

# Proceso de Normalización para Esquema de Base de Datos - Estadías

---

Este documento contiene la justificación del proceso de normalización realizado para el esquema de base de datos **ESTADIA**, el cual modela las estadías de pasajeros en diferentes hoteles.

## Esquema Original

---

### Tabla: ESTADIA

- **Atributos:** dniCliente, codHotel, cantidadHabitaciones, direccionHotel, ciudadHotel, dniGerente, nombreGerente, nombreCliente, ciudadCliente, fechaInicioHospedaje, cantDiasHospedaje, numeroHabitacion

### Restricciones

1. Existe un único gerente por hotel. Un gerente podría gerenciar más de un hotel.
2. Un cliente puede realizar la estadía sobre más de una habitación del hotel en la misma fecha, y para cada habitación puede reservar diferentes cantidades de días.
3. cantidadHabitaciones indica la cantidad de habitaciones existentes en un hotel.
4. El código de hotel (codHotel) es único y no se repite en diferentes ciudades.
5. Un cliente puede realizar reservas en diferentes hoteles para la misma fecha.
6. numeroHabitacion puede repetirse en distintos hoteles.
7. En la misma direccionHotel de una ciudadHotel puede haber más de un hotel.

## Paso 1: Dependencias Funcionales (DFs) y Claves Candidatas

---

### Dependencias Funcionales Identificadas

- `codHotel` determina `cantidadHabitaciones`, `direccionHotel`, `ciudadHotel`, `dniGerente`.
- `dniCliente` determina `nombreCliente`, `ciudadCliente`.
- `dniGerente` determina `nombreGerente`.

## Clave Candidata

- La combinación (`dniCliente`, `codHotel`, `fechaInicioHospedaje`, `numeroHabitacion`) se selecciona como **Clave Primaria**, ya que identifica de forma única una estadía específica de un cliente en un hotel.

## Paso 2: Elección de la Clave Primaria

---

- Optamos por la clave primaria (`dniCliente`, `codHotel`, `fechaInicioHospedaje`, `numeroHabitacion`) que permite identificar una estadía única.

## Paso 3: Normalización hasta la 3FN

---

### Primera Forma Normal (1FN)

Nos aseguramos de que todos los atributos tengan valores simples e indivisibles. Este esquema cumple con la Primera Forma Normal (1FN) porque no hay atributos con varios valores.

### Segunda Forma Normal (2FN)

- `nombreCliente` y `ciudadCliente` dependen solo de `dniCliente`, por lo tanto estos atributos se mueven a una nueva tabla `CLIENTE(dniCliente, nombreCliente, ciudadCliente)`.
- `nombreGerente` depende solo de `dniGerente`, así que lo trasladamos a una nueva tabla `GERENTE(dniGerente, nombreGerente)`.
- `cantidadHabitaciones`, `direccionHotel`, y `ciudadHotel` dependen solo de `codHotel`, así que estos atributos se mueven a una nueva tabla `HOTEL(codHotel, cantidadHabitaciones, direccionHotel, ciudadHotel, dniGerente)`.

### Tercera Forma Normal (3FN)

Para cumplir con la 3FN, eliminamos dependencias transitivas adicionales:

- dniGerente determina nombreGerente, lo que se resuelve con la tabla GERENTE, dejando a HOTEL sin dependencias transitivas.

## Esquema Normalizado Final

---

### 1. Tabla Estadia

- dniCliente (Clave foránea que referencia a Cliente)
- codHotel (Clave foránea que referencia a Hotel)
- fechaInicioHospedaje
- numeroHabitacion
- cantDiasHospedaje
- Clave primaria compuesta: (dniCliente, codHotel, fechaInicioHospedaje, numeroHabitacion)

### 2. Tabla Cliente

- dniCliente (Clave primaria)
- nombreCliente
- ciudadCliente

### 3. Tabla Hotel

- codHotel (Clave primaria)
- cantidadHabitaciones
- direccionHotel
- ciudadHotel
- dniGerente (Clave foránea que referencia a Gerente)

### 4. Tabla Gerente

- dniGerente (Clave primaria)
- nombreGerente