# Evaluación de Normalización (1FN, 2FN, 3FN)

A continuación se presenta la evaluación de normalización para todas las tablas del modelo relacional. Cada tabla incluye tres registros de ejemplo y un análisis para cada una de las tres primeras formas normales.

## Tabla: usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| usu\_id | usu\_nombre | usu\_correo | usu\_telefono | usu\_rol |
| 1 | Laura Gómez | laura@gmail.com | 3201112233 | Cliente |
| 2 | Mario Rodríguez | mario@gmail.com | 3104445566 | Diseñador |
| 3 | Ana Martínez | ana.martinez@gmail.com | 3137778899 | Administrador |

* 1FN: Cumple. Todos los valores son atómicos (sin repeticiones ni listas).
* 2FN: Cumple. La clave primaria es simple (usu\_id) y todos los atributos dependen totalmente de ella.
* 3FN: Cumple. No hay dependencias transitivas entre atributos no clave.

## Tabla: opcion\_personalizacion

|  |  |
| --- | --- |
| opc\_id | opc\_nombre |
| 1 | Tipo de gema |
| 2 | Tamaño de piedra |
| 3 | Diseño del engaste |

* 1FN: Cumple. Valores simples y únicos por celda.
* 2FN: Cumple. Clave primaria simple (opc\_id) y opc\_nombre depende directamente de ella.
* 3FN: Cumple. No existen dependencias transitivas.

## Tabla: valor\_personalizacion

|  |  |  |
| --- | --- | --- |
| val\_id | opc\_id | val\_nombre |
| 1 | 1 | Zafiro |
| 2 | 1 | Esmeralda |
| 3 | 2 | 1.5 mm |

* 1FN: Cumple. Cada campo contiene un solo valor.
* 2FN: Cumple. val\_nombre depende únicamente de val\_id, y opc\_id cumple como clave foránea.
* 3FN: Cumple. No hay datos derivados ni redundantes.

## Tabla: personalizacion

|  |  |  |  |
| --- | --- | --- | --- |
| per\_id | cli\_id | opc\_id | val\_id |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 3 |
| 3 | 2 | 1 | 2 |

* 1FN: Cumple. Cada celda contiene un valor único y claro.
* 2FN: Cumple. La clave es per\_id, y todas las dependencias son completas.
* 3FN: Cumple. No hay atributos derivados entre sí.

## Tabla: estado\_pedido

|  |  |
| --- | --- |
| est\_id | est\_nombre |
| 1 | En diseño |
| 2 | En producción |
| 3 | Enviado |

* 1FN: Cumple. Celdas simples y atómicas.
* 2FN: Cumple. est\_nombre depende directamente de est\_id.
* 3FN: Cumple. No hay atributos que dependan unos de otros.

## Tabla: pedido

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ped\_id | cli\_id | dis\_id | est\_id | per\_id | ped\_codigo | ped\_fecha\_creacion | ped\_comentarios |
| 1 | 1 | 2 | 1 | 1 | PED-0001 | 2025-01-12 10:00:00 | Prefiere piedra azul |
| 2 | 1 | 2 | 2 | 2 | PED-0002 | 2025-01-20 12:15:00 | Montura en oro blanco |
| 3 | 2 | 2 | 3 | 3 | PED-0003 | 2025-02-01 09:30:00 | Sin comentarios |

* 1FN: Cumple. Todos los campos son simples y sin repeticiones.
* 2FN: Cumple. La clave es ped\_id y los demás campos dependen de ella.
* 3FN: Cumple. No hay campos que dependan entre sí. La clave alternativa ped\_codigo es válida pero no genera redundancia.

## Tabla: render\_3d

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ren\_id | ped\_id | dis\_id | ren\_imagen | ren\_fecha\_aprobacion |
| 1 | 1 | 2 | render1.png | 2025-01-14 14:00:00 |
| 2 | 2 | 2 | render2.png | 2025-01-22 09:00:00 |
| 3 | 3 | 2 | render3.png | 2025-02-03 11:30:00 |

* 1FN: Cumple. Los campos tienen valores atómicos.
* 2FN: Cumple. La clave es ren\_id y todas las columnas dependen de ella.
* 3FN: Cumple. No hay atributos redundantes ni transitivos.

## Tabla: foto\_producto\_final

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fot\_id | ped\_id | dis\_id | fot\_imagen\_final | fot\_fecha\_subida |
| 1 | 1 | 2 | final1.jpg | 2025-01-25 15:00:00 |
| 2 | 2 | 2 | final2.jpg | 2025-02-03 13:00:00 |
| 3 | 3 | 2 | final3.jpg | 2025-02-12 08:45:00 |

* 1FN: Cumple. Todos los campos contienen un único valor.
* 2FN: Cumple. fot\_id es clave primaria y los datos se relacionan directamente con ella.
* 3FN: Cumple. No hay atributos que dependan entre sí o causen redundancia.

**OPCIÓN 2 (RECOMENDADA): Mantener cli\_id directamente en pedido**

**Razonamiento**

Aunque implica cierta **redundancia controlada**, esta opción mejora la **claridad del modelo**, facilita el análisis, consultas y generación de reportes, y **mantiene la trazabilidad del cliente** a lo largo del proceso.

**Justificación**

* ✅ **Mejora la legibilidad del modelo** y la autonomía de la tabla pedido.
* ✅ **Facilita las consultas operativas** y analíticas comunes.
* ✅ **La redundancia es mínima**, perfectamente controlada y coherente.
* ✅ **No rompe la 3FN**, ya que cli\_id no depende de otro campo, sino que representa una relación directa y funcional: *“un pedido es realizado por un cliente”*.
* ✅ **Permite separar lógicamente los roles** (cliente y diseñador) dentro del mismo pedido.

**🧠 Consideración Técnica Avanzada**

En modelos reales, se busca **un equilibrio entre pureza de normalización y usabilidad práctica**.  
Una pequeña redundancia bien documentada y coherente puede ser ventajosa, siempre que:

* No genere ambigüedad.
* No cree problemas de integridad.
* Aporte claridad o rendimiento.

En este caso, **mantener cli\_id en pedido es una decisión deliberada, funcional y fundamentada**, que cumple tanto con los principios de diseño como con los requerimientos operativos del sistema.