

# INF-239: Bases de Datos

## Tarea 2: Proceso de Normalización a 3FN

Javier Rojas - 202304532-2

Nicolás Pérez - 202304539-k

### 1. Modelo Inicial (Forma No Normalizada - FNN)

El primer modelo de la base de datos (basado en Modelo Logico y Diccionario de Datos.pdf y BDDno3FN.txt) presentaba dos problemas principales:

1. **Atributos no atómicos:** La tabla `Solicitudes_Funcionalidad` incluía un campo `Criterios_Aceptacion` que, según la función `Nueva_Funcionalidad`, estaba diseñado para almacenar un arreglo de valores.
2. **Redundancia de datos:** Existía una duplicación significativa de estructuras y datos:
  - Las tablas `Usuarios` e `Ingenieros` duplicaban columnas como `Nombre` y `Email`.
  - Las tablas `Solicitudes_Funcionalidad` y `Solicitudes_Errores` compartían la mayoría de sus atributos (`Titulo`, `id_topico`, `Rut_Solicitante`, `Estado`, `Fecha`).
  - Las tablas de asignación (`Asignacion_Funcionalidades` y `Asignacion_Errores`) eran idénticas.

### 2. Proceso de Normalización Paso a Paso

#### 2.1. Paso 1: Hacia la Primera Forma Normal (1NF)

La 1NF exige que todos los atributos sean **atómicos**, es decir, que no contengan conjuntos o arreglos de valores.

- **Problema:** El atributo `Criterios_Aceptacion` en `Solicitudes_Funcionalidad` incumplía esta regla.
- **Solución:** Se tomó este atributo y se llevo a una nueva tabla llamada `Funcionalidad_Aceptada`, en esta nueva tabla cada criterio es un registro individual que se vincula a la solicitud de funcionalidad mediante una llave foránea (`id_SF`).

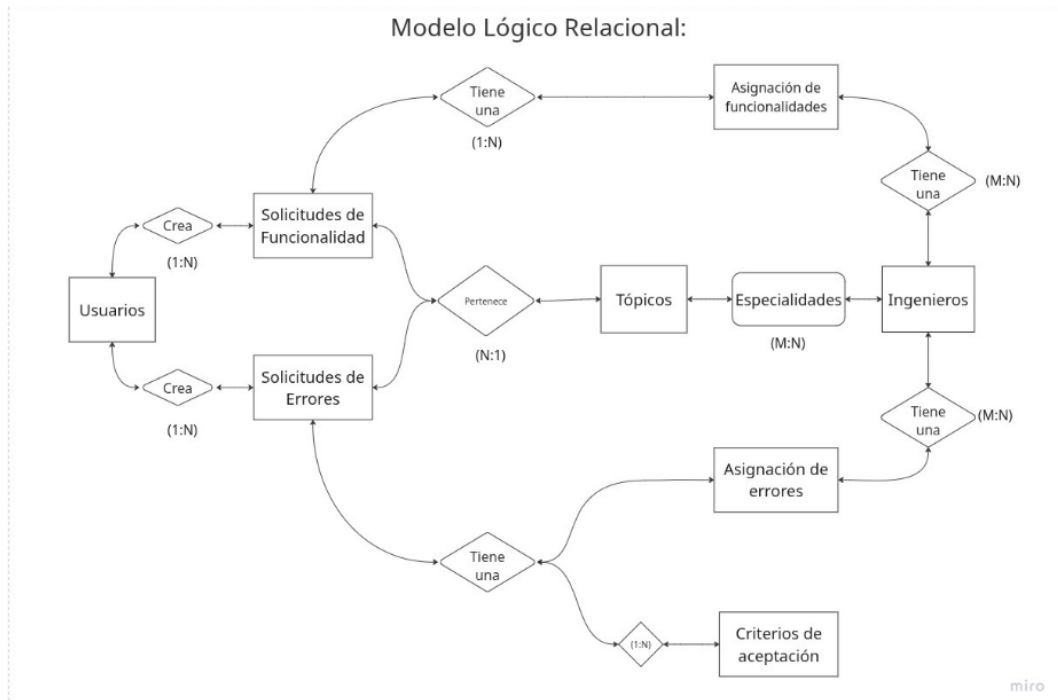


Figura 1: Diagrama Entidad-Relación del modelo inicial (FNN).

## 2.2. Paso 2: Hacia la Segunda Forma Normal (2NF)

La 2NF se aplica a tablas con claves primarias compuestas y exige que no existan dependencias parciales (es decir, que todos los atributos no clave dependan de la clave primaria completa).

- **Análisis:** Las tablas con claves compuestas en el modelo inicial eran **Especialidades**, **Asignacion\_Funcionalidades** y **Asignacion\_Errores**.
- **Conclusión:** Estas tablas no poseían atributos no-clave (solo contenían las llaves que formaban la PK). Por lo tanto, el modelo cumplía trivialmente con la 2NF.

## 2.3. Paso 3: Hacia la Tercera Forma Normal (3FN)

La 3FN exige que no existan **dependencias transitivas**, lo cual en la práctica se traduce en eliminar la redundancia de datos entre tablas no relacionadas.

- **Problema:** La alta **redundancia** de datos (Usuarios/Ingenieros y Solicitudes F/E) violaba la 3FN.
- **Solución (Generalización):** Se aplicó un enfoque de **generalización y especialización** para consolidar la información común en "tablas padre" y dejar solo los datos únicos en "tablas hija".

### 1. Generalización de Usuarios:

- Se creó una tabla padre **Usuarios** que contiene los datos comunes (RUT, nombre, email, contraseña).

- La tabla **Ingenieros** se convirtió en una tabla hija que solo almacena el `rut_ingeniero`, el cual es a la vez Llave Primaria y Llave Foránea que referencia a **Usuarios**.

## 2. Generalización de Solicitudes:

- Se creó una tabla padre **Solicitudes** con los campos comunes (ID, tipo, titulo, `id_topico`, `rut_usuario`, estado, fecha).
- Se crearon dos tablas hijas: **Solicitudes\_Funcionalidades** (solo con ambiente y resumen) y **Solicitudes\_Errores** (solo con descripcion). Sus IDs son Llaves Primarias y Foráneas que referencian a **Solicitudes**.

## 3. Unificación de Asignaciones:

- Las dos tablas **Asignacion\_Funcionalidades** y **Asignacion\_Errores** se unificaron en una sola tabla **Asignaciones**. Esta tabla ahora vincula el `id_asignacion` (que apunta al ID de la tabla padre **Solicitudes**) con el `rut_ingeniero`.

# 3. Modelo Final (Esquema 3FN)

El resultado de este proceso es el esquema implementado en `TablasyTriggers.txt`. Esta nueva estructura está completamente en 3FN:

- **Cumple 1NF:** Todos los atributos son atómicos.
- **Cumple 2NF:** No existen dependencias parciales.
- **Cumple 3NF:** Se eliminó toda la redundancia y dependencias transitivas mediante la generalización.

Este modelo es más robusto, eficiente y previene anomalías de inserción, actualización y borrado.

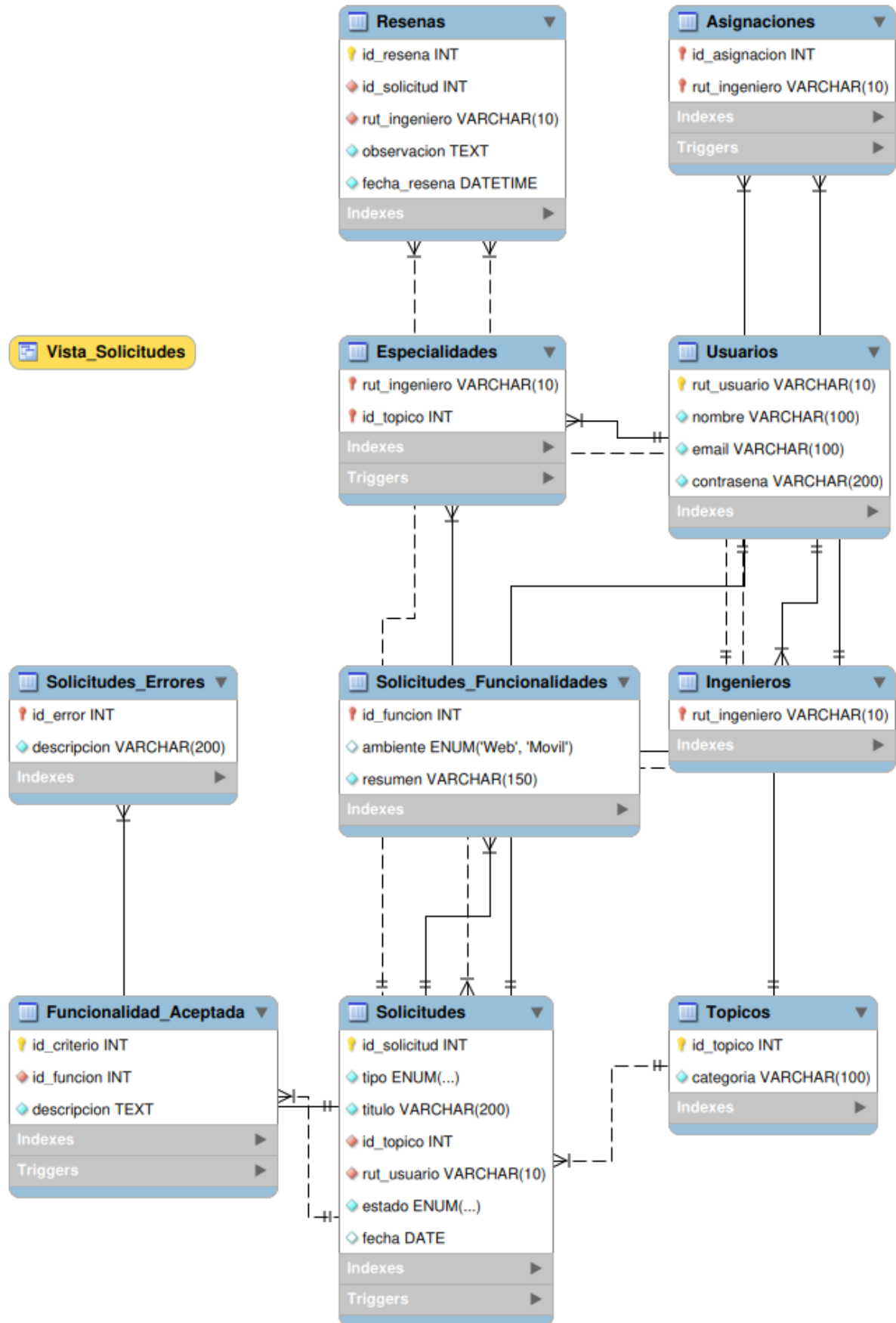


Figura 2: Diagrama Entidad-Relación del modelo final normalizado (3FN).