Arquitectura: JWT como Sistema Intermedio

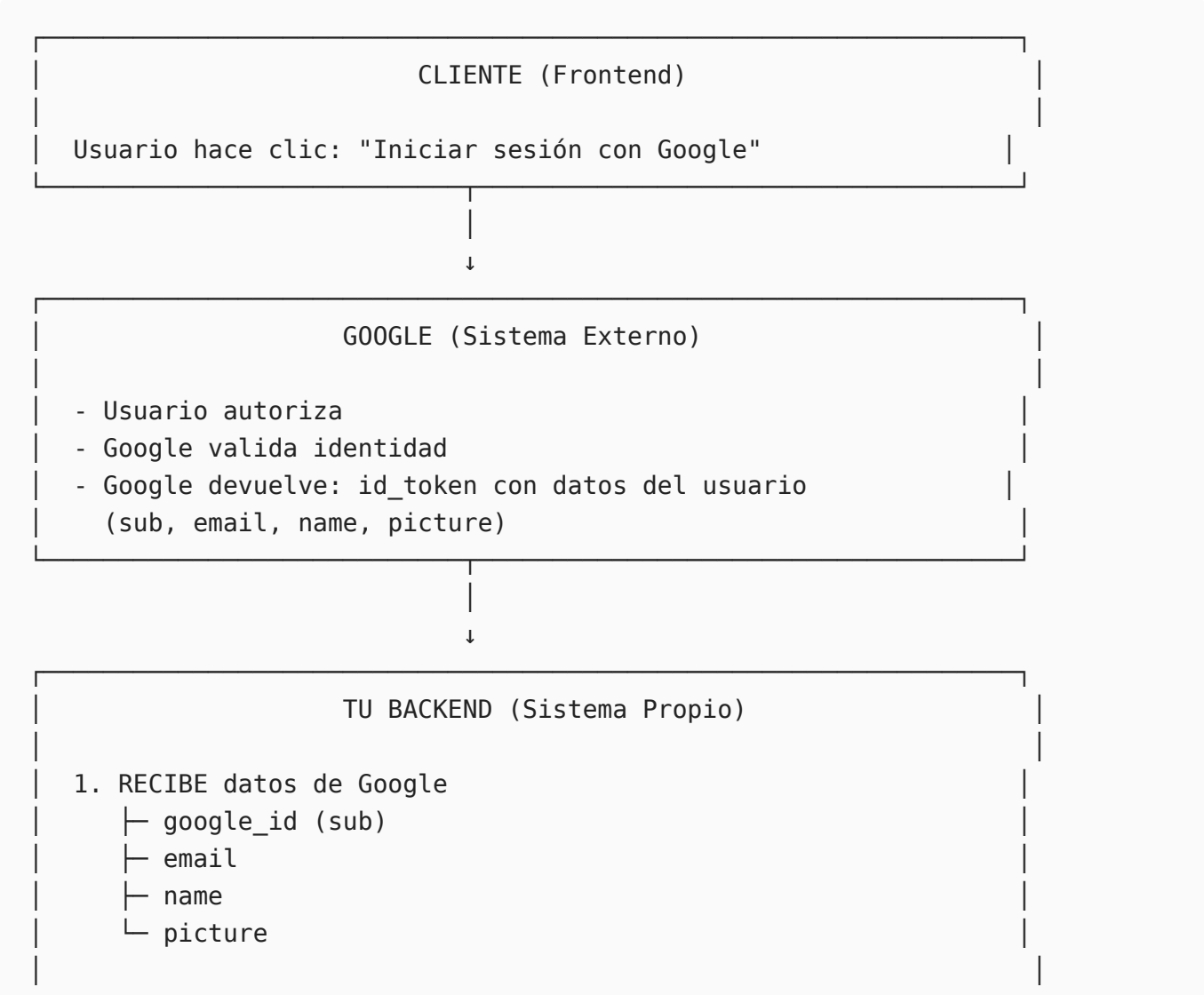
Plano arquitectónico + DER combinado sobre autenticación con Google OAuth y JWT.

🎯 Concepto: JWT como Sistema Intermedio

Sí, JWT es un INTERMEDIARIO/PUENTE entre:

- Tu sistema (Backend + BD)
- El cliente (Frontend)

📐 Plano Arquitectónico Conceptual - Registro/Login




2. CONSULTA/MODIFICA Base de Datos
├─ Busca usuario por google_id
├─ Si NO existe → Crea nuevo usuario
└─ Si existe → Actualiza last_login
3. GENERA Sistema Intermedio (JWT)
├─ Crea token con: user_id, email, role
├─ Firma el token (para seguridad)
└─ El JWT es como una "credencial portable"
4. DEVUELVE al cliente
├─ JWT (token)
└─ Datos del usuario (id, email, name, picture, role)
SISTEMA INTERMEDIO (JWT)
El JWT viaja entre Cliente ↔ Backend
Contiene:
├─ Quién es el usuario (user_id)
├─ Qué puede hacer (role)
├─ Cuándo expira
└─ Firma de seguridad
Es como una "tarjeta de identidad digital"
CLIENTE (Frontend)
GUARDA:
├─ JWT en almacenamiento local
└─ Datos del usuario para mostrar en UI
FUTURAS PETICIONES:
Envía JWT como "credencial" en cada solicitud
BASE DE DATOS (PostgreSQL)

```
│   ┌──────────────────────────────────────────┐                    │
│   │           Tabla: users                   │                    │
│   ├──────────────────────────────────────────┤                    │
│   │ id (PK)              ← Auto-generado     │                    │
│   │ google_id (UNIQUE)   ← De Google         │                    │
│   │ email (UNIQUE)       ← De Google         │                    │
│   │ name                 ← De Google         │                    │
│   │ picture_url          ← De Google         │                    │
│   │ role                 ← Definido por ti   │                    │
│   │ created_at           ← Auto              │                    │
│   │ last_login           ← Actualizado       │                    │
│   └──────────────────────────────────────────┘                    │
└──────────────────────────────────────────────────────────────────────┘
```

🔄 Flujo Conceptual Completo (Sin Código)

📍 FASE 1: Autenticación con Google

```
CLIENTE
   │
   │ (1) Usuario: "Quiero registrarme/iniciar sesión"
   │
   ↓
GOOGLE
   │
   │ (2) Google: "¿Me autorizas compartir tu info?"
   │
   │ (3) Usuario: "Sí, autorizo"
   │
   │ (4) Google valida identidad
   │
   │ (5) Google devuelve:
   │     - ID único del usuario (google_id)
   │     - Email
   │     - Nombre
   │     - Foto
   │
   ↓
TU BACKEND
```

📍 FASE 2: Procesamiento en Backend

```
TU BACKEND
   │
   │ (6) Recibe datos de Google
   │
   ↓
BASE DE DATOS
   │
   │ (7) Backend pregunta: "¿Existe usuario con este google_id?"
   │
   ├──→ NO EXISTE (Primera vez)
   │    │
   │    │ (8) Crear nuevo registro en tabla users:
   │    │     - google_id = ID de Google
   │    │     - email = Email de Google
   │    │     - name = Nombre de Google
   │    │     - picture_url = Foto de Google
   │    │     - role = "customer" (por defecto)
   │    │     - created_at = Ahora
   │    │     - last_login = Ahora
   │    │     - id = Auto-generado por PostgreSQL
   │    │
   │    └──→ Resultado: Usuario creado (id = 1)
   │
   └──→ SÍ EXISTE (Usuario recurrente)
        │
        │ (8) Actualizar registro existente:
        │     - last_login = Ahora
        │     - name = Actualizar (por si cambió)
        │     - picture_url = Actualizar (por si cambió)
        │
        └──→ Resultado: Usuario actualizado (id = 1)

   ↓
TU BACKEND
```

📍 FASE 3: Generación del Sistema Intermedio (JWT)

```
TU BACKEND
   │
   │ (9) Ahora backend tiene:
   │     - user.id = 1
   │     - user.email = "cliente@gmail.com"
   │     - user.role = "customer"
   │
   │ (10) GENERA JWT (Sistema Intermedio)
```

```
    │
    ├─→ Contenido del JWT:
    │   ├─ ID del usuario: 1
    │   ├─ Email: "cliente@gmail.com"
    │   ├─ Role: "customer"
    │   ├─ Fecha de creación
    │   ├─ Fecha de expiración (ej: 24 horas)
    │   └─ FIRMA (sello de seguridad)
    │
    │ (11) El JWT es como una "credencial portable"
    │      que el cliente puede llevar consigo
    │
    ↓
```

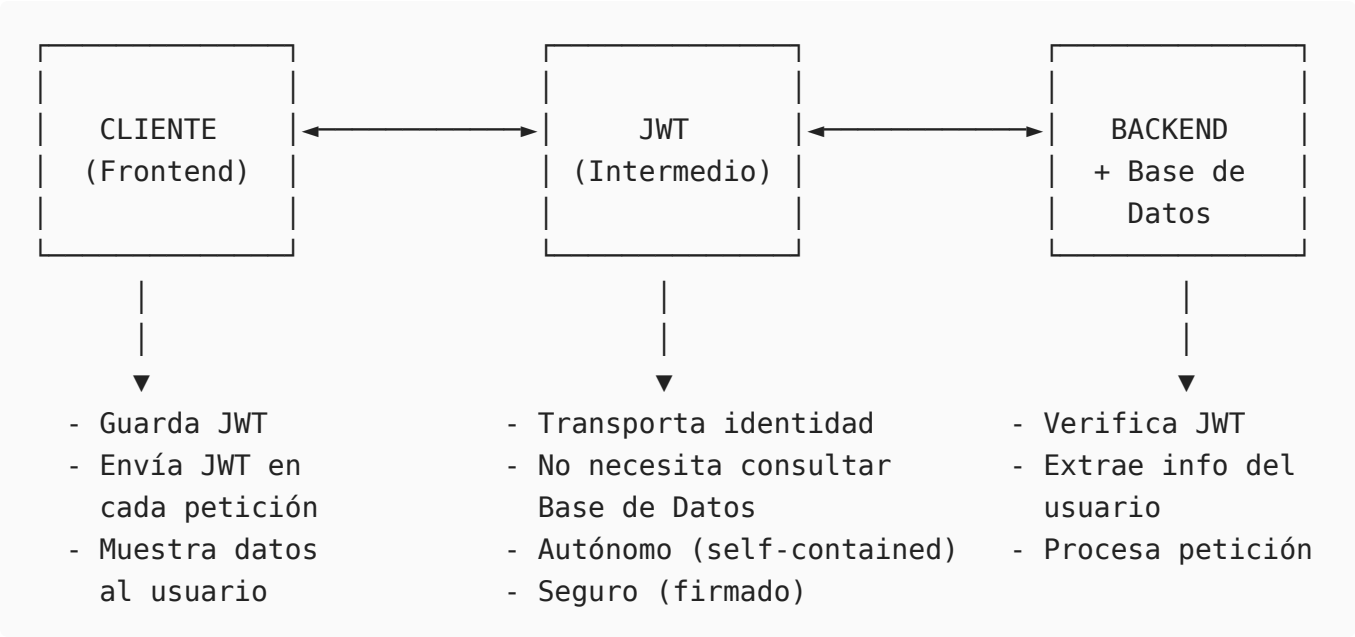
📍 FASE 4: Respuesta al Cliente

```
TU BACKEND
    │
    │ (12) Backend devuelve al cliente:
    │
    ├─→ JWT (token): "eyJhbGc..."
    │   └─ Este es el SISTEMA INTERMEDIO
    │
    └─→ Datos del usuario:
        ├─ id: 1
        ├─ email: "cliente@gmail.com"
        ├─ name: "Juan Pérez"
        ├─ picture_url: "https://..."
        └─ role: "customer"

    ↓
CLIENTE
    │
    │ (13) Cliente recibe y GUARDA:
    │      - JWT → En memoria local del navegador
    │      - Datos del usuario → Para mostrar en UI
    │
    └─→ Ahora el cliente tiene su "credencial"
```

🔑 JWT como Sistema Intermedio - Concepto

¿Por qué JWT es un INTERMEDIARIO?



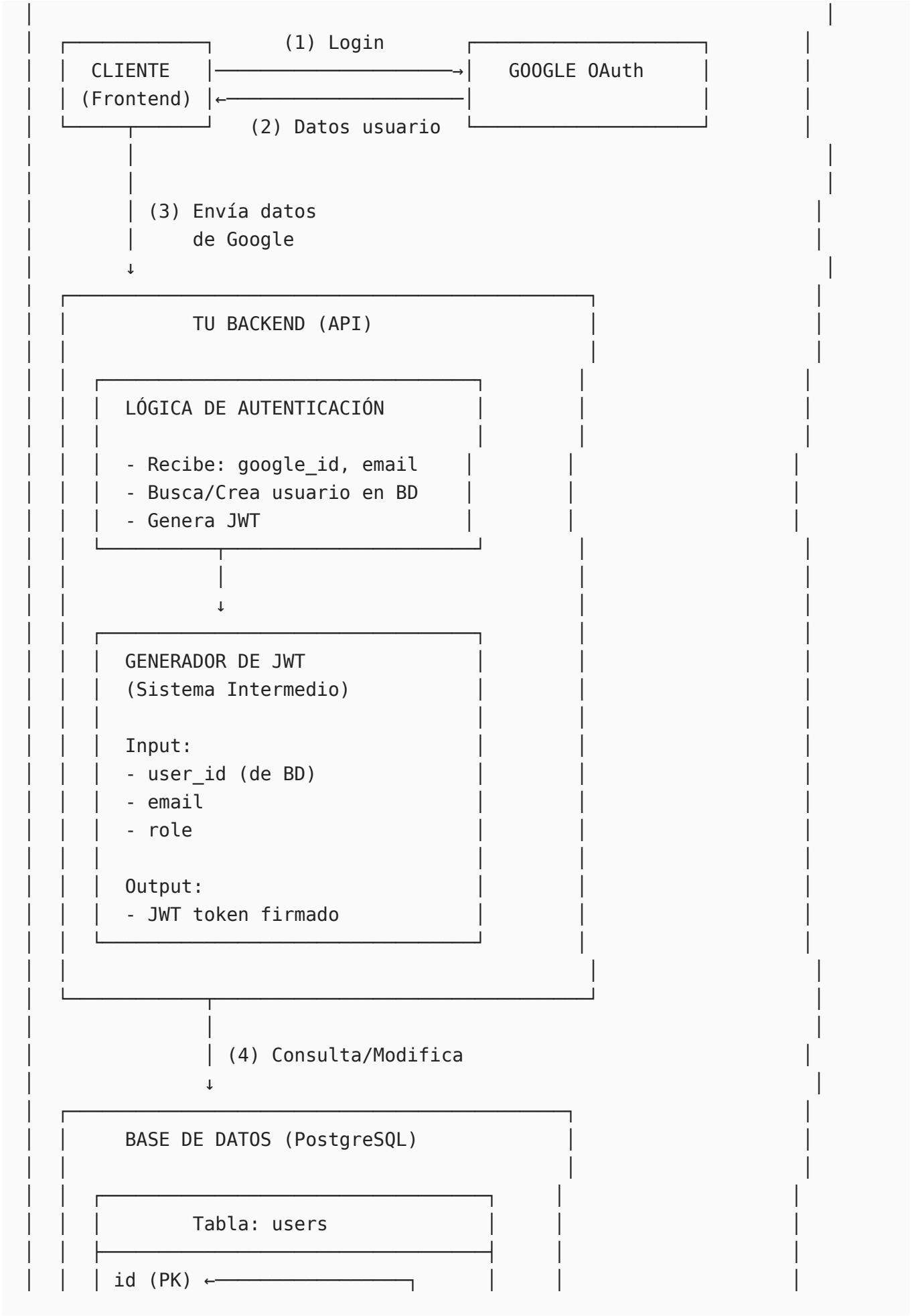
Analogía del Mundo Real:

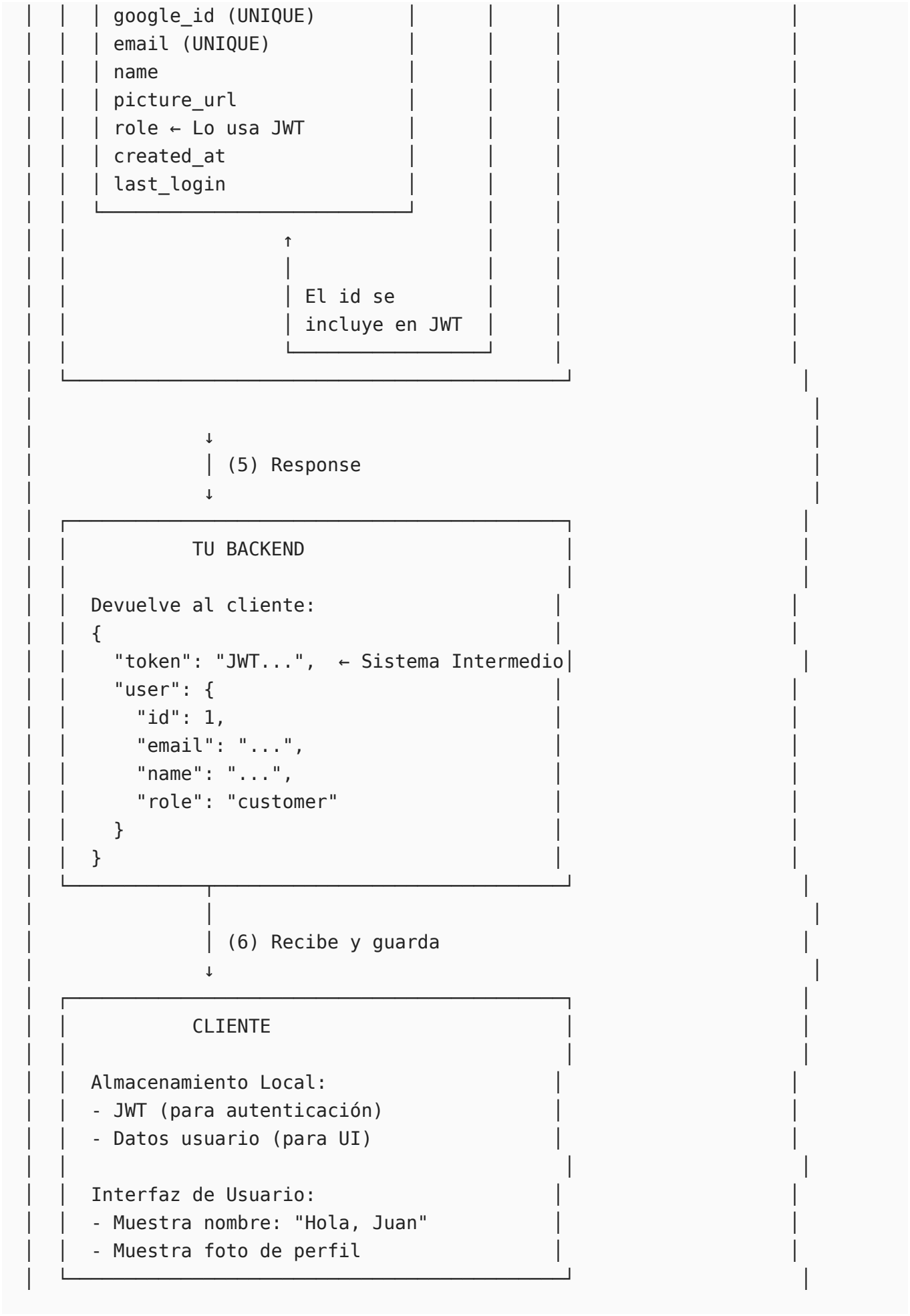
Elemento	Analogía
Backend	Oficina de identificación (DMV)
JWT	Licencia de conducir
Cliente	Tú con la licencia
Google	Agencia que verifica tu identidad original

Flujo:

1. Vas a la oficina (Backend) con tu acta de nacimiento (Google ID)
2. La oficina verifica tu identidad con Google
3. La oficina te da una licencia de conducir (JWT)
4. Guardas tu licencia (Frontend guarda JWT)
5. Cuando necesitas identificarte, muestras la licencia (envías JWT)
6. La gente verifica que la licencia sea válida (Backend verifica JWT)
7. **No necesitas volver a la oficina cada vez** (no consultas BD cada vez)

📊 DER + Arquitectura Combinado


CLIENTE
(Frontend)
(1) Login
(2) Datos usuario
GOOGLE OAuth
(3) Envía datos
de Google
TU BACKEND (API)
LÓGICA DE AUTENTICACIÓN
- Recibe: google_id, email
- Busca/Crea usuario en BD
- Genera JWT
GENERADOR DE JWT
(Sistema Intermedio)
Input:
- user_id (de BD)
- email
- role
Output:
- JWT token firmado
(4) Consulta/Modifica
BASE DE DATOS (PostgreSQL)
Tabla: users
id (PK)


google_id (UNIQUE)
email (UNIQUE)
name
picture_url
role ← Lo usa JWT
created_at
last_login
El id se incluye en JWT
(5) Response
TU BACKEND
Devuelve al cliente:
{
"token": "JWT...", ← Sistema Intermedio
"user": {
"id": 1,
"email": "...",
"name": "...",
"role": "customer"
}
}
(6) Recibe y guarda
CLIENTE
Almacenamiento Local:
- JWT (para autenticación)
- Datos usuario (para UI)
Interfaz de Usuario:
- Muestra nombre: "Hola, Juan"
- Muestra foto de perfil

```
└──────────────────────────────────────────────────────────────┘
```

📝 Resumen Conceptual

El JWT es un Sistema Intermedio porque:

1. **Conecta** Frontend ↔ Backend sin estado persistente
2. **Transporta** identidad del usuario
3. **Evita** consultas repetidas a la Base de Datos
4. **Permite** que el backend "confíe" en el cliente
5. **Es autónomo** - contiene toda la info necesaria

Componentes del Sistema:

Componente	Rol
Google	Proveedor de identidad externo
Backend	Validador y generador de credenciales
JWT	**Sistema Intermedio/Credencial portable**
Base de Datos	Almacenamiento persistente del usuario
Frontend	Portador de la credencial

Flujo en 6 pasos:

```
1. Cliente → Google (autenticación)
2. Google → Backend (datos del usuario)
3. Backend → BD (guardar/actualizar usuario)
4. Backend → Genera JWT (sistema intermedio)
5. Backend → Cliente (JWT + datos)
6. Cliente guarda JWT para futuras peticiones
```

Fecha: 2025-12-05
Versión: 1.0