



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Trabajo Práctico Final

BASE DE DATOS (75.15 / 95.05 / TA044)

Docentes:

Cabrera José Luís (Profesor Adjunto)
Bernardi Adrián Sergio (Jefe de Trabajos Prácticos)

2do Cuatrimestre 2024

Grupo N° 2

Integrantes:

Juan Bautista Botter Brun - 101505

Patricio Ibar - 109569

Valentina Llanos Pontaut - 104413

Valentín Marturet - 104526

Índice

1	Introducción	2
2	Supuestos	2
3	Modelo Entidad-Relación	3
3.1	Conjuntos de Entidades	3
3.2	Relaciones	5
4	Pasaje al Modelo Relacional	6
4.1	Comprador y Vendedor	6
4.2	Publicación	6
4.3	Característica	6
4.4	Compra	7
5	SQL	8
5.1	Creación de Tablas	8
5.1.1	Actualización de precio mínimo	9
5.2	Establecimiento de permisos	10
5.2.1	Administrador	10
5.2.2	Vendedor	10
5.2.3	Comprador	10
5.3	Ejemplos de inserción	11
5.4	Consultas varias	11

1 Introducción

El presente trabajo práctico aborda la construcción de un sistema de base de datos diseñado para un sitio web de compra y venta. Este sistema permitirá a los usuarios realizar diversas consultas, siempre y cuando cuenten con los permisos necesarios.

En términos generales, los usuarios que actúan como **vendedores** podrán crear publicaciones sobre productos, mientras que otros usuarios, en su rol de **compradores**, tendrán la posibilidad de adquirir dichos productos. Además, el sistema gestionará información clave sobre el estado de las compras realizadas y el envío de los productos adquiridos.

El desarrollo del trabajo se estructura en varias etapas:

1. En primer lugar, se presenta el diseño del modelo **Entidad-Relación**, que sirvió como base para la construcción del modelo **Relacional**.
2. En la segunda parte del informe, se explica el diseño del modelo **Relacional**, derivado del modelo Entidad-Relación.
3. Finalmente, a partir de este diseño, se implementó la base de datos en **PostgreSQL**, permitiendo la generación y consulta de datos mediante SQL.

Este proyecto busca ofrecer una solución robusta y eficiente para la administración de un sitio web de compra y venta, brindando funcionalidades esenciales tanto para vendedores como para compradores.

2 Supuestos

A continuación algunos supuestos que se hicieron para diseñar el modelo sobre el problema que se presenta en el enunciado.

- Un vendedor puede tener sólo una publicación con un precio para un mismo producto.
- Si un vendedor se queda sin stock de un producto, debe marcar como "no disponible" esa publicación.
- Los productos cuyo estado de envío es **RECIBIDO**, no pueden tener estado de compra **CANCELADO**.
- Los compradores pueden comprar mas de una vez la misma publicación en distintas fechas.

3 Modelo Entidad-Relación

A continuación, se presenta el diagrama de Entidad-Relación elaborado a partir de la interpretación del enunciado:

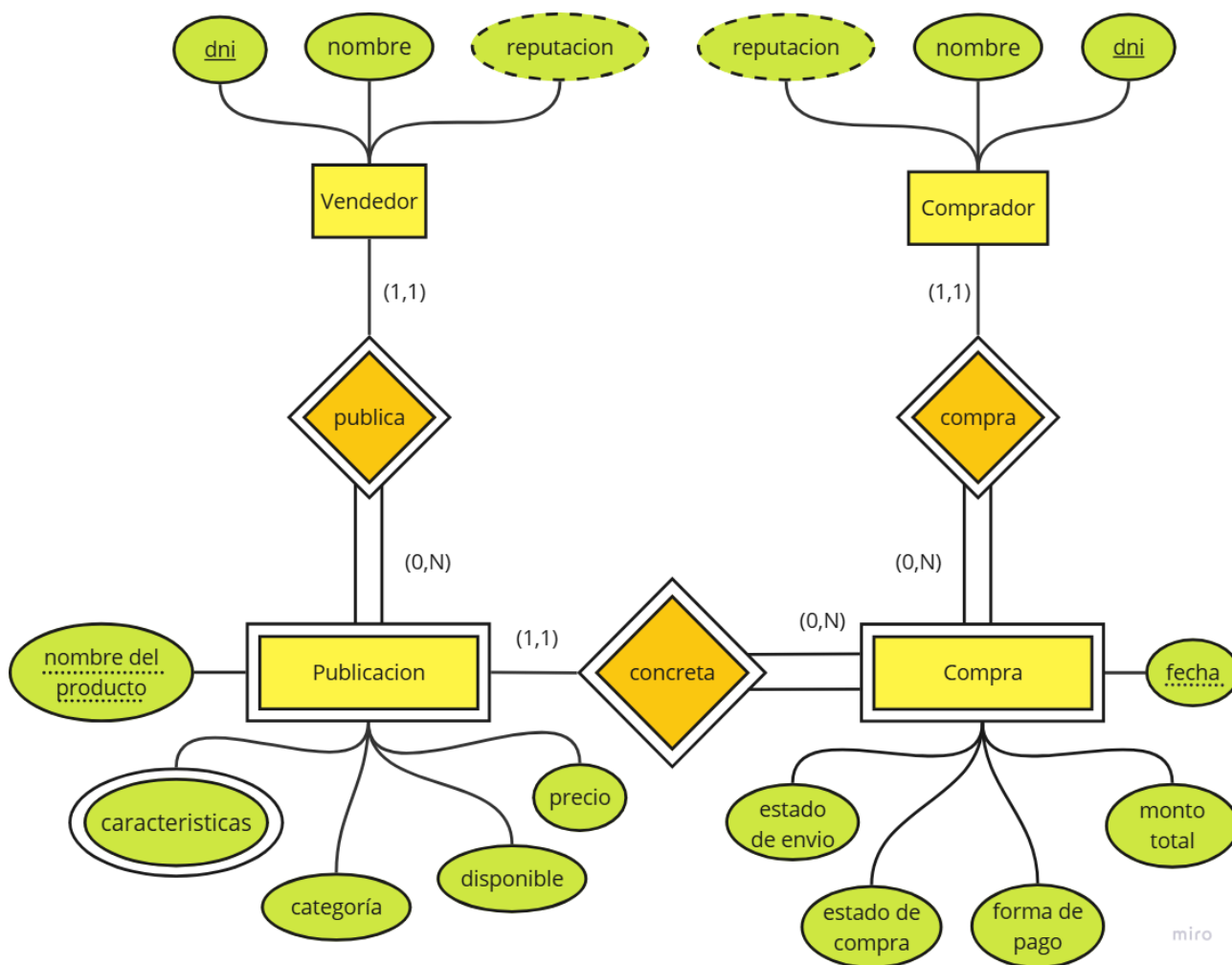


Figure 1: Diagrama del Modelo Entidad-Relación.

En este diagrama:

- Las entidades se representan mediante rectángulos amarillos.
- Las relaciones se muestran como rombos naranjas.
- Los atributos se representan por óvalos verdes; aquellos que están subrayados forman parte de la clave primaria de la entidad; los subrayados con líneas punteadas son atributos discriminantes de una clase débil.
- Las entidades y relaciones con doble borde son débiles.

3.1 Conjuntos de Entidades

A continuación se describen los conjuntos de entidades propuestos en el modelo. Utilizamos el concepto "entidades" como sinónimo de "conjunto de entidades" para facilitar la lectura.

- **Comprador y Vendedor:** Estas entidades representan a los usuarios que utilizan la web de compra-venta.

A continuación una lista de los atributos de esta entidad.

- **DNI:** Identificador único (clave primaria).
- **Nombre:** Nombre del usuario.
- **Reputación:** Atributo derivado que se calcula en base a sus transacciones (compras o ventas). Para este trabajo, se definió que la reputación se calcula en base al monto_total de las compras del ultimo mes en las que ese usuario fue comprador/vendedor

- **Publicación:** Esta entidad débil representa los productos en venta de un **Vendedor**. Cada vendedor sólo puede tener una publicación para un producto determinado con un nombre, es por esto que el atributo *nombre* de esta entidad es su *atributo discriminante*.

A continuación una lista de los atributos de esta entidad.

- **Nombre:** Nombre del producto (atributo discriminante).
- **Características:** Atributo multivaluado que describe detalles específicos del producto (estado, color, etc.).
- **Categoría:** Atributo asocia el producto a una categoría.
- **Disponible:** Indica si el producto aún está disponible para la venta.

La **clave primaria** de esta entidad se compone de la clave primaria de su entidad fuerte y de su atributo discriminante. Esto es el **dni del vendedor** y el **nombre del producto**.

- **Compra:** Esta entidad débil representa las compras realizadas por un **Comprador**. La entidad compra es débil respecto de la entidad **Comprador** y también respecto de la entidad **Producto**.

A continuación una lista de los atributos de esta entidad.

- **Fecha:** Fecha de la compra (atributo discriminante).
- **Monto total:** Precio pagado al momento de la compra.
- **Estado de compra:** Puede ser:
 - * **COMPRADO:** Compra exitosa.
 - * **CANCELADO:** Compra anulada; la publicación vuelve a estar disponible.
- **Estado de envío:** Puede ser:
 - * **EN_CAMINO:** Producto en proceso de envío.
 - * **RECIBIDO:** Producto recibido por el comprador.
 - * **CANCELADO:** Envío cancelado.
- **Medio de pago:** Puede ser:
 - * **EFFECTIVO**
 - * **DÉBITO**
 - * **CRÉDITO**

Como la clave primaria de las entidades débiles se componen de las claves primarias de sus entidades fuertes y de su atributo discriminante, la **clave primaria** de esta entidad se compone por: **dni del vendedor, nombre del producto, dni del comprador y fecha de la compra**.

Dado este diseño, un usuario sólo puede comprar una vez un producto de una misma publicación en la misma fecha.

3.2 Relaciones

Las relaciones existentes en este modelo son sólo las que definen las entidades débiles. Estas son la relación **publica** que define la entidad débil **Publicación** a partir de la entidad fuerte **Vendedor**, y las relaciones **compra** y **concreta** que definen la entidad débil **Compra** a partir de las entidades **Comprador** y **Publicación**, respectivamente.

4 Pasaje al Modelo Relacional

4.1 Comprador y Vendedor

Las relaciones **Comprador** y **Vendedor** están compuestas por los atributos **dni** y **nombre**. El atributo **reputación** al ser derivado no se incluye como parte de la relación, sino que es calculado en base a otros datos.

En este caso, **dni** actúa como clave primaria (primary key), lo que asegura la unicidad de los registros en la base de datos. La fórmula que representa esta relación es:

$$\text{Vendedor}(\underline{\text{dni}}, \text{nombre})$$
$$\text{Comprador}(\underline{\text{dni}}, \text{nombre})$$

Donde:

- **dni** es la clave primaria, que identifica de forma única a cada usuario en el sistema.
- **nombre** es un atributo que almacena el nombre del usuario.

4.2 Publicación

La relación **Publicación** está definida por los atributos **producto** y **vendedor** como clave primaria, y los atributos **precio**, **categoría** y **disponible** como atributos adicionales.

$$\text{Publicación}(\underline{\text{producto}}, \underline{\text{vendedor}}, \text{precio}, \text{categoría}, \text{disponible})$$

Donde:

- **producto** es el nombre del producto que se está vendiendo.
- **vendedor** es el dni del vendedor que creó la publicación. Es una clave foránea para la relación **Vendedor**.
- **precio** es el precio por unidad del producto que se está vendiendo.
- **categoría** es la categoría a la que está asociado el producto vendido.
- **disponible** es la disponibilidad del atributo. Este atributo puede tomar como valores **verdadero** o **falso**, dependiendo si el vendedor está actualmente dispuesto a hacer una venta descrita por la **Publicación**.

4.3 Característica

La relación **Característica** se corresponde con el atributo multivaluado del mismo nombre del Modelo Entidad-Relación. Esta entidad se define mediante los atributos **característica**, **producto** y **vendedor**, los cuales, en conjunto, forman la clave primaria de esta relación. Los atributos **producto** y **vendedor** actúan como claves foráneas que hacen referencia a la entidad **Publicación**. La fórmula que representa esta relación es:

$$\text{Característica}(\underline{\text{producto}}, \underline{\text{vendedor}}, \underline{\text{característica}})$$

Donde:

- **característica** representa las propiedades del producto, como puede ser su estado, tamaño, color, etc.
- **producto** es una clave foránea que hace referencia al producto asociado a esta característica.
- **vendedor** es una clave foránea que indica al usuario vendedor responsable de la publicación.

4.4 Compra

Esta relación registra todas las compras realizadas por los usuarios compradores a partir de las publicaciones disponibles. La clave primaria de esta relación está compuesta por los atributos **producto**, **vendedor** (ambos clave foránea que hacen referencia a la relación **Publicación**), **comprador** (clave foránea de la relación **Comprador**) y **fecha_compra**. Además, incluye los atributos **forma_pago**, **estado_compra**, **estado_envío** y **monto_total**. La fórmula que representa esta relación es:

Compra(producto, vendedor, comprador, fecha_compra,
estado_envío, forma_pago, monto_total)

Donde:

- **producto** es parte de una clave foránea que hace referencia a la publicación asociada con esta transacción.
- **vendedor** es una clave foránea que identifica al usuario vendedor dueño de la publicación del producto comprado.
- **comprador** clave foránea del usuario que realizó la compra del producto
- **fecha_de_compra** indica la fecha en la que el comprador adquirió el producto.
- **forma_de_pago** detalla el método de pago utilizado en la transacción, como tarjeta de crédito, transferencia bancaria, etc.
- **estado_compra** Indica el estado actual de la transacción. Los valores posibles son:
 - **EN_VENTA**: El producto está disponible para su compra.
 - **COMPRADO**: El producto ha sido adquirido por un comprador.
 - **CANCELADO**: La publicación ha sido cancelada.
- **estado_envio** Describe el estado del envío del producto adquirido. Los valores posibles son:
 - **EN_CAMINO**: El producto está en proceso de ser entregado.
 - **RECIBIDO**: El comprador ha recibido el producto.
 - **CANCELADO**: El envío del producto fue cancelado.
- **monto_total** Indica el monto que debe pagar el comprador. Se corresponde con el precio de la publicación relacionada al momento de la compra.

5 SQL

En esta sección, se detallará el proceso de creación y configuración de la base de datos en **PostgreSQL**. Los archivos SQL necesarios para la implementación están disponibles en el siguiente repositorio: <https://github.com/BBDD-Cabrera-Grupo2/BBDD-web-compra-venta>. Sin embargo, a continuación se presenta un resumen paso a paso de las consultas que se ejecutaron para alcanzar el objetivo final.

Para comenzar, se creó la base de datos `web_compra_venta` utilizando la consola de `psql` con la configuración predeterminada.

```
CREATE DATABASE web_compra_venta;
```

Figure 2: Creación de la base de datos

5.1 Creación de Tablas

A continuación se muestran las consultas SQL propuestas para crear las tablas correspondientes a cada relación del modelo relacional propuesto.

```
CREATE TABLE comprador (  
    dni INT PRIMARY KEY,  
    nombre VARCHAR(200)  
);  
  
CREATE TABLE vendedor (  
    dni INT PRIMARY KEY,  
    nombre VARCHAR(200)  
);
```

Figure 3: Creación de las tablas de **Compradores** y **Vendedores**

```
CREATE TABLE publicacion (  
    vendedor INT,  
    producto VARCHAR(200),  
    categoria VARCHAR(200),  
    precio DECIMAL(10, 2),  
    disponible BOOLEAN,  
    PRIMARY KEY (vendedor, producto),  
    FOREIGN KEY (vendedor) REFERENCES vendedor(dni),  
    CHECK (precio > 0)  
);
```

Figure 4: Creación de la tabla de **Publicaciones**

Para la tabla de publicaciones se agrega la siguiente **constraint**, que verifica que las publicaciones disponibles deben tener un precio mayor a un mínimo.

```
ALTER TABLE publicacion
ADD CONSTRAINT publicacion_precio_min
CHECK (precio > 900 OR NOT disponible);
```

Figure 5: Creación de la constraint de **precios mínimos**

```
CREATE TABLE caracteristica (
    caracteristica VARCHAR(200),
    producto VARCHAR(200),
    vendedor INT,
    PRIMARY KEY (caracteristica, producto, vendedor),
    FOREIGN KEY (producto, vendedor)
    REFERENCES publicacion(producto, vendedor)
);
```

Figure 6: Creación de la tabla de **Características**

```
CREATE TABLE compra (
    vendedor INT,
    producto VARCHAR(200),
    comprador INT,
    estado_envio VARCHAR(50),
    estado_compra VARCHAR(50),
    fecha_compra DATE,
    forma_pago VARCHAR(50),
    monto_total DECIMAL(10, 2),
    PRIMARY KEY (comprador, vendedor, producto, fecha_compra),
    FOREIGN KEY (vendedor, producto)
    REFERENCES publicacion(vendedor, producto),
    FOREIGN KEY (comprador)
    REFERENCES comprador(dni),
    CHECK (
        NOT (
            estado_compra = 'CANCELADO'
            AND estado_envio = 'RECIBIDO'
        )
    )
);
```

Figure 7: Creación de la tabla de **Compras**

5.1.1 Actualización de precio mínimo

El valor del precio mínimo que deben tener las **publicaciones disponibles** sólo puede modificarse a un valor menor que el precio de la publicación más barata. De otra manera, el SGBD fallará al intentar hacer la consulta correspondiente.

Esto puede solucionarse por parte del administrador, por ejemplo marcando las publicaciones menores al nuevo precio mínimo que se quiere establecer como no disponibles.

```
ALTER TABLE publicacion
DROP CONSTRAINT publicacion_precio_min;

ALTER TABLE publicacion
ADD CONSTRAINT publicacion_precio_min
CHECK (precio > 900 OR NOT disponible);
```

Figure 8: Actualización del precio mínimo

5.2 Establecimiento de permisos

5.2.1 Administrador

Los usuarios de la base de datos con rol **admin** son los encargados de crear y asignar permisos a los compradores y vendedores. Tienen permisos de acceso total sobre la base de datos.

```
CREATE ROLE admin LOGIN PASSWORD 'admin_pass' CREATEROLE;
GRANT ALL PRIVILEGES ON SCHEMA public TO admin;
```

Figure 9: Creación del rol **admin**

5.2.2 Vendedor

Los usuarios con rol **vendedor** pueden crear y modificar sus propias publicaciones de venta. Esto incluye control sobre las tablas de **publicaciones** y de **características** de las mismas. También puede hacer consultas de lectura sobre las tablas de **compras**, de manera que pueda saber cuántos productos vendió.

```
CREATE ROLE vendedor LOGIN PASSWORD 'vendedor_pass';
GRANT
    SELECT, UPDATE, INSERT ON TABLE publicacion TO vendedor;
GRANT
    SELECT, UPDATE, INSERT, DELETE ON TABLE caracteristica TO vendedor;
GRANT
    SELECT ON TABLE compra TO vendedor;
```

Figure 10: Creación del rol **vendedor**

Notar que el vendedor no puede hacer operaciones de **DELETE** sobre las publicaciones. Esto si esa publicación tiene una compra, la compra perdería la referencia del producto al que fue comprado. El vendedor sí puede hacer un **UPDATE** de sus publicaciones para marcarlas como *no disponibles*, modificando el atributo correspondiente.

5.2.3 Comprador

Los usuarios con rol **comprador** pueden crear y modificar sus compras. Esto incluye control sobre las tablas de **compras**, y poder consultar las tablas de **publicaciones** para saber qué productos están disponibles para su venta.

```
CREATE ROLE comprador LOGIN PASSWORD 'comprador_pass';
GRANT
    SELECT, UPDATE, INSERT ON TABLE compra TO comprador;
GRANT
    SELECT ON TABLE publicacion TO comprador;
```

Figure 11: Creación del rol **comprador**

5.3 Ejemplos de inserción

Para popular nuestra base de datos, se insertaron varios valores. Los mismos pueden verse en el archivo `insert_examples.sql` del repositorio.

5.4 Consultas varias

Para esta sección mostraremos las consultas **SQL** que se solicitan en el enunciado. También se incluirán los resultados de correr dichas consultas en **PostgreSQL** las tablas creadas previamente, y los datos de ejemplo insertados.

```
SELECT DISTINCT producto FROM publicacion;
```

Figure 12: Consulta - Listar todos los productos de la plataforma.

```
SELECT *
FROM publicacion p
WHERE categoria = 'Tecnología';
```

Figure 13: Consulta - Listar productos de la plataforma por categoría (categoría ejemplo: 'Tecnología')

```
INSERT INTO publicacion
(vendedor, producto, categoria, precio, disponible)
VALUES (98765432, 'Bicicleta', 'Deportes', 154000, TRUE);
```

Figure 14: Consulta - Crear una publicación para vender.

```
UPDATE publicacion
SET precio = 160000
WHERE vendedor = 98765432
AND producto = 'Bicicleta';
```

Figure 15: Consulta - Actualizar una publicación de venta.

```
INSERT INTO compra
(vendedor, producto, comprador, estado_envio, estado_compra, fecha_compra, forma_pago)
VALUES
(98765432, 'Bicicleta', 71717171, 'EN PREPARACION', 'COMPRADO', '2024-11-25', 'EFECTIVO');
```

Figure 16: Consulta - Realizar una compra.

```
UPDATE compra
SET estado_envio = 'RECIBIDO'
WHERE vendedor = 98765432 AND producto = 'Bicicleta'
AND comprador = 71717171 AND fecha_compra = '2024-11-25';
```

Figure 17: Consulta - Guardar un envío.

```
UPDATE compra
SET estado_compra = 'CANCELADO', estado_envio = 'CANCELADO'
WHERE vendedor = 98765432
AND producto = 'Bicicleta'
AND comprador = 71717171
AND fecha_compra = '2024-11-25';
```

Figure 18: Consulta - Cancelar una compra.

```
UPDATE compra
SET estado_envio = 'CANCELADO'
WHERE vendedor = 98765432 AND producto = 'Bicicleta'
AND comprador = 71717171 AND fecha_compra = '2024-11-25';
```

Figure 19: Consulta - Cancelar un envío.

```
WITH compras_ultimos_30_dias AS (
    SELECT comprador, SUM(monto_total) AS total_gastado
    FROM compra
    WHERE fecha_compra >= date_trunc('month', CURRENT_DATE) - INTERVAL '1 month'
    AND estado_compra = 'COMPRADO'
    GROUP BY comprador
)
SELECT c.dni, c.nombre, co.total_gastado AS reputacion
FROM comprador c
JOIN compras_ultimos_30_dias co ON c.dni = co.comprador;
```

Figure 20: Consulta - Ver las reputaciones de los compradores

```

WITH ventas_ultimos_30_dias AS (
  SELECT vendedor, SUM(monto_total) AS total_cobrado
  FROM compra
  WHERE fecha_compra >= date_trunc('month', CURRENT_DATE) - INTERVAL '1 month'
  AND estado_compra = 'COMPRADO'
  GROUP BY vendedor
)
SELECT v.dni, v.nombre, ve.total_cobrado AS reputacion
FROM vendedor v
JOIN ventas_ultimos_30_dias ve ON v.dni = ve.vendedor;

```

Figure 21: Consulta - Ver las reputaciones de los vendedores

```

SELECT COUNT(*) AS cantidad_de_ventas
FROM compra
WHERE fecha_compra < '2024-11-28'
AND fecha_compra > '2024-11-01'
AND estado_compra = 'COMPRADO';

```

Figure 22: Consulta - Contabilizar la cantidad de ventas de plataforma para un rango de fechas.

```

SELECT SUM(monto_total) AS valor_total_ventas
FROM compra
WHERE fecha_compra < '2024-11-28'
AND fecha_compra > '2024-11-01'
AND estado_compra = 'COMPRADO';

```

Figure 23: Consulta - Contabilizar el valor total de ventas de la plataforma para un rango de fechas.