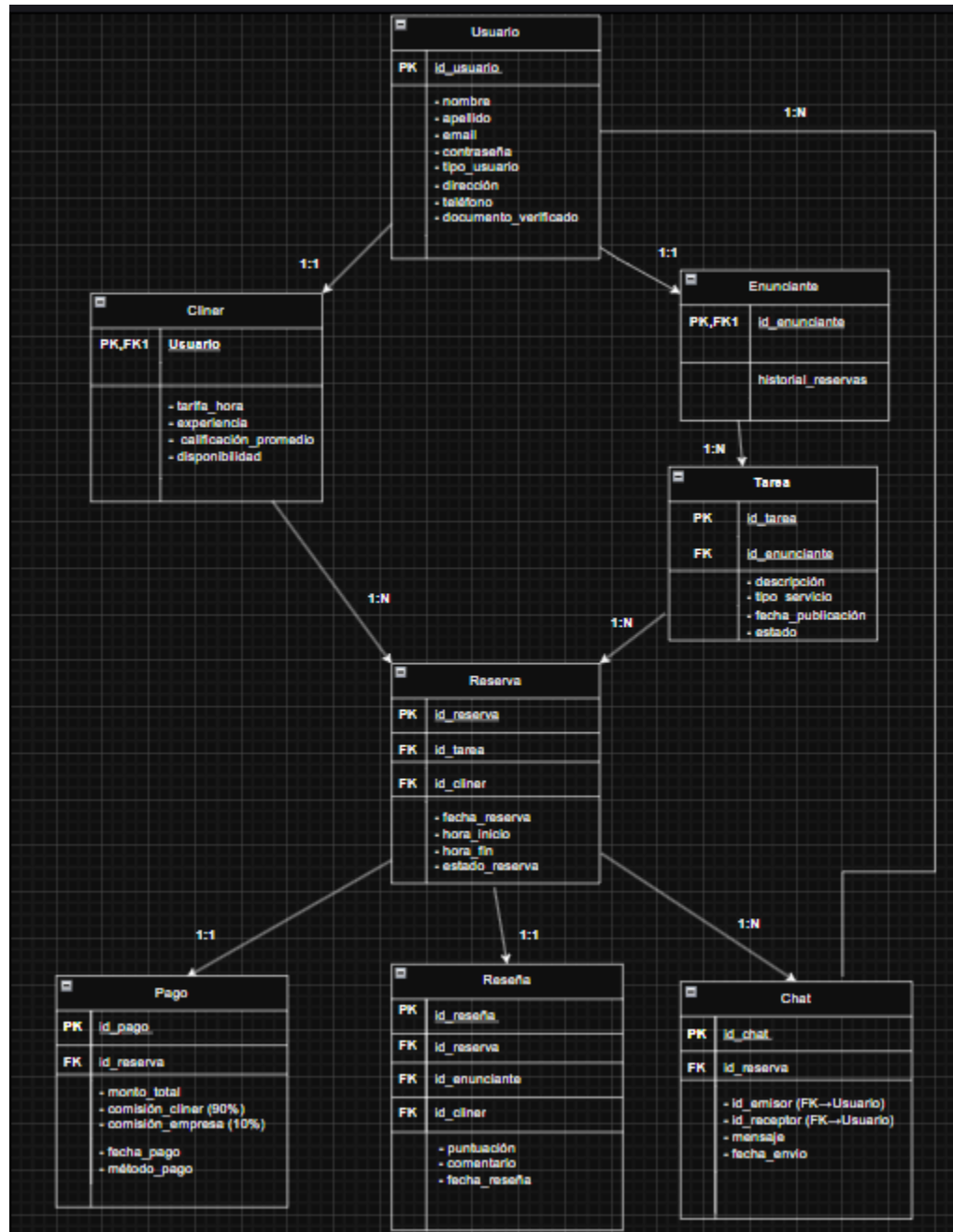


# Cliner

## DER



## Script:

-- 1. Tabla Usuario

```
CREATE TABLE Usuario (  
    id_usuario SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido VARCHAR(100) NOT NULL,  
    email VARCHAR(150) UNIQUE NOT NULL,  
    contraseña VARCHAR(200) NOT NULL,  
    tipo_usuario VARCHAR(20) CHECK (tipo_usuario IN ('cliner','enunciante'))  
NOT NULL,  
    direccion TEXT,  
    telefono VARCHAR(30),  
    documento_verificado BOOLEAN DEFAULT FALSE  
);
```

-- 2. Tabla Enunciante (subtipo de Usuario)

```
CREATE TABLE Enunciante (  
    id_enunciante INT PRIMARY KEY REFERENCES Usuario(id_usuario) ON DELETE  
CASCADE,  
    historial_reservas INT DEFAULT 0  
);
```

-- 3. Tabla Cliner (subtipo de Usuario)

```
CREATE TABLE Cliner (  
    id_cliner INT PRIMARY KEY REFERENCES Usuario(id_usuario) ON DELETE CASCADE,  
    tarifa_hora NUMERIC(10,2),  
    experiencia TEXT,  
    calificacion_promedio NUMERIC(3,2) DEFAULT 0,  
    disponibilidad TEXT  
);
```

-- 4. Tabla Tarea

```
CREATE TABLE Tarea (  
    id_tarea SERIAL PRIMARY KEY,  
    id_enunciante INT NOT NULL REFERENCES Enunciante(id_enunciante) ON DELETE  
CASCADE,  
    descripcion TEXT NOT NULL,  
    tipo_servicio VARCHAR(100) NOT NULL,  
    fecha_publicacion TIMESTAMP DEFAULT NOW(),  
    estado VARCHAR(20) DEFAULT 'pendiente'  
);
```

```

-- 5. Tabla Reserva
CREATE TABLE Reserva (
    id_reserva SERIAL PRIMARY KEY,
    id_tarea INT NOT NULL REFERENCES Tarea(id_tarea) ON DELETE CASCADE,
    id_cliner INT NOT NULL REFERENCES Cliner(id_cliner) ON DELETE CASCADE,
    fecha_reserva DATE NOT NULL,
    hora_inicio TIME NOT NULL,
    hora_fin TIME NOT NULL,
    estado_reserva VARCHAR(20) DEFAULT 'pendiente'
);

-- 6. Tabla Pago
CREATE TABLE Pago (
    id_pago SERIAL PRIMARY KEY,
    id_reserva INT UNIQUE NOT NULL REFERENCES Reserva(id_reserva) ON DELETE
CASCADE,
    monto_total NUMERIC(10,2) NOT NULL,
    comision_cliner NUMERIC(10,2),
    comision_empresa NUMERIC(10,2),
    fecha_pago TIMESTAMP DEFAULT NOW(),
    metodo_pago VARCHAR(50)
);

-- 7. Tabla Reseña
CREATE TABLE Reseña (
    id_reseña SERIAL PRIMARY KEY,
    id_reserva INT UNIQUE NOT NULL REFERENCES Reserva(id_reserva) ON DELETE
CASCADE,
    id_enunciante INT NOT NULL REFERENCES Enunciante(id_enunciante) ON DELETE
CASCADE,
    id_cliner INT NOT NULL REFERENCES Cliner(id_cliner) ON DELETE CASCADE,
    puntuacion INT CHECK (puntuacion BETWEEN 1 AND 5),
    comentario TEXT,
    fecha_reseña TIMESTAMP DEFAULT NOW()
);

-- 8. Tabla Chat
CREATE TABLE Chat (
    id_chat SERIAL PRIMARY KEY,
    id_reserva INT NOT NULL REFERENCES Reserva(id_reserva) ON DELETE CASCADE,
    id_emisor INT NOT NULL REFERENCES Usuario(id_usuario) ON DELETE CASCADE,
    id_receptor INT NOT NULL REFERENCES Usuario(id_usuario) ON DELETE CASCADE,

```

```

    mensaje TEXT NOT NULL,
    fecha_envio TIMESTAMP DEFAULT NOW()
);

-- =====
-- INDICES EXTRA PARA OPTIMIZACIÓN
-- =====

CREATE INDEX idx_usuario_email ON Usuario(email);
CREATE INDEX idx_tarea_estado ON Tarea(estado);
CREATE INDEX idx_reserva_estado ON Reserva(estado_reserva);
CREATE INDEX idx_chat_reserva ON Chat(id_reserva);

```

## Explicación completa del DER de Cliner

### 1. Usuario (base de todo)

- La entidad **Usuario** es la raíz del sistema.
- Cada persona que entra a Cliner debe estar registrada: ya sea como **enunciante** (cliente que pide limpieza) o como **cliner** (trabajador autónomo que realiza la tarea).
- Por eso, Usuario guarda datos básicos: nombre, email, contraseña, dirección, teléfono y verificación de identidad.
- De aquí se desprenden **dos subtipos**:
  - **Enunciante** → cliente.
  - **Cliner** → freelancer.

Esto refleja que ambos son parte de la misma plataforma, pero tienen roles distintos y funcionalidades diferentes en la app.

---

### 2. Enunciante y Cliner (subtipos de Usuario)

- **Enunciante**: representa a la persona que **publica tareas**. Tiene su historial de reservas y de pagos.

- **Cliner:** representa al **trabajador autónomo**. Además de los datos básicos, agrega atributos propios:
  - tarifa por hora
  - experiencia
  - calificación promedio
  - disponibilidad
- Esto está alineado con lo que contaste: los cliners son autónomos, ponen sus precios, eligen cuándo trabajan y pueden aceptar/rechazar tareas.

La relación **Usuario – Enunciante/Cliner** es 1:1, porque cada usuario es exclusivamente uno de los dos roles.

---

### 3. Tarea

- Es la entidad que **inicia todo el flujo**.
- Una tarea la publica un **enunciante** (relación Enunciante → Tarea es 1:N, ya que un cliente puede publicar varias tareas).
- Contiene:
  - descripción
  - tipo de servicio (ejemplo: limpieza general, vidrios, colchones, etc.)
  - fecha de publicación
  - estado (pendiente, en proceso, finalizada).
- Es como el **anuncio** en el marketplace.

Representa el corazón de Cliner: lo que el cliente necesita resolver.

---

### 4. Reserva

- Una vez publicada la tarea, los cliners pueden postularse o ser seleccionados.
- Eso genera una **Reserva**, que une:
  - una **Tarea** (qué se hace)
  - un **Cliner** (quién la hace).
- Por eso, Reserva tiene dos claves foráneas:
  - id\_tarea (FK)
  - id\_cliner (FK)
- Además, guarda: fecha, horario de inicio y fin, y el estado de la reserva (aceptada, rechazada, completada).

Es lo que en la práctica confirma el “match” entre cliente y cliner.

---

## 5. Pago

- Cada Reserva tiene asociado un **Pago** (relación 1:1).
- Pago guarda:
  - monto total
  - comisión para el cliner (85%)
  - comisión para Cliner app (15%)
  - método de pago (tarjeta, transferencia, etc.)
  - fecha.
- Esto refleja lo que me contaste: **no hay efectivo**, todo va por la app y se factura automáticamente.

Es el paso que asegura monetización y sostenibilidad del modelo de negocio.

---

## 6. Reseña

- Cuando termina una reserva, el enunciante puede dejar una **Reseña**.
- Relaciona:
  - la **Reserva**
  - el **Enunciante** (que califica)
  - el **Cliner** (que recibe la calificación).
- Incluye puntuación, comentario y fecha.
- Sirve para que los cliners con más calificación puedan cobrar más caro y generar confianza en futuros clientes.

Es el mecanismo de **feedback y reputación**, clave en un marketplace.

---

## 7. Chat

- Representa la **comunicación interna** dentro de la plataforma.
- Cada Chat está vinculado a:
  - una **Reserva** (para saber sobre qué tarea hablan),
  - un **Usuario emisor** y un **Usuario receptor**.
- Guarda mensajes y fecha de envío.
- Esto evita malentendidos y asegura que todos los detalles (precio final, aclaraciones de la tarea) queden registrados en la app.

Refleja lo que mencionaste: antes de aceptar, cliente y cliner pueden conversar dentro de la aplicación.

---

## Cómo fluye todo el sistema (vista desde el DER)

1. **Un Enunciante crea un Usuario** → luego publica una **Tarea**.
  2. **Cliners (Usuarios)** ven la tarea y uno de ellos es seleccionado → se genera una **Reserva** que conecta a ambos.
  3. Cliente y cliner pueden hablar en el **Chat** antes de confirmar.
  4. Una vez realizada la tarea, se procesa el **Pago** → Cliner recibe 85%, la app retiene 15%.
  5. El Enunciante deja una **Reseña** sobre el Cliner → esto actualiza la reputación del trabajador.
- 

## Por qué el DER refleja fielmente tu proyecto

- El **Usuario dividido en dos roles** refleja perfectamente la interfaz doble de la app (enunciante y cliner).
- La **Tarea** es el punto de inicio de todo (publicación).
- La **Reserva** es la unión de la oferta y demanda (el “match”).
- El **Pago** asegura monetización del modelo y transparencia en la facturación.
- La **Reseña** y el **Chat** sostienen confianza, reputación y seguridad.

En conjunto, este DER no solo organiza los datos, sino que también **muestra el modelo de negocio de Cliner en la práctica**: un marketplace dual que conecta oferta y demanda de limpieza, asegurando seguridad, reputación y pago online.

## Entidades y atributos principales

- **Usuario**
  - id\_usuario (PK)
  - nombre



- apellido
- email
- contraseña
- tipo\_usuario (Enunciante / Cliner)
- dirección
- teléfono
- documento\_verificado (sí/no)
- **Cliner (subtipo de Usuario)**
  - id\_cliner (PK, FK → Usuario)
  - tarifa\_hora
  - experiencia
  - calificación\_promedio
  - disponibilidad
- **Enunciante (subtipo de Usuario)**
  - id\_enunciante (PK, FK → Usuario)
  - historial\_reservas
- **Tarea**
  - id\_tarea (PK)
  - id\_enunciante (FK → Enunciante)
  - descripción
  - tipo\_servicio (limpieza general, vidrios, tapicería, etc.)
  - fecha\_publicación
  - estado (pendiente / en proceso / completada / cancelada)
- **Reserva**
  - id\_reserva (PK)

- id\_tarea (FK → Tarea)
- id\_cliner (FK → Cliner)
- fecha\_reserva
- hora\_inicio
- hora\_fin
- estado\_reserva (aceptada / rechazada / finalizada)

- **Pago**

- id\_pago (PK)
- id\_reserva (FK → Reserva)
- monto\_total
- comisión\_cliner (90%)
- comisión\_empresa (10%)
- fecha\_pago
- método\_pago (tarjeta, transferencia, etc.)

- **Reseña**

- id\_reseña (PK)
- id\_reserva (FK → Reserva)
- id\_enunciante (FK → Enunciante)
- id\_cliner (FK → Cliner)
- puntuación (1–5)
- comentario
- fecha\_reseña

- **Chat**

- id\_chat (PK)
- id\_reserva (FK → Reserva)

- id\_emisor (FK → Usuario)
  - id\_receptor (FK → Usuario)
  - mensaje
  - fecha\_envio
- 

## Relaciones principales

- **Un Enunciante** → puede **publicar muchas Tareas**.
- **Un Cliner** → puede estar en **muchas Reservas**.
- **Una Tarea** → genera **una o varias Reservas** (dependiendo de si varios cliners aplican, pero sólo una se concreta).
- **Una Reserva** → tiene **un Pago** y puede tener **una Reseña**.
- **Un Chat** → se da entre **Enunciante y Cliner** dentro de una Reserva.