

**Universidad Politécnica de Chiapas.**

<b>Cuatrimestre.</b>	Septiembre- Diciembre 2025
<b>Grupo.</b>	Noveno B
<b>Asignatura.</b>	SOA
<b>Corte</b>	C2
<b>Actividad.</b>	Elaborar el modelo lógico de la base de datos (tablas, claves primarias y foráneas, relaciones y cardinalidades).
<b>Fecha</b>	
<b>Matricula.</b>	221189 231208 221214 231198
<b>Nombre.</b>	Luis Alberto Batalla. Willber Hernandez. Esduardo Palomeque Roblero. Maria Fernanda Sanchez

# Modelo Lógico de Base de Datos

---

**Versión:** 1.0

**Fecha:** 24 de octubre de 2025

## 1. Introducción

Este documento detalla el modelo lógico de la base de datos para el sistema "Reconexión Humana". Traduce el modelo conceptual en una estructura de tablas concreta, definido columnas, tipos de datos, claves primarias (PK), claves foráneas (FK) y las tablas pivote necesarias para resolver las relaciones de muchos a muchos (NM).

El enfermo se alinea con el patrón **Base de datos por servicio**, por lo que las tablas están agrupadas según el microservicio que las gestiona.

---

## 2. Contexto: AuthIdentity (Base de Datos Relacionales - PostgreSQL)

Este contexto maneja la identidad y el perfil del usuario. Requiere consistencia fuerte (ACID).

- **users**
  - `user_id` (PK, UUID): Identificador único del usuario.
  - `username` (VARCHAR, UNIQUE): Nombre de usuario público.
  - `email` (VARCHAR, UNIQUE): Email para login y notificaciones.
  - `password_hash` (VARCHAR): Hash de la contraseña.
  - `created_at` (TIMESTAMP): Fecha de registro.
- **user\_profiles**
  - `user_id` (PK, FK -> users.user\_id): Clave primaria y foránea.
  - `full_name` (VARCHAR): Nombre completo del usuario.
  - `bio` (TEXT): Biografía del perfil.
  - `profile_picture_url` (VARCHAR): URL de la imagen de perfil.

**Relación:**

- `users` 1–1 `user_profiles` (Un usuario tiene un perfil).
- 

## 3. Contexto: SocialConnect (Persistencia Políglota)

Este es el contexto más complejo. El modelo lógico se presenta aquí en un formato relacional para mayor claridad, pero en la implementación, estas tablas se distribuirán en diferentes tecnologías de base de datos (documental, grafo, etc.) para optimizar el rendimiento.

### 3.1. Tablas de Contenido (Implementación sugerida: Base de Datos Documental)

- `publications`
  - `publication_id` (PK, UUID): Identificador único de la publicación.
  - `user_id` (FK -> `users.user_id`): El autor de la publicación.
  - `type` (ENUM: 'POST', 'STORY'): Tipo de publicación.
  - `caption` (TEXT): Texto o descripción.
  - `location` (VARCHAR, Nullable): Ubicación asociada.
  - `expires_at` (TIMESTAMP, Nullable): Fecha de expiración para `STORY`.
  - `created_at` (TIMESTAMP): Fecha de creación.
- `media`
  - `media_id` (PK, UUID): Identificador único del medio.
  - `publication_id` (FK -> `publications.publication_id`): Publicación a la que pertenece.
  - `media_url` (VARCHAR): URL del archivo multimedia.
  - `media_type` (ENUM: 'IMAGE', 'VIDEO'): Tipo de archivo.
  - `order_index` (INTEGER): Orden del medio en una publicación con varios archivos.

#### Relación:

- `publications` 1–N `media` (Una publicación puede tener varios archivos multimedia).

### 3.2. Tablas de Interacción

- `comments`
  - `comment_id` (PK, UUID): Identificador único del comentario.
  - `publication_id` (FK -> `publications.publication_id`): Publicación comentada.

- `user_id` (FK -> users.user\_id): Autor del comentario.
  - `text` (TEXT): Contenido del comentario.
  - `created_at` (TIMESTAMP): Fecha de creación.
- **`publication_likes` (Tabla Pivote)**
    - `user_id` (PK, FK -> users.user\_id): Usuario que da "like".
    - `publication_id` (PK, FK -> publications.publication\_id): Publicación que recibe el "like".
    - `created_at` (TIMESTAMP): Fecha de la interacción.

#### Relaciones:

- `publications` 1-N `comments`
- `users` 1-N `comments`
- `users` N-M `publications` (resuelta por `publication_likes` ).

### 3.3. Tablas de Grafo Social (Implementación sugerida: Base de Datos de Grafo)

- **`user_follows` (Tabla Pivote)**
  - `follower_id` (PK, FK -> users.user\_id): El usuario que sigue.
  - `following_id` (PK, FK -> users.user\_id): El usuario que es seguido.
  - `created_at` (TIMESTAMP): Fecha en que se inició el seguimiento.
- **`user_blocks` (Tabla Pivote)**
  - `blocker_id` (PK, FK -> users.user\_id): El usuario que bloquea.
  - `blocked_id` (PK, FK -> users.user\_id): El usuario que es bloqueado.
  - `created_at` (TIMESTAMP): Fecha del bloqueo.

#### Relaciones:

- `users` N-M `users` (Seguimiento, resuelta por `user_follows` ).
- `users` N-M `users` (Bloqueo, resuelta por `user_blocks` ).

### 3.4. Tablas de Mensajería (Implementación sugerida: Base de Datos de Columna Ancha)

- `conversations`

- `conversation_id` (PK, UUID): Identificador único de la conversación.
- `type` (ENUM: 'PRIVATE', 'GROUP'): Tipo de conversación.
- `conversation_name` (VARCHAR, Nullable): Nombre para conversaciones de grupo.
- **`conversation_participants` (Tabla Pivote)**
  - `conversation_id` (PK, FK -> `conversations.conversation_id`): La conversación.
  - `user_id` (PK, FK -> `users.user_id`): El participante.
  - `role` (ENUM: 'ADMIN', 'MEMBER'): Rol en el grupo.
- **`messages`**
  - `message_id` (PK, UUID): Identificador único del mensaje.
  - `conversation_id` (FK -> `conversations.conversation_id`): Conversación a la que pertenece.
  - `sender_id` (FK -> `users.user_id`): Usuario que envió el mensaje.
  - `text` (TEXT): Contenido del mensaje.
  - `sent_at` (TIMESTAMP): Fecha de envío.

## Relaciones:

- `conversations` 1—N `messages`
- `users` 1—N `messages`
- `users` N—M `conversations` (resuelta por `conversation_participants` ).

