



# **Diseño del modelo de datos: Diagrama E/R o Diagrama Relacional**

## **Raíces**

**“Conectando tierra y vida”**

**Desarrollo de Aplicaciones Web**

**Proyecto de Desarrollo de Aplicaciones Web**

**Javier Hipólito Rodríguez**

**Versión 1.0**

<b>Nombre del fichero:</b>	DAW_PRW_[APCODE]_UT01.3. Diagrama_ER.pdf
<b>Fecha de esta versión:</b>	05-01-26

## **Historial de revisiones**

<b>Fecha</b>	<b>Descripción</b>	<b>Autor</b>
21-12-25	Creación del documento	Javier Hipólito Rodríguez
05-01-26	Justificación documento, revisión redacción, añadir definiciones atributos tablas	Javier Hipólito Rodríguez

# Índice

1. Introducción .....	3
2. Descripción general del modelo de datos .....	3
3. Entidades y atributos.....	4
3.1. Entidad: USUARIOS.....	4
3.2. Entidad: PRODUCTOS .....	5
3.3. Entidad: PEDIDOS .....	6
3.4. Entidad: PEDIDO_DETALLES.....	6
4. Relaciones y cardinalidades.....	7
4.1. Relación: USUARIOS – PRODUCTOS.....	7
4.2. Relación: USUARIOS – PEDIDOS.....	7
4.3. Relación: PEDIDOS – PEDIDO_DETALLES.....	7
4.4. Relación: PRODUCTOS – PEDIDO_DETALLES.....	7
5. Diagrama Entidad - Relación .....	8
6. Modelo físico .....	9
7. Justificación de las decisiones de diseño .....	9

## 1. Introducción

El proyecto Raíces consiste en una plataforma web orientada a la venta de productos agroalimentarios de proximidad, fomentando el consumo local y el comercio directo entre productores y clientes finales.

La aplicación permite la gestión de usuarios con diferentes roles (clientes y administradores/productores), la publicación de productos con información detallada sobre su procedencia y unidad de venta, así como la realización de pedidos que pueden incluir múltiples productos.

Con el objetivo de garantizar un diseño de base de datos coherente, escalable y alineado con las necesidades del sistema, se ha desarrollado un Diagrama Entidad–Relación (ER) que representa las entidades principales del sistema, sus atributos y las relaciones existentes entre ellas.

## 2. Descripción general del modelo de datos

El modelo de datos se ha diseñado siguiendo los principios de normalización y buenas prácticas en bases de datos relacionales. Se han identificado las siguientes **entidades principales**:

- **USUARIOS:** representan a las personas que interactúan con la plataforma, tanto clientes como administradores/productores.
- **PRODUCTOS:** representan los productos agroalimentarios ofrecidos en la plataforma.
- **PEDIDOS:** representan las transacciones realizadas por los clientes.
- **DETALLE DE PEDIDOS:** representan los productos incluidos en cada pedido.

Estas entidades permiten representar de forma clara los actores del sistema, los elementos comercializados y las transacciones realizadas.

Las entidades **USUARIOS**, **PRODUCTOS** y **PEDIDOS** se consideran **entidades fuertes**, ya que poseen atributos que permiten identificarlas de forma única. La entidad **PEDIDO\_DETALLES** se considera una **entidad débil**, ya que su existencia depende de la existencia previa de un pedido y de un producto, tal y como se describe en el modelo E/R para la resolución de relaciones de cardinalidad muchos a muchos.

### 3. Entidades y atributos

#### 3.1. Entidad: USUARIOS

Representa a todas las personas que interactúan con la plataforma, tanto clientes como productores (administradores).

##### Atributos:

- **id (PK):** Identificador único y numérico de cada usuario. Se define como Clave Primaria (*Primary Key*) autoincremental para garantizar la integridad de los registros. Atributo no nulo.
- **nombre:** Cadena de caracteres que almacena el nombre del usuario. Atributo no nulo.
- **apellido:** Cadena de caracteres que almacena los apellidos del usuario. Atributo no nulo.
- **correo:** Cadena de caracteres que almacena el correo electrónico. Este atributo se ha definido como único (*unique*) para evitar duplicidad de cuentas. Atributo no nulo.
- **contraseña\_hash:** Almacena la clave de acceso del usuario tras ser procesada por un algoritmo de cifrado, garantizando la seguridad de la información sensible. Atributo no nulo.
- **telefono:** Cadena de caracteres que almacena el número de teléfono. Se ha establecido como cadena de caracteres para que el valor introducido por el usuario permita diferentes formatos de teléfono (Ejemplo: +34689374826, 620-45-67-83, etc). Atributo opcional, puede ser nulo.
- **rol (cliente / admin):** Campo que determina, mediante un enumerador los diferentes roles dentro del sistema, diferenciando entre cliente o administrador. Atributo no nulo. Valor por defecto: ‘cliente’.
- **direccion:** Cadena de caracteres donde se almacena la calle, número y puerta del domicilio del usuario para envíos y facturación. Atributo opcional, puede ser nulo, ya que el alcance del proyecto no incluye la logística del envío del pedido.
- **codigo\_postal:** Cadena de caracteres donde se almacena el código postal. Se ha establecido como cadena de caracteres para los códigos postales que empiezan por el dígito 0. Atributo opcional, puede ser nulo, ya que el alcance del proyecto no incluye la logística del envío del pedido.
- **poblacion:** Cadena de caracteres donde se almacena la localidad de residencia. Atributo opcional, puede ser nulo, ya que el alcance del proyecto no incluye la logística del envío del pedido.

- **municipio:** Cadena de caracteres donde se almacena el municipio de residencia. Atributo opcional, puede ser nulo, ya que el alcance del proyecto no incluye la logística del envío del pedido.
- **fecha\_creacion:** sello de tiempo (timestamp) que registra automáticamente el momento en el que el usuario se dio de alta en el sistema.

La entidad USUARIOS no contiene claves foráneas, ya que es una entidad fuerte e independiente dentro del sistema. Otras entidades hacen referencia a ella mediante claves foráneas.

### **3.2. Entidad: PRODUCTOS**

Representa los productos disponibles para la venta. Cada producto pertenece a un único productor (usuario administrador).

#### **Atributos:**

- **id (PK):** Identificador único y numérico de cada producto. Se establece como clave primaria autoincremental para asegurar la distinción única de cada artículo. Atributo no nulo.
- **nombre:** Cadena de caracteres que almacena el nombre del producto. Atributo no nulo.
- **descripción:** Cadena de caracteres donde se especifica información descriptiva sobre el producto. Atributo opcional, puede ser nulo.
- **precio:** Valor monetario del producto definido como DECIMAL(10,2). Esta configuración permite almacenar importes de hasta 8 dígitos enteros y 2 decimales (ej. 99.999.999,99). Se ha decidido este tipo de datos para garantizar precisión y reducir errores en la suma de totales del pedido y del producto. Atributo no nulo.
- **stock:** Cantidad numérica disponible en base a la unidad de medida especificada. Atributo no nulo. Cuando no hay stock se especifica con el número 0.
- **unidad\_medida (kg, g, unidad):** Atributo del tipo enumerador que define la métrica de venta del producto para así poder calcular el precio. Atributo no nulo. El valor por defecto es 'kg'.
- **imagen\_url:** Enlace a la ubicación de la imagen del producto. Atributo opcional, puede ser nulo.
- **procedencia:** cadena de caracteres que indica el origen del producto. Atributo opcional, puede ser nulo.
- **usuario\_id (FK):** Clave foránea que hace referencia al atributo **id** de la entidad USUARIOS (identifica al usuario administrador o socio responsable del producto). Atributo no nulo.

- **fecha\_creacion:** sello de tiempo (timestamp) que registra automáticamente el momento en el que el producto se dio de alta en el sistema. Atributo no nulo.

### **3.3. Entidad: PEDIDOS**

Representa los pedidos realizados por los clientes dentro del sistema.

**Atributos:**

- **id (PK):** Identificador único y numérico del pedido. Se utiliza como clave primaria para referenciar cada transacción de forma única. Atributo no nulo.
- **usuario\_id (FK):** Clave foránea que hace referencia al atributo **id** de la entidad USUARIOS (identifica al usuario cliente que realiza el pedido). Este atributo asegura que todo pedido pertenezca a un usuario. Atributo no nulo.
- **total:** Valor numérico de tipo DECIMAL (10,2) que representa el importe final del pedido. Atributo no nulo.
- **direccion\_entrega:** Cadena de caracteres que almacena la ubicación donde debe enviarse el pedido. Atributo opcional, puede ser nulo.
- **fecha\_creacion:** sello de tiempo (timestamp) que registra automáticamente el momento en el que el pedido se dio de alta en el sistema. Atributo no nulo.
- **estado:** Atributo del tipo enumerador donde se especifica el estado del pedido siendo las opciones disponibles: ‘pendiente’, ‘procesando’, ‘enviado’, ‘completado’, ‘cancelado’. El valor predeterminado de este atributo es ‘Pendiente’. Atributo no nulo.

### **3.4. Entidad: PEDIDO\_DETALLES**

Representa el detalle de los productos incluidos en cada pedido. Actúa como entidad intermedia entre pedidos y productos.

**Atributos:**

- **id (PK):** Identificador único y numérico del detalle del pedido. Atributo no nulo.
- **pedido\_id (FK):** Clave foránea que hace referencia al atributo **id** de la entidad PEDIDOS (indica a qué pedido pertenece el detalle). Atributo no nulo.
- **producto\_id (FK):** Clave foránea que hace referencia al atributo **id** de la entidad PRODUCTOS (indica qué producto se ha incluido en el pedido). Atributo no nulo.
- **cantidad:** atributo de tipo numérico que especifica el número unidades (unidad de medida) del producto. Atributo no nulo.

- **precio\_unitario:** atributo del tipo DECIMAL(10,2) que registra el precio del producto en el momento de hacer el pedido. Atributo no nulo.

## 4. Relaciones y cardinalidades

### 4.1. Relación: USUARIOS – PRODUCTOS

- Un usuario administrador puede gestionar **uno o varios productos**.
- Cada producto pertenece a **un único usuario**.

**Cardinalidad:**

USUARIOS (1) — (N) PRODUCTOS

### 4.2. Relación: USUARIOS – PEDIDOS

- Un cliente puede realizar **varios pedidos**.
- Cada pedido pertenece a **un solo cliente**.

**Cardinalidad:**

USUARIOS (1) — (N) PEDIDOS

### 4.3. Relación: PEDIDOS – PEDIDO\_DETALLES

- Un pedido puede contener **varios productos**.
- Cada detalle pertenece a **un único pedido**.

**Cardinalidad:**

PEDIDOS (1) — (N) PEDIDO\_DETALLES

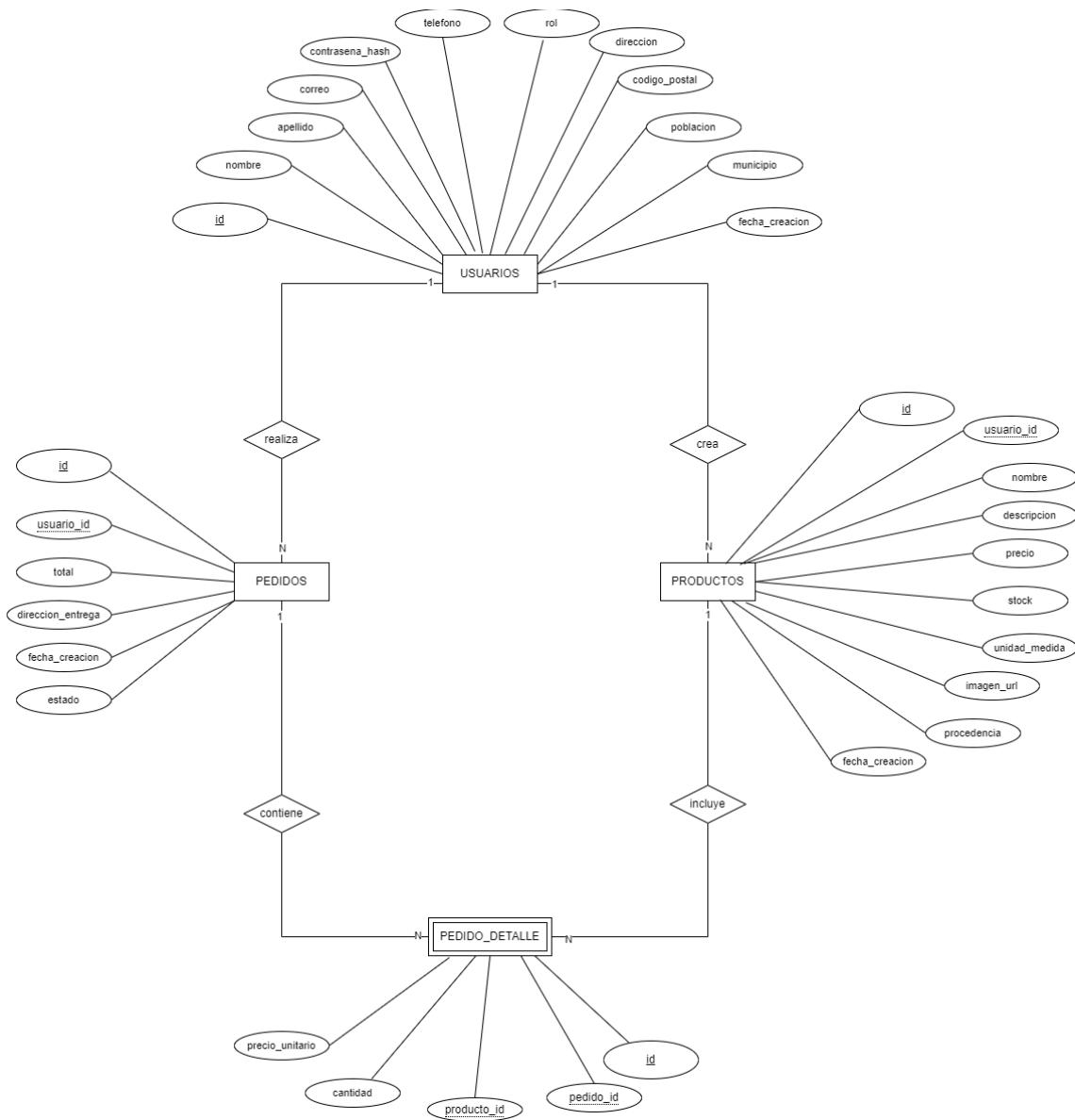
### 4.4. Relación: PRODUCTOS – PEDIDO\_DETALLES

- Un producto puede aparecer en **múltiples pedidos**.
- Cada detalle hace referencia a **un único producto**.

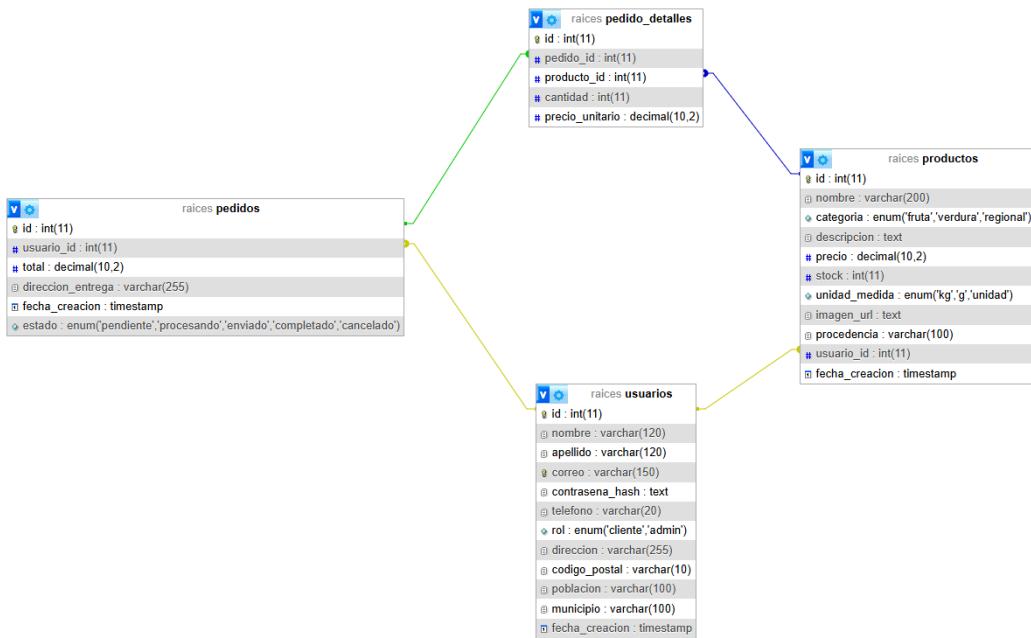
**Cardinalidad:**

PRODUCTOS (1) — (N) PEDIDO\_DETALLES

## 5. Diagrama Entidad - Relación



## 6. Modelo físico



El modelo físico se ha exportado desde el panel de phpMyAdmin, una herramienta de administración gráfica basada en web para bases de datos, a través del apartado de “Diseñador”.

## 7. Justificación de las decisiones de diseño

El diseño del modelo de datos se ha realizado priorizando la claridad, la escalabilidad y la coherencia con un entorno real de comercio electrónico. La separación entre pedidos y detalle de pedidos permite gestionar pedidos con múltiples productos sin redundancia de información.

La incorporación de atributos como **procedencia** y **unidad\_medida** en la entidad **PRODUCTOS** responde a la necesidad de representar productos reales, que no siempre se venden por unidad, sino también por peso (kilogramos o gramos). Del mismo modo, la distinción de roles dentro de la entidad **USUARIOS**, mediante el atributo *rol*, permite controlar los permisos del sistema y diferenciar entre clientes y productores sin necesidad de duplicar entidades.

Durante el proceso de normalización se han aplicado las formas normales hasta alcanzar la Tercera Forma Normal (3FN), una etapa del diseño de bases de datos que asegura que todas las columnas de una tabla dependan únicamente de la clave primaria, eliminando así redundancias y dependencias transitivas. En una fase inicial, se planteó la existencia de entidades separadas para Cliente y Administrador (socio); sin embargo, tras el análisis de dependencias y con el objetivo de evitar duplicidad de datos personales, ambas se unificaron en una única entidad **USUARIOS**, incorporando el tipo de usuario como un

atributo. De igual forma, se descartó una entidad independiente para categorías al comprobar que no aportaba valor funcional en el alcance actual del proyecto.

El uso de claves foráneas garantiza la integridad referencial entre las entidades, asegurando que no existan pedidos sin usuario asociado ni detalles de pedido sin un pedido o producto válido. Además, la estructura general del modelo se ha diseñado teniendo en cuenta posibles ampliaciones futuras, como la incorporación de entidades para cajas preparadas, envíos (gestionados por agentes externos) o pagos, sin que estas extensiones supongan una modificación de la estructura principal. Al no formar parte del alcance funcional de esta fase de desarrollo, dichas entidades no se han incluido en el modelo final, aunque se han tenido en cuenta a nivel de diseño.