

# Normalización y Transformación del Modelo Entidad-Relación

## Análisis de Bases de Datos

Diciembre 2025 - Versión 2.0 Profesional

Proceso de Normalización hasta 3FN y Transformación a Modelo Relacional

12 Páginas - Incluye DDL SQL y Análisis Completo

---

## Tabla de Contenidos

1. [Introducción](#)
  - [Contexto y Alcance](#)
  - [Metodología Aplicada](#)
  - [Justificación de la Normalización](#)
2. [Identificación de Entidades y Atributos](#)
  - [Entidad USUARIOS \(Arquitectos\)](#)
  - [Entidad CLIENTE](#)
  - [Entidad PLANOS](#)
  - [Entidad SUSCRIPCIONES](#)
3. [Normalización](#)
  - [Primera Forma Normal \(1FN\)](#)
  - [Segunda Forma Normal \(2FN\)](#)
  - [Tercera Forma Normal \(3FN\)](#)
4. [Transformación de Relaciones](#)
  - [Relación USUARIOS -- CALCULAN -- PLANOS](#)
  - [Relación USUARIOS -- HABLAN -- CLIENTE](#)
  - [Relación CLIENTE -- PAGA -- SUSCRIPCIONES](#)
5. [Análisis Detallado de Dependencias](#)
  - [Dependencias Funcionales Identificadas](#)
  - [Dependencias Transitivas Eliminadas](#)
6. [Modelo Relacional Final](#)
  - [Definición de Tablas](#)
  - [Especificación Detallada de Claves Foráneas](#)
7. [Sentencias SQL DDL](#)
  - [Creación de Tabla USUARIOS](#)
  - [Creación de Tabla CLIENTE](#)
  - [Creación de Tabla PLANOS](#)
  - [Creación de Tabla SUSCRIPCIONES](#)
8. [Restricciones y Reglas de Negocio](#)
  - [Restricciones de Integridad](#)
  - [Reglas de Negocio](#)
9. [Análisis de Anomalías](#)
  - [Anomalía de Inserción](#)
  - [Anomalía de Actualización](#)
  - [Anomalía de Eliminación](#)

## 10. Conclusión

- [Características del Modelo Final](#)
- [Recomendaciones para Implementación](#)

## 11. Referencias

---

# Introducción

El objetivo de este documento es presentar el proceso completo de normalización y transformación del modelo Entidad-Relación (E-R) a un modelo relacional totalmente normalizado hasta la Tercera Forma Normal (3FN). Se parte del diagrama E-R proporcionado, que representa un sistema de gestión integral para arquitectos, clientes, planos y suscripciones de servicios profesionales[1].

Este sistema permite a los usuarios (arquitectos) diseñar y compartir planos con clientes, quienes pueden contratar diferentes tipos de suscripción para acceder a servicios y documentación técnica. La estructura organiza toda la información de forma coherente y sin redundancias innecesarias, garantizando consistencia de datos y facilitando consultas complejas.

## Contexto y Alcance

El modelo propuesto está diseñado para:

- Gestionar información completa de arquitectos y su información de contacto
- Registrar clientes empresariales y sus datos de pago
- Almacenar planos diseñados con metadatos (tamaño, formato, fechas)
- Mantener tipos de suscripción disponibles y su asignación a clientes
- Garantizar integridad referencial entre todas las entidades
- Facilitar consultas complejas sin anomalías de datos
- Prevenir corrupción de datos mediante restricciones de integridad

El proceso de análisis incluye identificación exhaustiva de entidades y atributos, normalización paso a paso hasta Tercera Forma Normal (3FN), análisis detallado de cardinalidades, transformación sistemática de relaciones a tablas, obtención del modelo relacional final coherente y optimizado, generación de sentencias SQL DDL para implementación, y análisis de anomalías de datos evitadas mediante normalización.

## Metodología Aplicada

Se ha utilizado la metodología estándar de normalización relacional, basada en los trabajos seminales de Codd (1970) y ampliada por los marcos de Date (2013) y Elmasri/Navathe (2016)[2][3]. El análisis incluye:

1. **Análisis de dependencias funcionales** para identificar problemas de normalización
2. **Revisión de cardinalidades** del diagrama E-R original
3. **Transformación sistemática** de relaciones E-R a tablas relacionales
4. **Verificación de restricciones** de integridad referencial
5. **Documentación de decisiones** de diseño
6. **Generación de DDL SQL** para implementación en SGBD
7. **Análisis de anomalías** eliminadas mediante normalización

## Justificación de la Normalización

La normalización de bases de datos es fundamental para garantizar calidad, consistencia y mantenibilidad. Sin normalización adecuada, surgen anomalías como:

- **Anomalías de inserción:** Imposibilidad de insertar datos sin completar información irrelevante
- **Anomalías de actualización:** Inconsistencias si un dato se actualiza en unos lugares pero no en otros
- **Anomalías de eliminación:** Pérdida de información cuando se elimina un registro

El modelo normalizado a 3FN elimina estas anomalías completamente.

---

## Identificación de Entidades y Atributos

En esta sección se describen todas las entidades presentes en el modelo E-R, junto con sus atributos correspondientes, claves primarias y restricciones aplicables.

### Entidad USUARIOS (Arquitectos)

Representa a los usuarios del sistema que actúan como arquitectos responsables del cálculo de planos y comunicación con clientes. Esta entidad es fundamental pues centraliza toda la información profesional de los diseñadores que utilizan la plataforma.

#### Atributos:

- **DNI** (clave primaria): Documento Nacional de Identidad del usuario. Identificador único y no nulo. Formato: 8 dígitos + 1 letra según normativa española.
- **Nombre:** Nombre del arquitecto. Campo de texto de hasta 100 caracteres. No nulo.
- **Apellidos:** Apellidos del arquitecto. Campo de texto de hasta 150 caracteres. No nulo.
- **Teléfono:** Número de contacto telefónico. Formato estándar internacional opcional. Máximo 20 caracteres.
- **Correo Electrónico:** Dirección de correo para comunicación. Debe ser válida y única según RFC 5322.
- **Cuenta Bancaria:** Número de cuenta bancaria del usuario. Utilizado para pagos y transferencias. IBAN válido.

### Entidad CLIENTE

Representa a los clientes empresariales que contratan servicios de diseño y suscripciones. Estos pueden ser empresas de cualquier tamaño que requieren servicios de arquitectura o diseño técnico.

#### Atributos:

- **NIF** (clave primaria): Número de Identificación Fiscal del cliente.
- **Nom\_Empresa:** Nombre comercial de la empresa cliente. Nombre oficial registrado de hasta 200 caracteres.
- **Teléfono:** Número de contacto del cliente. Para comunicaciones empresariales. Máximo 20 caracteres.
- **Correo Electrónico:** Dirección de correo electrónico de contacto principal. Válida según RFC 5322.

- **Cuenta Bancaria:** Número de cuenta para pagos de suscripciones y servicios. IBAN válido.

## Entidad PLANOS

Representa los planos diseñados por los arquitectos para los clientes. Constituye el "producto" principal del sistema y su existencia está vinculada directamente a un usuario responsable.

### Atributos:

- **Id\_Plano** (clave primaria): Identificador único del plano. Generado automáticamente (autoincrement o UUID).
- **Tamaño:** Tamaño del plano expresado en unidades de medida (m<sup>2</sup>, cm, etc.). Numérico con precisión decimal.
- **Formato:** Formato del documento de salida (A4, A3, PDF, DWG, CAD, etc.). Texto enumerado con valores válidos.
- **Fecha\_Subida:** Fecha en que se carga el plano al sistema. Timestamp automático del servidor.
- **Fecha\_Entrega:** Fecha de entrega prevista al cliente. Plazo contractual. Posterior a Fecha\_Subida.

## Entidad SUSCRIPCIONES

Representa los tipos de suscripciones que pueden contratar los clientes. Esta entidad define el catálogo de servicios disponibles y establece la relación comercial entre cliente y servicios.

### Atributos:

- **Id\_Suscripción** (clave primaria): Identificador único de la suscripción. Generado automáticamente.
- **Tipo\_Suscripción:** Tipo de plan comercial (básico, premium, enterprise, custom). Enumerado con valores predefinidos.
- **Precio:** Precio mensual o anual del plan. Numérico con dos decimales, precisión currency.

---

## Normalización

Este apartado detalla el proceso de normalización del modelo desde el estado inicial hasta la Tercera Forma Normal (3FN). La normalización es crucial para eliminar redundancias, anomalías de actualización y garantizar consistencia de datos en todas las operaciones[2].

### Primera Forma Normal (1FN)

**Definición:** Una relación está en Primera Forma Normal si todos los atributos contienen valores atómicos (indivisibles) y no existen atributos multivaluados ni atributos compuestos.

### Análisis del modelo:

Todas las entidades definidas en el modelo E-R cumplen con 1FN porque:

- **Atomicidad:** Todos los atributos son indivisibles y no pueden descomponerse en partes más pequeñas.
- **No hay multivaluados:** Ningún atributo contiene múltiples valores en una sola celda.
- **No hay compuestos:** Ningún atributo puede separarse en subcomponentes más granulares.
- **Cada registro contiene exactamente un valor por atributo:** No hay filas con múltiples valores en un campo.

**Conclusión:** El modelo respeta 1FN sin necesidad de transformaciones. ✓

## Segunda Forma Normal (2FN)

**Definición:** Una relación está en Segunda Forma Normal si está en 1FN y todos los atributos no clave tienen dependencia funcional completa de la clave primaria.

### Análisis del modelo:

La Segunda Forma Normal afecta únicamente a tablas con claves primarias compuestas. En el modelo actual, todas las claves primarias son simples:

- **USUARIOS:** clave primaria simple (DNI)
- **CLIENTE:** clave primaria simple (NIF)
- **PLANOS:** clave primaria simple (Id\_Plano)
- **SUSCRIPCIONES:** clave primaria simple (Id\_Suscripción)

Como todas las claves primarias son simples, no pueden existir dependencias parciales. Todos los atributos no clave dependen completamente de la clave primaria.

**Conclusión:** El modelo respeta 2FN sin necesidad de transformaciones. ✓

## Tercera Forma Normal (3FN)

**Definición:** Una relación está en Tercera Forma Normal si está en 2FN y no existen dependencias transitivas.

### Análisis de dependencias transitivas:

#### Entidad SUSCRIPCIONES (Estado Original):

Originalmente contenía: Id\_Suscripción, Tipo\_Suscripción, Precio, **Cuenta\_Bancaria**.

#### Problema identificado:

- Id\_Suscripción → Tipo\_Suscripción: ✓ Dependencia directa (válida)
- Id\_Suscripción → Precio: ✓ Dependencia directa (válida)
- Id\_Suscripción → Cuenta\_Bancaria: ✗ Dependencia transitiva (inválida)

La cuenta bancaria no depende del tipo de suscripción, sino del cliente que la contrata. Esto genera una cadena de dependencia:

$$\text{Id\_Suscripción} \rightarrow (\text{Cliente v\acute{ı} a FK}) \rightarrow \text{Cuenta\_Bancaria}$$

Esta es una **dependencia transitiva** que viola 3FN.

#### Corrección a 3FN:

Se elimina Cuenta\_Bancaria de SUSCRIPCIONES (donde viola 3FN) y se confirma su presencia correcta en CLIENTE.

#### Tabla SUSCRIPCIONES normalizada a 3FN:

- Id\_Suscripción (PK)
- Tipo\_Suscripción
- Precio
- NIF\_Cliente (FK a CLIENTE)

**Conclusión:** Tras eliminar dependencias transitivas, todas las entidades cumplen 3FN completamente. ✓

---

## Transformación de Relaciones

Este apartado describe cómo se transforman cada una de las relaciones del modelo E-R al modelo relacional, implementando las cardinalidades mediante claves foráneas.

### Relación USUARIOS -- CALCULAN -- PLANOS (Cardinalidad 1:N)

**Descripción:** Un usuario (arquitecto) puede calcular/diseñar muchos planos. Cada plano es diseñado por exactamente un único usuario.

#### Cardinalidad:

- **USUARIOS:** 1 (un usuario puede diseñar múltiples planos)
- **PLANOS:** N (cada plano tiene exactamente un usuario diseñador)

#### Transformación al modelo relacional:

Para implementar esta relación 1:N, se añade la clave primaria de USUARIOS como clave foránea en PLANOS.

#### Estructura resultante:

- DNI\_Usuario (clave foránea que referencia USUARIOS.DNI)
- Restricción: **NOT NULL**
- Integridad referencial: PLANOS.DNI\_Usuario → USUARIOS.DNI

### Relación USUARIOS -- HABLAN -- CLIENTE (Cardinalidad 1:N)

**Descripción:** Un usuario puede hablar/comunicarse con muchos clientes. Un cliente está vinculado a un usuario específico (su "gestor" o "responsable de cuenta").

#### Transformación al modelo relacional:

Se añade la clave primaria de USUARIOS como clave foránea en CLIENTE.

#### Estructura resultante:

- DNI\_Usuario (clave foránea que referencia USUARIOS.DNI)
- Restricción: **Puede ser NULL** (cliente sin usuario asignado es posible)
- Integridad referencial: CLIENTE.DNI\_Usuario → USUARIOS.DNI

## Relación CLIENTE -- PAGA -- SUSCRIPCIONES (Cardinalidad 1:N)

**Descripción:** Un cliente puede tener múltiples suscripciones activas. Cada suscripción está asociada a un único cliente pagador. Esta relación captura la esencia del modelo de negocio.

### Transformación al modelo relacional:

Se añade la clave primaria de CLIENTE como clave foránea en SUSCRIPCIONES.

### Estructura resultante:

- NIF\_Cliente (clave foránea que referencia CLIENTE(NIF))
- Restricción: **NOT NULL**
- Integridad referencial: SUSCRIPCIONES.NIF\_Cliente → CLIENTE.NIF

### Ejemplo práctico:

Cliente	Suscripción	Precio
A (NIF 12345678A)	Básico	9.99€/mes
A (NIF 12345678A)	Premium	29.99€/mes
B (NIF 87654321B)	Enterprise	99.99€/mes

Table 1: Ejemplo de relación CLIENTE -- PAGA -- SUSCRIPCIONES

---

## Análisis Detallado de Dependencias

### Dependencias Funcionales Identificadas

#### En USUARIOS:

DNI → {Nombre, Apellidos, Teléfono, Correo, Cuenta\_Bancaria}

#### En CLIENTE:

NIF → {Nom\_Empresa, Teléfono, Correo, Cuenta\_Bancaria, DNI\_Usuario}

#### En PLANOS:

Id\_Plano → {Tamaño, Formato, Fecha\_Subida, Fecha\_Entrega, DNI\_Usuario}

#### En SUSCRIPCIONES:

Id\_Suscripción → {Tipo\_Suscripción, Precio, NIF\_Cliente}

### Dependencias Transitivas Eliminadas

Originalmente en SUSCRIPCIONES:

- Id\_Suscripción → Tipo\_Suscripción (✓ Directo)
- Id\_Suscripción → Precio (✓ Directo)
- Id\_Suscripción → Cuenta\_Bancaria (✗ TRANSITIVO - ELIMINADO)

La cuenta bancaria depende del cliente, no del tipo de suscripción:

$$\text{Id\_Suscripción} \rightarrow \text{NIF\_Cliente} \rightarrow \text{Cuenta\_Bancaria}$$

### Justificación de la eliminación:

La presencia de Cuenta\_Bancaria en SUSCRIPCIONES causaría:

1. **Anomalía de inserción:** No se podría insertar una suscripción sin conocer la cuenta del cliente
2. **Anomalía de actualización:** Si un cliente cambia su cuenta, habría que actualizar TODOS sus registros de suscripción
3. **Redundancia:** La cuenta se repetiría una vez por cada suscripción del cliente

---

## Modelo Relacional Final

### Definición de Tablas

Tabla	Estructura Completa
USUARIOS	(DNI PK, Nombre, Apellidos, Teléfono, Correo, Cuenta_Bancaria)
CLIENTE	(NIF PK, Nom_Empresa, Teléfono, Correo, Cuenta_Bancaria, DNI_Usuario FK)
PLANOS	(Id_Plano PK, Tamaño, Formato, Fecha_Subida, Fecha_Entrega, DNI_Usuario FK)
SUSCRIPCIONES	(Id_Suscripción PK, Tipo_Suscripción, Precio, NIF_Cliente FK)

Table 2: Estructura completa del modelo relacional normalizado a 3FN

### Especificación Detallada de Claves Foráneas

#### CLIENTE.DNI\_Usuario

- **Tipo:** Clave foránea
- **Referencia:** USUARIOS(DNI)
- **Restricción NULL:** Puede ser NULL o NOT NULL (según política)
- **Relación:** HABLAN 1:N
- **Acción eliminación:** SET NULL o RESTRICT

#### PLANOS.DNI\_Usuario

- **Tipo:** Clave foránea
- **Referencia:** USUARIOS(DNI)
- **Restricción NULL:** NOT NULL
- **Relación:** CALCULAN 1:N
- **Acción eliminación:** RESTRICT



## SUSCRIPCIONES.NIF\_Cliente

- **Tipo:** Clave foránea
  - **Referencia:** CLIENTE(NIF)
  - **Restricción NULL:** NOT NULL
  - **Relación:** PAGA 1:N
  - **Acción eliminación:** RESTRICT
- 

## Sentencias SQL DDL

### Creación de Tabla USUARIOS

```
CREATE TABLE USUARIOS (  
DNI VARCHAR(9) PRIMARY KEY NOT NULL,  
Nombre VARCHAR(100) NOT NULL,  
Apellidos VARCHAR(150) NOT NULL,  
Teléfono VARCHAR(20),  
Correo_Electrónico VARCHAR(100) NOT NULL UNIQUE,  
Cuenta_Bancaria VARCHAR(24) NOT NULL,  
CONSTRAINT chk_correo CHECK (Correo_Electrónico LIKE '%@%.%'),  
CONSTRAINT chk_iban CHECK (LENGTH(Cuenta_Bancaria) = 24)  
);
```

### Creación de Tabla CLIENTE

```
CREATE TABLE CLIENTE (  
NIF VARCHAR(9) PRIMARY KEY NOT NULL,  
Nom_Empresa VARCHAR(200) NOT NULL UNIQUE,  
Teléfono VARCHAR(20),  
Correo_Electrónico VARCHAR(100) NOT NULL,  
Cuenta_Bancaria VARCHAR(24) NOT NULL,  
DNI_Usuario VARCHAR(9),  
FOREIGN KEY (DNI_Usuario) REFERENCES USUARIOS(DNI)  
ON DELETE SET NULL,  
CONSTRAINT chk_correo_cliente CHECK (Correo_Electrónico LIKE '%@%.%'),  
CONSTRAINT chk_iban_cliente CHECK (LENGTH(Cuenta_Bancaria) = 24)  
);
```

### Creación de Tabla PLANOS

```
CREATE TABLE PLANOS (  
Id_Plano INT PRIMARY KEY AUTO_INCREMENT NOT NULL,  
Tamaño DECIMAL(10,2) NOT NULL,  
Formato VARCHAR(50) NOT NULL,  
Fecha_Subida TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
Fecha_Entrega DATE NOT NULL,  
DNI_Usuario VARCHAR(9) NOT NULL,  
FOREIGN KEY (DNI_Usuario) REFERENCES USUARIOS(DNI)  
ON DELETE RESTRICT,  
CONSTRAINT chk_tamaño CHECK (Tamaño > 0),  
CONSTRAINT chk_fechas CHECK (Fecha_Subida <= Fecha_Entrega),
```

```
CONSTRAINT chk_formato CHECK (Formato IN ('A4', 'A3', 'PDF', 'DWG', 'CAD'))
);
```

## Creación de Tabla SUSCRIPCIONES

```
CREATE TABLE SUSCRIPCIONES (
  Id_Suscripción INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  Tipo_Suscripción VARCHAR(50) NOT NULL,
  Precio DECIMAL(10,2) NOT NULL,
  NIF_Cliente VARCHAR(9) NOT NULL,
  FOREIGN KEY (NIF_Cliente) REFERENCES CLIENTE(NIF)
  ON DELETE RESTRICT,
  CONSTRAINT chk_precio CHECK (Precio > 0),
  CONSTRAINT chk_tipo_suscripcion CHECK
  (Tipo_Suscripción IN ('Básico', 'Premium', 'Enterprise', 'Custom'))
);
```

---

## Restricciones y Reglas de Negocio

### Restricciones de Integridad

#### Restricciones de Entidad:

- Toda fila en USUARIOS debe tener DNI no nulo y único
- Toda fila en CLIENTE debe tener NIF no nulo y único
- Toda fila en PLANOS debe tener Id\_Plano no nulo y único
- Toda fila en SUSCRIPCIONES debe tener Id\_Suscripción no nulo y único

#### Restricciones de Dominio:

- DNI: Formato 8 dígitos + 1 letra (regulación española)
- Correo electrónico: Formato válido RFC 5322
- Precio: > 0 (positivo y no nulo)
- Tamaño: > 0 (positivo y no nulo)
- Cuenta Bancaria: IBAN válido (24 caracteres)

#### Restricciones de Referencia:

- Todo PLANOS.DNI\_Usuario debe existir en USUARIOS.DNI
- Todo CLIENTE.DNI\_Usuario debe existir en USUARIOS.DNI o ser NULL
- Todo SUSCRIPCIONES.NIF\_Cliente debe existir en CLIENTE.NIF

### Reglas de Negocio

1. Un usuario puede tener múltiples planos: Relación 1:N (USUARIOS-PLANOS)
2. Un plano tiene exactamente un usuario responsable: FK NOT NULL
3. Un usuario puede gestionar múltiples clientes: Relación 1:N (USUARIOS-CLIENTE)
4. Un cliente puede tener múltiples suscripciones: Relación 1:N (CLIENTE-SUSCRIPCIONES)
5. Cada suscripción pertenece a exactamente un cliente: FK NOT NULL
6. Un usuario no puede eliminarse si tiene planos: ON DELETE RESTRICT
7. Un cliente no puede eliminarse si tiene suscripciones: ON DELETE RESTRICT

---

## Análisis de Anomalías

Esta sección analiza cómo el modelo normalizado a 3FN previene las anomalías de datos que ocurrirían en un modelo denormalizado[4].

### Anomalía de Inserción

#### En modelo denormalizado (SIN 3FN):

Si Cuenta\_Bancaria estuviera en SUSCRIPCIONES, sería imposible insertar una nueva suscripción sin conocer y duplicar la cuenta del cliente:

```
INSERT INTO SUSCRIPCIONES_DENORM  
VALUES (15, 'Básico', 9.99, '12345678A', 'ES1234567890');
```

**Problema:** No se puede insertar sin Cuenta\_Bancaria redundante.

#### En modelo normalizado (CON 3FN):

```
INSERT INTO SUSCRIPCIONES  
VALUES (15, 'Básico', 9.99, '12345678A');
```

**Solución:** Solo se necesita NIF\_Cliente, la cuenta se obtiene de CLIENTE.

### Anomalía de Actualización

#### En modelo denormalizado:

Si el cliente 12345678A cambia de cuenta, habría que actualizar TODOS los registros:

```
UPDATE SUSCRIPCIONES_DENORM  
SET Cuenta_Bancaria = 'ES222222222'  
WHERE NIF_Cliente = '12345678A';
```

**Riesgo:** Si alguna actualización falla, base de datos queda inconsistente.

#### En modelo normalizado:

```
UPDATE CLIENTE  
SET Cuenta_Bancaria = 'ES222222222'  
WHERE NIF = '12345678A';
```

**Solución:** Una única actualización en CLIENTE, sin riesgo de inconsistencia.

### Anomalía de Eliminación

#### En modelo denormalizado:

Si se elimina la única suscripción de un cliente, se pierden sus datos de contacto y cuenta bancaria.

#### En modelo normalizado:

Los datos de CLIENTE se mantienen independientemente de SUSCRIPCIONES, previniendo pérdida de información.

---

## Conclusión

Tras aplicar el proceso completo de normalización hasta Tercera Forma Normal y transformar sistemáticamente todas las relaciones del modelo Entidad-Relación, se obtiene un modelo relacional robusto, coherente y optimizado.

### Características del Modelo Final

#### Consistencia de datos:

- Eliminación de redundancias innecesarias
- Cada dato reside en un único lugar
- Actualizaciones atómicas garantizan coherencia

#### Integridad referencial:

- Todas las claves foráneas están claramente definidas
- Restricciones NOT NULL garantizan presencia de datos críticos
- Imposible crear registros huérfanos o referencias rotas

#### Escalabilidad:

- El modelo permite crecimiento sin reestructuración masiva
- Nuevos registros se añaden sin afectar la estructura
- Capacidad de soportar miles de registros eficientemente

#### Semántica empresarial:

- Las relaciones reflejan correctamente el dominio del negocio
- Cada relación tiene sentido comercial y operacional claro

#### Normalización garantizada:

- Cumplimiento de 3FN elimina anomalías de inserción, actualización y eliminación
- Previene corrupción de datos y mantiene consistencia automática
- Cumple estándares de calidad de bases de datos

### Recomendaciones para Implementación

1. **SGBDR Recomendado:** MySQL 8.0+, PostgreSQL 12+, SQL Server 2019+ o similar
2. **Índices:** Crear índices en todas las claves foráneas para optimizar joins
3. **Backups:** Implementar estrategia de backups automáticos diarios
4. **Auditoría:** Considerar añadir timestamps de creación/actualización
5. **Versiónado:** Usar control de versiones para scripts DDL

---

## Referencias

[1] Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6), 377-387. <https://doi.org/10.1145/362384.362685>

[2] Date, C. J. (2013). *An Introduction to Database Systems* (8th ed.). Pearson Education.

[3] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson Education.

[4] ISO/IEC 9075:2016. Information technology -- Database languages -- SQL. International Organization for Standardization.

[5] Ullman, J. D., & Widom, J. (2008). *A First Course in Database Systems* (3rd ed.). Pearson Education.

---

**Fin del documento**

*Documento generado automáticamente | Diciembre 2025*

*Versión profesional normalizada a 3FN sin dependencias transitivas*

*Incluye análisis completo, sentencias SQL DDL, tabla de contenidos con enlaces navegables, y recomendaciones de implementación*