y no-identificación (relationship)
- Relación de muchos a muchos (Many-to-many relationship)

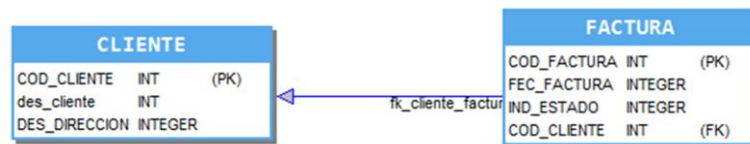
**Para definir la relación:** se definen dos entidades, **cliente** y **factura**, con algunos atributos.

Para CLIENTE: cod\_cliente, des\_cliente, des\_direccion y para FACTURA: cod\_factura, fec\_factura, ind\_estado.

1. En **ER/One Toolbox**, hacer clic en el botón  **Add Relationship**,
2. Seguido, hacer clic sobre la entidad padre (CLIENTE) y arrastrar el mouse presionando el botón izquierdo hacia la entidad hija (FACTURA).

**Muy importante:**

Observe que, en una relación no identificada, la llave foránea (FK) es un atributo no clave.



**Definición de una relación identificada**

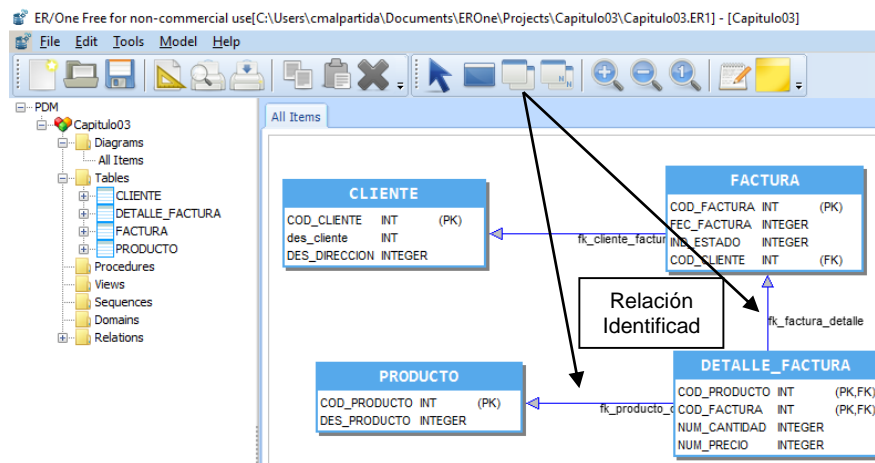
Se presenta cuando la entidad hija depende de la entidad padre para su identificación, es decir, una instancia de la entidad hija no puede existir sin su correspondiente instancia de la entidad padre. De este modo, una instancia de la entidad padre puede estar relacionada con muchas instancias de la entidad hija.

Por ejemplo, la relación entre las entidades FACTURA y DETALLE\_FACTURA. Una factura puede tener muchas líneas de detalle y para identificar a cada una de las líneas de detalle es necesario especificar a cuál de las facturas pertenece cada una de estas; por lo tanto, no puede existir una línea de detalle sin su correspondiente factura.

**Para definir la relación:**

1. Definir dos entidades: PRODUCTO y DETALLE\_FACTURA, con algunos atributos. PRODUCTO: **cod\_producto (Primary Key)**, **des\_producto** y DETALLE\_FACTURA con num\_cantidad y num\_precio, sin **Primary Key**.
2. En **ER/One Toolbox** dar clic en el botón **Add Relationship**.
3. Hacer clic sobre la entidad padre (FACTURA) y arrastrar el mouse presionando el botón izquierdo hacia la entidad hija (DETALLE\_FACTURA).
4. Hacer clic sobre la otra entidad padre (PRODUCTO) y arrastrar el mouse presionando el botón izquierdo hacia la entidad hija (DETALLE\_FACTURA).
5. Dar doble clic sobre la entidad DETALLE\_FACTURA y luego clic sobre la ficha **Columns**.
6. Nos posicionamos en el atributo COD\_FACTURA que es la llave foránea y hacemos clic en el ícono del lápiz para poder editar el atributo.
7. En la ficha **General**, seleccionar la opción **Primary Key**.

8. Dar clic en el ícono check , para confirmar el cambio.
9. Hacer los pasos 6, 7 y 8 para el atributo COD\_PRODUCTO.
10. Finalmente, posicionamos las columnas en las primeras posiciones usando la flecha superior .
11. Finalmente, dar clic en el botón **OK**.



**Muy importante:**

En una relación identificada, la llave foránea forma parte de la llave primaria de la entidad hija.

