

Universidad Politécnica de Chiapas.

| | |
|----------------------|--|
| Cuatrimestre. | Septiembre- Diciembre 2025 |
| Grupo. | Noveno B |
| Asignatura. | SOA |
| Corte | C2 |
| Actividad. | Diseñar el modelo conceptual (entidades y relaciones a alto nivel) |
| Fecha | |
| Matricula. | 221189 231208 221214 231198 |
| Nombre. | Luis Alberto Batalla. Willber Hernandez. Esduardo Palomeque Roblero. Maria Fernanda Sanchez |

Requisitos Funcionales y No Funcionales del Sistema

Versión: 1.0

Fecha: 24 de octubre de 2025

Este documento detalla los requisitos que guían el desarrollo y la implementación del sistema "Reconexión Humana", sirviendo como base para el modelado de la base de datos y la arquitectura de microservicios.

1. Requisitos Funcionales (RF)

Los requisitos funcionales describen **qué debe hacer** el sistema.

- **RF-AUTH-01: Registro de Usuario:** Los usuarios deben poder crear una cuenta con un nombre de usuario, email y contraseña.
- **RF-AUTH-02: Autenticación:** Los usuarios deben poder iniciar sesión con su correo electrónico y contraseña para obtener un token de acceso.
- **RF-AUTH-03: Gestión de Perfil:** Los usuarios deben poder ver y editar su perfil público (nombre completo, biografía, foto de perfil).
- **RF-CONTENT-01: Creación de Publicaciones:** Los usuarios deben poder crear publicaciones permanentes (Posts) con texto y contenido multimedia.
- **RF-CONTENT-02: Creación de Contenido Efímero:** Los usuarios deben poder crear publicaciones temporales (Historias) que desaparecen después de 24 horas.
- **RF-SOCIAL-01: Seguimiento de Usuarios:** Los usuarios deben poder seguir y dejar de seguir a otros usuarios.
- **RF-SOCIAL-02: Bloque de Usuarios:** Los usuarios deben poder bloquear a otros usuarios para impedir cualquier tipo de interacción.
- **RF-INTERACTION-01: Dar "Me gusta":** Los usuarios deben poder indicar que les gusta una publicación.

- **RF-INTERACTION-02: Comentar:** Los usuarios deben poder dejar comentarios de texto en las publicaciones.
 - **RF-MESSAGING-01: Mensajería Privada:** Los usuarios deben poder iniciar conversaciones uno a uno con otros usuarios.
 - **RF-MESSAGING-02: Mensajería Grupal:** Los usuarios deben poder crear conversaciones con múltiples participantes.
 - **RF-FEED-01: Feed Personalizado:** El sistema debe presentar a cada usuario un feed de contenido relevante de las personas a las que sigue.
 - **RF-RISK-01: Análisis de Comportamiento:** El sistema debe analizar de forma asíncrona y anónima los patrones de interacción para identificar perfiles de riesgo.
 - **RF-RISK-02: Sugerencia de Recursos:** El sistema debe poder sugerir recursos de salud (artículos, guías) a los usuarios identificados con perfiles de riesgo, sin revelar el motivo de la sugerencia.
-

2. Requisitos Sin Funcionales (RNF)

Los requisitos no funcionales descritos **cuerpo debe ser** el sistema en términos de calidad y restricciones.

- **RNF-SECURITY-01: Aislamiento de Datos:** Debe existir una separación estricta y total entre los datos del contexto social (`SocialConnect`) y los datos del contexto de análisis de riesgo (`RiskMitigation`). La única conexión permitida es el `user_id`.
- **RNF-SECURITY-02: Contraseñas Seguras:** El sistema debe verificar que las contraseñas de los nuevos usuarios no están en bases de datos de contraseñas comprometidas conocidas. Las contraseñas deben alcanzar siempre hasheadas con un algoritmo robusto (ej. Argón2, bcrypt).
- **RNF-PERFORMANCE-01: Baja Latencia en Conferencias Críticas:** La carga del feed de un usuario y la entrega de mensajes en tiempo real deben tener una latencia mínima para garantizar una buena experiencia de usuario.
- **RNF-PERFORMANCE-02: Escrituras Asíncronas:** Las operaciones de escritura que no requieren una respuesta inmediata (ej. registrador una vista, alimentar el motor de análisis) deben procesar de forma asíncrona para no bloquear al usuario.
- **RNF-SCALABILITY-01: Escalabilidad horizontal:** Cada microservicio debe poder escalarse de forma independiente para manejar la carga. El sistema debe estar preparado para un alto volumen de conferencias (feeds) y escrituras (interacciones, mensajes).

- **RNF-DISPONIBILIDAD-01: Alta Disponibilidad:** Los servicios críticos de cara al usuario (`AuthIdentity` , `SocialConnect`) deben tener una alta disponibilidad.
- **RNF-DISPONIBILIDAD-02: Resistencia y Tolerancia a Fallos:** La falla de un servicio no crítico (ej. `RiskMitigation` , `NotificationService`) no debe afectar la funcionalidad principal de la red social. El uso de un broker de mensajes ayuda a hacer este objetivo.
- **RNF-MANTENIBILIDAD-01: Bajo Acoplamiento:** Los microservicios deben estar debilmente acoplados para permitir que los equipos trabajen en ellos y los desplieguen de forma independiente.
- **RNF-CONSISTENCIA-01: Consistencia Fuerte para Identidad:** Las operaciones en el servicio `AuthIdentity` (registro, cambio de contraste) deben ser transaccionales y fuertes consistentes (ACID).
- **RNF-CONSISTENCIA-02: Consistencia Eventual para Datos Sociales:** Se acepta la consistencia eventual para datos no críticos. Por ejemplo, si un usuario da "like" a un post, el contador puede tardar unos segundos en actualizar en todas las vistas.

Entidades Principales, Relaciones y Reglas de Negocio

Versión: 1.0

Fecha: 24 de Octubre de 2025

Este documento traduce los requerimientos del sistema en un conjunto de entidades de datos, las relaciones que las conectan y las reglas de negocio que gobiernan su comportamiento.

1. Entidades Principales

Estas son las entidades fundamentales que componen el dominio del sistema "Reconexión Humana".

- **User:** La entidad central que representa a un individuo en la plataforma. Contiene los datos de autenticación y la información esencial.
- **UserProfile:** Datos públicos y editables del usuario, como su biografía y foto de perfil. Está directamente asociada a un `User`.
- **Publication:** Una **generalización** para agrupar todo el contenido generado por el usuario (`Post`, `Story`). Esto simplifica las interacciones como `Likes` y `Comments`, ya que pueden apuntar a una única entidad `Publication` en lugar de a múltiples tipos de contenido.
- **Media:** Representa un archivo multimedia individual (imagen, video) que puede estar asociado a una `Publication` o a un `Message`.
- **Comment:** Un comentario de texto realizado por un `User` en una `Publication` específica.
- **Conversation:** Representa una sala de chat, que puede ser una conversación privada (dos participantes) o un grupo (múltiples participantes).
- **Message:** Un mensaje individual (texto o multimedia) enviado por un `User` dentro de una `Conversation`.
- **Resource:** Contenido educativo y de apoyo gestionado por el microservicio `ResourcesDocs`.
- **RiskProfile:** Un perfil de datos interno y aislado, asociado a un `User`, que contiene la puntuación de riesgo calculada por el sistema.

2. Relaciones Clave

Las relaciones definen cómo interactúan las entidades entre sí.

- Un **User** **tiene un** **UserProfile** (Relación 1 a 1).
 - Un **User** **crea muchas** **Publications** (Relación 1 a N).
 - Un **User** **escribe muchos** **Comments** (Relación 1 a N).
 - Un **User** **envía muchos** **Messages** (Relación 1 a N).
 - Una **Publication** **puede tener muchos** **Comments** (Relación 1 a N).
 - Una **Publication** **puede contener muchos** **Media** (Relación 1 a N).
 - Una **Conversation** **contiene muchos** **Messages** (Relación 1 a N).
 - La relación de **seguimiento** entre **Users** es de muchos a muchos (N a M).
 - La relación de **bloqueo** entre **Users** es de muchos a muchos (N a M).
 - La relación de **"like"** entre **User** y **Publication** es de muchos a muchos (N a M).
 - La relación de **participación** entre **User** y **Conversation** es de muchos a muchos (N a M).
-

3. Reglas de Negocio Críticas

Estas reglas imponen restricciones y lógica sobre el modelo de datos.

1. **Unicidad de Identidad:** Los campos **username** y **email** en la entidad **User** deben ser únicos en todo el sistema (Regla derivada de RF-AUTH-01).
2. **Propiedad de Contenido:** Solo el **User** que creó una **Publication** o un **Comment** puede eliminarlo.
3. **Interacción Única:** Un **User** solo puede dar "like" una vez a una misma **Publication**. Un segundo intento debe deshacer el "like".
4. **Lógica de Bloqueo:** Si **UserA** bloquea a **UserB**, el sistema debe impedir que **UserB** vea el **UserProfile** o las **Publications** de **UserA**, y que pueda iniciar una **Conversation** con **UserA** (Regla derivada de RF-SOCIAL-02).
5. **Temporalidad de Contenido:** Las **Publications** de tipo "Story" deben tener una fecha de expiración. El sistema no debe mostrarlas en los feeds después de esa fecha (Regla derivada de RF-CONTENT-02).
6. **Aislamiento de Riesgo:** El **user_id** es la única clave foránea permitida para conectar el mundo social con la entidad **RiskProfile**. No debe existir ninguna otra relación directa o indirecta entre las entidades del **SocialConnect** y las del **RiskMitigation** para cumplir con RNF-SECURITY-01.

Descripción del Diagrama del Modelo Conceptual

Versión: 1.0

Fecha: 24 de Octubre de 2025

Este documento proporciona una explicación detallada del diagrama Entidad-Relación (ER) que representa el modelo conceptual del sistema "Reconexión Humana".

1. Propósito del Diagrama

El diagrama del modelo conceptual es una representación visual de alto nivel de las principales entidades de datos del sistema y las relaciones que existen entre ellas. Su objetivo es:

- Comunicar la estructura:** Ofrecer una visión clara y unificada de la arquitectura de datos a todos los miembros del equipo (desarrolladores, analistas, stakeholders).
- Validar el entendimiento:** Asegurar que el modelo de datos refleja correctamente los requerimientos funcionales y las reglas de negocio.
- Servir como base:** Actuar como el punto de partida para el diseño de los modelos lógicos y físicos de las diferentes bases de datos que componen el sistema.

2. Descripción de Componentes

El diagrama está organizado en torno a los contextos de dominio clave del sistema.

2.1. Contexto de Identidad y Perfil

- User**: Es la entidad raíz. Contiene los datos críticos para la autenticación (`email` , `password_hash`) y la identificación única (`user_id` , `username`). Las restricciones `UNIQUE` en `username` y `email` son fundamentales para la integridad del sistema.
- UserProfile**: Contiene datos públicos y descriptivos del usuario. La relación uno a uno (`||--||`) con `User` indica que cada usuario tiene exactamente un perfil.

2.2. Contexto de Contenido y Social

- **Publication** : Es una entidad generalizada que representa cualquier contenido creado por un usuario (un post, una story, etc.). El campo `type` distingue entre ellos. Esta decisión de diseño simplifica enormemente las interacciones, ya que **Comment** y las relaciones de **likes** solo necesitan apuntar a **Publication**.
- **Media** : Representa un archivo multimedia. Su relación uno a muchos (`||--o{`) con **Publication** permite que una publicación contenga múltiples imágenes o videos.
- **Comment** : Un comentario hecho por un **User** en una **Publication**.

2.3. Contexto de Mensajería

- **Conversation** : Representa una sala de chat. El campo `type` diferencia entre conversaciones privadas y grupales.
- **Message** : Un mensaje individual enviado por un **User** dentro de una **Conversation**.

2.4. Contexto de Salud (Aislado)

- **RiskProfile** y **Resource** : Estas entidades están visualmente separadas para reforzar la regla de negocio de **aislamiento de datos (RNF-SECURITY-01)**. La única conexión con el resto del sistema es a través del `user_id` en **RiskProfile**, que debe ser manejado con extremo cuidado.

3. Interpretación de las Relaciones

El diagrama utiliza la notación de "patas de gallo" para representar la cardinalidad:

- **Uno a Uno (`||--||`)**: Un **User** tiene un **UserProfile**.
- **Uno a Muchos (`||--o{`)**: Un **User** puede crear muchas **Publications**. Una **Publication** puede tener muchos **Comments**.
- **Muchos a Muchos (`}o--o{`)**:
 - **User }o--o{ User** : Representa relaciones como "follows" y "blocks". Un usuario puede seguir a muchos otros, y ser seguido por muchos.
 - **User }o--o{ Publication** : Representa los "likes". Un usuario puede dar like a muchas publicaciones, y una publicación puede recibir likes de muchos usuarios.
 - **User }o--o{ Conversation** : Representa la participación en chats. Un usuario puede estar en muchas conversaciones, y una conversación (de grupo) puede tener muchos usuarios.

En un modelo lógico, las relaciones de muchos a muchos se implementarán como tablas intermedias (ej. `user_follows` , `publication_likes` , `conversation_participants`).

4. Decisiones de Diseño Reflejadas

- **Generalización:** La entidad `Publication` es el ejemplo más claro de generalización para reducir la complejidad y evitar la duplicación de tablas de interacciones.
- **Anotaciones:** El uso de etiquetas como `UNIQUE` , `Nullable` y `Enum` directamente en el diagrama conceptual ayuda a capturar reglas de negocio importantes desde una etapa temprana.
- **Aislamiento de Contextos:** La separación visual de los contextos (Identidad, Social, Mensajería, Salud) se alinea con la arquitectura de microservicios definida, donde cada contexto será gestionado por un servicio diferente.

