



# **INSTITUTO POLITÉCNICO NACIONAL**

## **ESCUELA SUPERIOR DE CÓMPUTO**

### **BASES DE DATOS**

**GRUPO: 3CV5**

Profesor: Gabriel Hurtado Avilés

### **“Práctica 1 Modelo Entidad-Relación”**

**Alumno:**

- González González Erick Emiliiano

Fecha de Entrega: 19/09/25

## Ejercicio 1: Instalación de su Sistema de Gestión de Bases de Datos (SGDB)

```
C:\Users\ggani\OneDrive\Escritorio\ESCUELA\DB\Sillas y Mesas Hernández>docker compose up -d
time="2025-09-06T11:18:45-06:00" level=warning msg="C:\\\\Users\\\\ggani\\\\OneDrive\\\\Escritorio\\\\ESCUELA\\\\DB\\\\Sillas y Mesas Hernández\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 2/2
  ✓ Network sillasyMesashernández_default      Created
  ✓ Container postgres_container_sillasyMesas  Started
                                         0.1s
                                         0.6s
```

### Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

The screenshot shows the Docker interface with the following details:

- Container CPU usage:** 0.00% / 1200% (12 CPUs available)
- Container memory usage:** 33.09MB / 7.5GB
- Show charts:** Link to view performance charts.
- Search:** Input field for searching containers.
- Only show running containers:** Filter option.
- Table:**

	Name ↑	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	sillasyMesashernández	-	-	-	0.01%	18 minutes ago	[Logs] [Stop] [More] [Delete]
□	postgres_container_sillasyMesas	d8917434a2ce	postgres:17	5433:5432	0.01%	18 minutes ago	[Logs] [Stop] [More] [Delete]
- Terminal:**

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. <https://aka.ms/PSWindows>

```
PS C:\Users\ggani> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
d8917434a2ce        postgres:17       "docker-entrypoint.s..."   About a minute ago   Up 6 seconds          0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp   postgres_container_sillasyMesas
PS C:\Users\ggani>
```

## SGDB

Evidencia del SGDB en correcto funcionamiento y contenedor de PostgreSQL en ejecución dentro de Docker, mostrando que la base de datos se encuentra levantada y disponible para su uso.

## Ejercicio 2: El Caso de Estudio (Problema de la vida real)

### Problemática (ENTREVISTA)

El negocio “Sillas y mesas Hernández” de la familia de mi amigo Isaac enfrenta problemas para organizar su inventario, los pedidos de los clientes, la disponibilidad de los artículos y el control de pagos. Actualmente, la administración se hace de forma manual por sus padres en libretas y hojas de cálculo como Excel, lo que genera errores al confirmar disponibilidad, duplicación de reservas y dificultades para llevar un registro eficiente de ingresos.

Es por eso por lo que me dijo que una base de datos ayudaría a organizar la información de clientes, artículos, pedidos, eventos y pagos, evitando confusiones más que nada.

### Simulación de entrevista con el cliente

*Consultor (yo):* ¿Qué información necesita guardar sobre los artículos que renta?

*Cliente (dueño del negocio):* Necesito registrar el nombre del artículo, su tipo (silla, mesa o mantel) la cantidad total disponible, su estado (en buen estado, en reparación, etc.) y el costo de renta por unidad.

*Consultor:* ¿Y qué datos desea tener de los clientes?

*Cliente:* Ocupo tener el nombre completo, teléfono, dirección y correo electrónico.

*Consultor:* Cuando alguien hace un pedido, ¿qué información se debe guardar?

*Cliente:* Quiero que el pedido incluya la fecha del evento, la fecha de entrega, la fecha de devolución, los artículos rentados con sus cantidades, y el monto total a pagar.

*Consultor:* ¿Cómo desea manejar los pagos?

*Cliente:* Que quede registrado si el pago fue completo o parcial, la fecha en que se realizó, el monto y si se pude, saber si fue en efectivo o tarjeta.

*Consultor:* ¿Le interesa generar reportes?

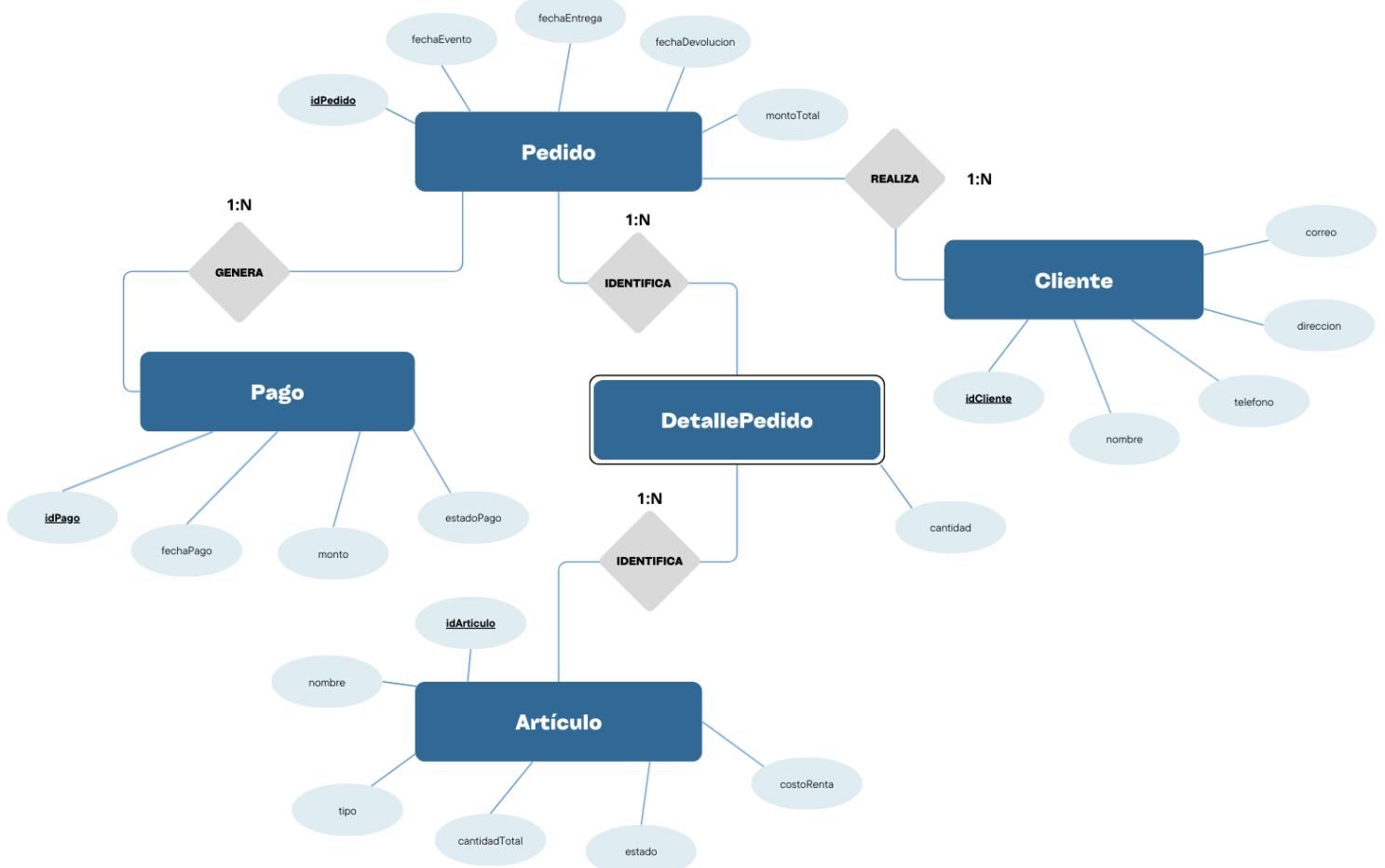
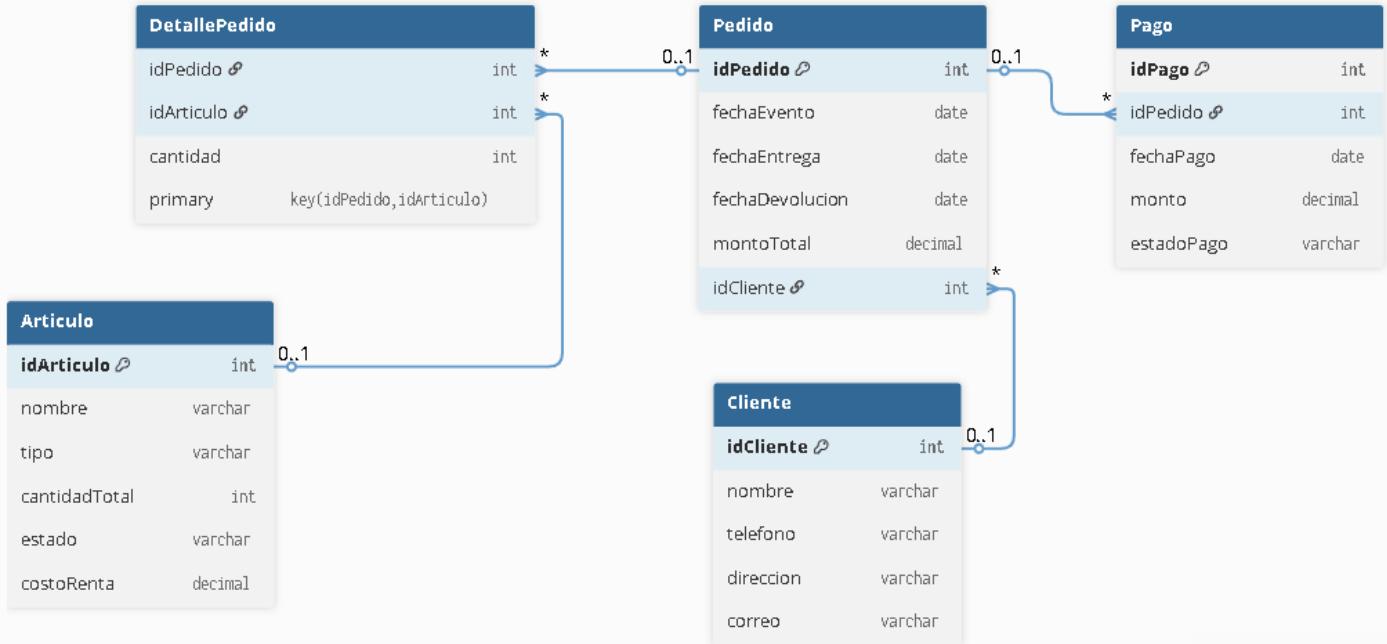
*Cliente:* Sí, me interesa saber qué artículos son los más rentados, cuáles están disponibles en cierta fecha, y también tener un historial de pedidos por cliente.

### Requerimientos del sistema

1. El sistema debe guardar información de los clientes (nombre, teléfono, dirección, correo).
2. El sistema debe registrar los artículos disponibles para renta (nombre, tipo, cantidad, estado, costo de renta).

3. El sistema debe registrar los pedidos de los clientes, incluyendo:
  - Cliente que lo realiza.
  - Fechas de evento, entrega y devolución.
  - Lista de artículos rentados y cantidad.
  - Monto total del pedido.
4. El sistema debe manejar pagos de cada pedido (monto, fecha, estado del pago: parcial o completo, método de pago).
5. El sistema debe permitir reportes como: artículos más rentados, disponibilidad de artículos, historial de pedidos por cliente.

## Ejercicio 3: Modelado del Problema



### Consideraciones:

- ERD significado de las puntas de flecha

 Many también denotado con \*

 Zero or one también denotado con 0..1

- Cardinalidades

El modelo Crow's Foot Notation no incluye cardinalidades extendidas. Por esa razón se incluyen aparte.

**1:N**



**N:N**



**1:N**



### **Explicación del modelo:**

El diseño de las entidades y sus respectivas relaciones se decidió así tomando en consideración los requerimientos que el cliente nos expresó en la entrevista. La entidad Cliente almacena la información de las personas que solicitan el servicio; cada cliente puede realizar varios pedidos, pero cada pedido está asociado únicamente a un cliente, lo que genera una relación 1:N.

La entidad Artículo representa los objetos disponibles para renta. Como un pedido puede incluir distintos artículos y, a la vez, un artículo puede aparecer en varios pedidos, se requiere la tabla intermedia DetallePedido, que registra también la cantidad solicitada. Esto forma una relación N:M entre Pedido y Artículo.

Y por último, la entidad Pago permite llevar control de los cobros asociados a cada pedido. Un pedido puede liquidarse en uno o varios pagos (por ejemplo, anticipo y pago final), mientras que cada pago está ligado a un solo pedido, lo que da lugar a una relación 1:N.

Todo esto en su conjunto abarca las peticiones originales y soluciona los problemas que el cliente mencionaba en la entrevista.

## Bibliografía

*A free database designer for developers and analysts.* (s/f). Dbdiagram.io. Recuperado el 6 de septiembre de 2025, de <https://dbdiagram.io/home>

*Entity-relationship (ER) diagram symbols and notations.* (s/f). Wondershare.com. Recuperado el 6 de septiembre de 2025, de <https://edraw.wondershare.com/er-diagram-symbols.html>