



# **INSTITUTO POLITÉCNICO NACIONAL**

## **ESCUELA SUPERIOR DE CÓMPUTO**

### **BASES DE DATOS**

**GRUPO: 3CV5**

Profesor: Gabriel Hurtado Avilés

## **“Práctica 2 Modelo Entidad-Relación-Extendido”**

**Alumno:**

- González González Erick Emilio
- De La Rosa Hernández Tania

Fecha de Entrega: 03/10/25

## Ejercicio 1: Análisis y Expansión del Caso de Estudio

### 1.1 Revisión del problema original

El negocio “Sillas y mesas Hernández” gestiona alquileres para eventos pequeños y medianos. El problema original resuelve aspectos muy básicos, sin embargo, ya en la práctica surgen necesidades que aún no se cubren con un modelo ER normal. Los problemas y/o carencias que detectamos son las siguientes:

- Necesitamos llevar un control de disponibilidad en fechas concretas para saber si hay suficientes unidades libres para una fecha dada.
- Diferentes tipos de artículos con propiedades distintas. Como ejemplo una silla que puede ser plegable o de banquete.
- Registro histórico de mantenimiento y daños de artículos (para sacar unidades de servicio).
- Gestión de clientes de tipo empresa vs. particulares (datos y facturación distintos).
- Pagos con posibles anticipos o pagos parciales, y distintas formas de pago.
- Paquetes de evento (grupos de artículos que se ofrecen juntos) y precios especiales por paquete.
- Necesidad de acotar los pedidos, por ejemplo: un pedido no puede existir sin al menos un artículo, un pago debe estar ligado a un pedido.

### 1.2 Identificación de Nuevos Requisitos

Aquí se especifican las necesidades de expandir a un modelo entidad relación extendido.

#### **Pedidos (mínimo un producto)**

- *Necesidad EER:* cardinalidad mínima (Pedido — DetallePedido) con (min = 1).
- *Motivo:* evitar pedidos vacíos.

#### **Clientes de dos clases con datos distintos: particulares y empresas**

- *Necesidad EER: generalización / especialización* (supertipo Cliente → subtipos ClienteParticular, ClienteEmpresa)
- *Atributos por subtipo:*
  - ClienteParticular: fechaNacimiento, CURP (o identificación).
  - ClienteEmpresa: razonSocial, RFC, contactoEmpresa, giro.
- *Motivo:* reglas de facturación y campos obligatorios distintos.

### Artículos con subtipos que tienen atributos propios

- *Necesidad EER:* generalización Articulo → Silla, Mesa, Accesorio (o Paquete)
- *Atributos por subtipo:*
  - Silla: tipoSilla (apilable/banquete/plegable), material.
  - Mesa: forma, capacidadPersonas, tamaño.
  - Accesorio: descripcion, fragilidad.
- *Motivo:* consultas y mantenimientos según tipo; precios y disponibilidad por características.

### Registro de mantenimiento y daños del artículo

- *Necesidad EER:* entidad débil RegistroMantenimiento o HistorialArticulo ligada a Articulo (dependencia de existencia).
- *Atributos:* fecha, tipoAccion (reparación/limpieza/daño), costo, notas.
- *Motivo:* sacar unidades del inventario temporalmente; historial de costes.

### DetallePedido como entidad débil de identificación

- *Necesidad EER:* DetallePedido identificado por (idPedido, idArticulo) y con atributo cantidad.
- *Motivo:* implementar la relación N:M Pedido–Artículo y atributos propios (cantidad, precioUnitario en caso de registros históricos).

### Cliente con varios teléfonos / correos

- *Necesidad EER:* atributo multivaluado (Teléfonos) → en SQL, tabla Telefono.
- *Motivo:* contacto múltiple.

### Paquetes de artículos (ofertas)

- *Necesidad EER:* entidad Paquete que se relaciona con Articulo (N:M) y con Pedido (un pedido puede solicitar un paquete).
- *Motivo:* precios especiales y simplificación de pedidos.

### Reglas de integridad y restricciones

- *Necesidad EER:* cardinalidades mínimas/máximas explícitas (por ejemplo, un pago debe estar ligado a exactamente un pedido; un pedido puede tener 0..n pagos).
- *Motivo:* mantener consistencia y forzar políticas (no permitir pagos huérfanos).

## Ejercicio 2: Aplicación de Conceptos del Modelo E-R Extendido

### 2.1 Cardinalidades Mínima y Máxima

#### 1. Cliente – Realiza – Pedido

- **Cardinalidad:** Cliente (1, n) : Pedido (1, 1)
- **Justificación:** Un cliente puede hacer uno o muchos pedidos a lo largo del tiempo. Todo pedido debe estar asociado a un cliente que exista. No se permite un pedido sin cliente, ni un cliente que nunca pueda hacer pedidos.

#### 2. Pedido – Contiene – DetallePedido

- **Cardinalidad:** Pedido (1, 1) : DetallePedido (1, n)
- **Justificación:** Un pedido debe tener al menos un detalle (no se permiten pedidos vacíos). Cada detalle pertenece a un solo pedido, pero un pedido puede tener muchos detalles.

#### 3. Articulo – SeAsocia – DetallePedido

- **Cardinalidad:** Articulo (0, n) : DetallePedido (1, 1)
- **Justificación:** Un artículo puede no estar en ningún pedido (ej. recién ingresado al inventario) o en muchos pedidos diferentes. Cada detalle de pedido corresponde siempre a un solo artículo.

### 2.2 Entidades débiles

#### ‘DetallePedido’

- **Tipo de dependencia:** Dependencia de **identificación**.
- **Justificación:** DetallePedido no puede existir sin un Pedido y un Articulo. Su identificador se compone de ambas claves (idPedido, idArticulo). Además, almacena atributos propios como **cantidad** (y podría extenderse a precioUnitario o descuentos).

#### ‘PedidoPaquete’

- **Tipo de dependencia:** Dependencia de **identificación**.
- **Justificación:** La entidad **PedidoPaquete** surge para resolver la relación de **muchos a muchos (N:M)** entre Pedido y Paquete. Un pedido puede incluir varios paquetes y un mismo paquete puede estar presente en distintos pedidos.

### 2.3 Herencia: Subtipos y Supertipos

#### ‘Jerarquía 1: Cliente’

- **Supertipo:** Cliente

- Atributos generales: idCliente, nombre, dirección, correo.
- **Subtipos:**
  - ClienteParticular → atributos propios: fechaNacimiento, CURP.
  - ClienteEmpresa → atributos propios: razonSocial, RFC, contactoEmpresa.
- **Tipo de especialización: Disjunta y Total.**
  - Disjunta: un cliente es empresa o particular, pero no ambos.
  - Total: todo cliente debe pertenecer a uno de los subtipos.

### ‘Jerarquía 2: Artículo’

- **Supertipo:** Articulo
  - Atributos generales: idArticulo, nombre, cantidadTotal, estado, costoRenta.
- **Subtipos:**
  - Silla → atributos propios: tipoSilla, material.
  - Mesa → atributos propios: forma, capacidadPersonas, tamaño.
  - Accesorio → atributos propios: descripcion, fragilidad.
- **Tipo de especialización: Disjunta y Total.**
  - Disjunta: un artículo solo puede ser de un tipo.
  - Total: todo artículo debe clasificarse en alguno de los subtipos.

## 2.4 Otros conceptos avanzados

**-Atributos multivaluados:** Cliente puede tener varios teléfonos → atributo multivaluado telefono.

**-Atributos compuestos:** Dirección puede dividirse en calle, número, colonia, ciudad, CP.

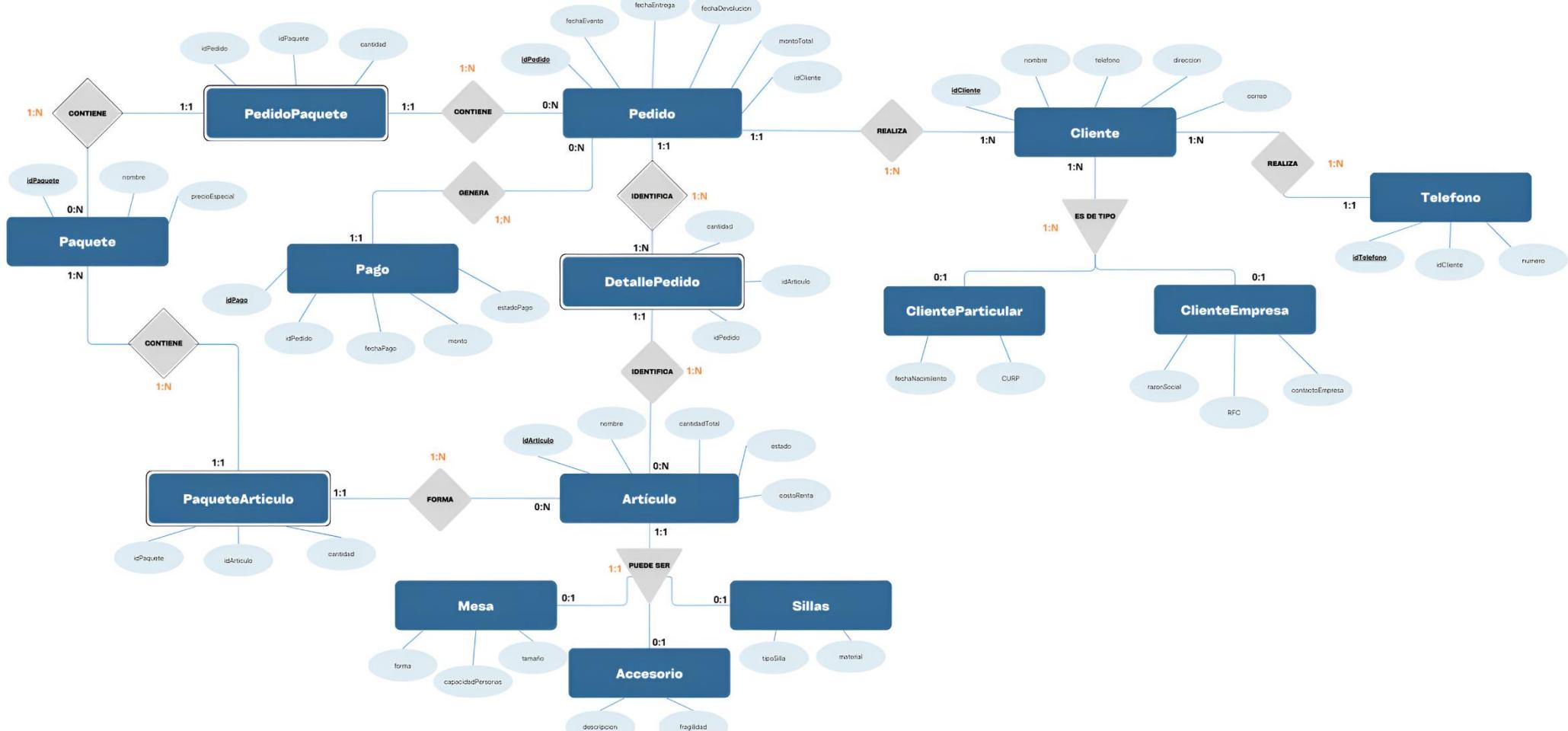
**-Agregación / Categorización:** No aplica en este caso (escala pequeña).

### **-Relaciones adicionales:**

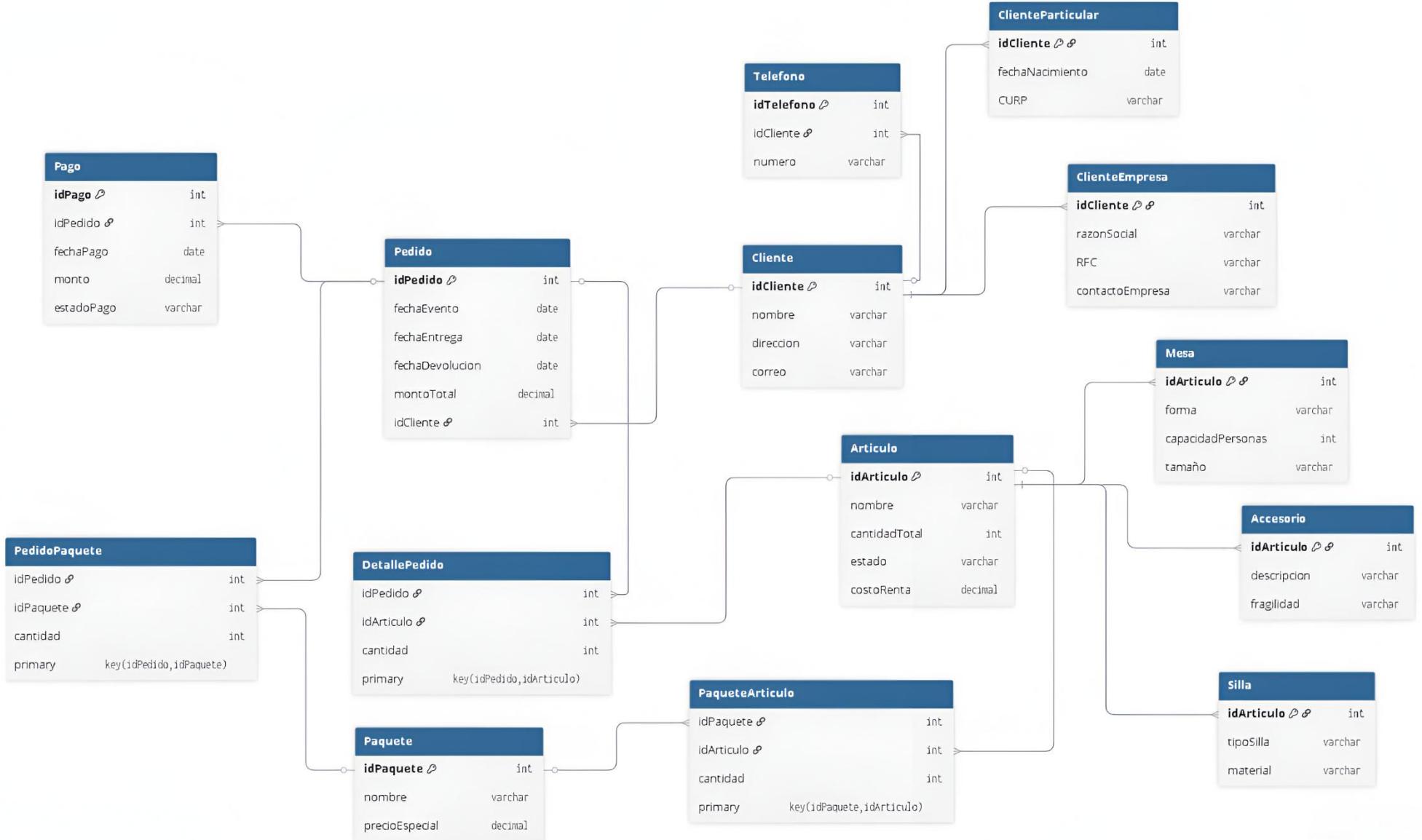
- Paquete (entidad) que se relaciona con Articulo (N:M) para permitir ofertas especiales.
- Pedido puede incluir tanto artículos individuales como paquetes.

## Ejercicio 3: Modelado EER con Múltiples Notaciones

### 3.1 Diagrama en Notación Peter Chen



### 3.2 Diagrama en Notación Crow's Feet



### 3.3 Herramientas Utilizadas

#### 1. Canva

Se utilizó para construir el diagrama con la notación de **Peter Chen**, ya que permite mayor libertad gráfica en el diseño y acomodo de los elementos (rectángulos para entidades, óvalos para atributos y rombos para relaciones).

#### 2. dbdiagram.io

Se usó para representar el modelo con notación de **Crow's Foot**. Esta herramienta está orientada directamente al modelado de bases de datos, lo que nos permitió generar diagramas a partir de código SQL o de su propio lenguaje de definición.

## Ejercicio 4: Análisis y Justificación del Modelo

### 4.1 Explicación Detallada del Diseño

#### 1. Justificación de entidades débiles

- **DetallePedido** es una entidad débil por **dependencia de identificación**, ya que no puede existir sin un Pedido ni un Artículo, y su clave primaria se forma a partir de ambas. Además, almacena atributos propios como **cantidad**.
- **PaqueteArticulo** y **PedidoPaquete** funcionan como tablas de relación débiles, porque solo existen como extensión de Paquete–Artículo y Pedido–Paquete. Estas permiten modelar relaciones N:M y capturar atributos adicionales (cantidad).

#### 2. Análisis de la herencia

- **Cliente** se generaliza en **ClienteParticular** y **ClienteEmpresa**, porque cada subtipo requiere información distinta: CURP y fecha de nacimiento en el caso de particulares, y razón social y RFC en el caso de empresas. Esta especialización es **disjunta y total**, ya que un cliente no puede ser ambos a la vez y todo cliente debe ser clasificado.
- **Artículo** se generaliza en **Silla, Mesa y Accesorio**, con atributos específicos (ej. capacidadPersonas solo aplica a Mesa). Esta especialización también es **disjunta y total**, ya que cada artículo pertenece a un solo subtipo y no existen artículos fuera de estas categorías.
- **Beneficio:** esta jerarquía permite consultas más claras (ej. “todas las mesas con capacidad mayor a 10 personas”) y asegura que cada entidad tenga los atributos correctos.

#### 3. Cardinalidades específicas

- En **Pedido–DetallePedido**, la cardinalidad mínima (1) asegura que no existan pedidos vacíos.
- En **Cliente–Pedido**, se estableció (1,n) para reflejar que un cliente puede realizar varios pedidos, pero todo pedido debe estar ligado a un cliente.
- En **Pedido–Pago**, se estableció (0,n) para pagos, reflejando la posibilidad de anticipos y múltiples pagos asociados a un pedido. Estas reglas de cardinalidad garantizan consistencia en el modelo y evitan datos inválidos.

#### 4. Comparación entre notaciones

- **Peter Chen:** permite un modelado conceptual más expresivo y visual, mostrando entidades, atributos y relaciones con óvalos, rectángulos y rombos. Es más claro en general.
- **Crow's Foot:** es más técnico, orientado a bases de datos relacionales, mostrando llaves primarias y foráneas directamente. Es más útil para el diseño lógico y posterior implementación.

## 4.2 Validación del Modelo

### **Consultas complejas que requieren el modelo extendido:**

1. *Listar todos los pedidos realizados por clientes de tipo empresa junto con su RFC y los artículos rentados en cada pedido.*
  - Requiere la jerarquía Cliente → ClienteEmpresa.
2. *Obtener todos los artículos de tipo mesa con capacidad mayor a 10 personas que fueron incluidos en pedidos durante septiembre.*
  - Requiere la jerarquía Articulo → Mesa y la relación Pedido–DetallePedido.
3. *Calcular los ingresos generados por cada paquete de artículos en el último mes.*
  - Requiere la entidad Paquete y la relación PedidoPaquete.

### **Integridad de datos**

- Cardinalidades mínimas/máximas aseguran reglas de negocio (no pedidos vacíos, no pagos sin pedido).
- Entidades débiles aseguran que DetallePedido y PaqueteArticulo no existan aisladas.
- Jerarquías ISA aseguran que cada cliente y artículo tenga los atributos correctos.

### **Escalabilidad**

- El modelo puede crecer fácilmente para manejar más tipos de artículos (añadiendo subtipos a Articulo).
- Puede incluir más métodos de pago o ampliar el control de inventario sin romper la estructura existente.
- La inclusión de Paquetes permite escalar hacia promociones o combos más complejos.

## Ejercicio 5: Reflexión sobre el Proceso de Diseño

### 5.1 Comparación con el Modelo Básico

El modelo básico de la **Práctica 1** resultaba suficiente para representar lo esencial: clientes, artículos, pedidos, pagos y el detalle de pedidos. Sin embargo, tenía varias **limitaciones**:

- No diferenciaba entre tipos de clientes (empresa o particular).
- No distinguía artículos por sus características específicas (sillas, mesas, accesorios).
- No manejaba atributos multivaluados (como teléfonos de clientes).
- No expresaba restricciones de negocio mediante cardinalidades mínimas/máximas (ej. pedidos vacíos).
- No contemplaba paquetes de artículos como una opción comercial.

Con el **modelo extendido (EER)**, estas limitaciones se resolvieron:

- Se aplicó **generalización/especialización** para diferenciar clientes y artículos, lo que permite registrar atributos más específicos y consultas más precisas.
- Se introdujeron **cardinalidades mínimas/máximas** que aseguran reglas de integridad, como que todo pedido tenga al menos un detalle.
- Se incluyeron **atributos multivaluados** para reflejar la realidad de que un cliente puede tener varios teléfonos.
- Se añadieron **entidades débiles** (DetallePedido, PaqueteArticulo, PedidoPaquete) que garantizan la correcta representación de relaciones N:M.
- Se incorporó la entidad **Paquete**, lo que amplía la funcionalidad del sistema para manejar promociones o combos de artículos.

### 5.2 Lecciones Aprendidas

#### **Dificultades encontradas:**

- La identificación de entidades débiles y la forma correcta de representarlas con sus relaciones de identificación fue un reto conceptual al inicio.
- La definición de cardinalidades mínimas/máximas hizo necesario analizar con más detenimiento las reglas del negocio.

#### **Diferencias en la expresividad de las notaciones:**

- La notación **Peter Chen** es ideal para comunicar la lógica de negocio porque permite visualizar atributos, relaciones y jerarquías de manera pedagógica.
- La notación **Crow's Foot** es más técnica, orientada al diseño lógico de la base de datos, mostrando llaves primarias y foráneas, lo que facilita su implementación.

#### **Importancia del modelo conceptual:**

- Haber trabajado primero con un modelo básico permitió comprender los elementos fundamentales.
- El paso al modelo extendido nos mostró cómo los conceptos avanzados enriquecen el diseño, y lo proveen de un detalle que es de suma importancia.
- Se reafirma que un buen modelo conceptual es clave para que la implementación en un SGBD sea confiable y más escalable en el tiempo (al realizar el ERE por ejemplo).

## Bibliografía

. Connolly, T. M., & . Begg, C. E. (2014). *Database systems: A practical approach to design, implementation and management.*

*A free database designer for developers and analysts.* (s/f). Dbdiagram.io. Recuperado el 27 de septiembre de 2025, de <https://dbdiagram.io/home>

(S/f). Canva.com. Recuperado el 27 de septiembre de 2025, de <https://www.canva.com/>

Criss, P. (s/f). *6 Como crear Un supertipo subtipo mere logico* Youtube. Recuperado el 28 de septiembre de 2025, de <https://www.youtube.com/watch?v=lMu9FI7YQA4>