



# **NORMAS BÁSICAS PARA NOMBRAR ENTIDADES (TABLAS) Y ATRIBUTOS (CAMPOS) EN UNA BASE DE DATOS**

UPEMOR

# Usar nombres claros y significativos

- **Qué significa:** Los nombres deben indicar claramente qué contienen.
- **Ejemplo mal:** t1, x, dato1
- **Ejemplo bien:** Clientes, Productos, FechaNacimiento

Esto ayuda a entender la base de datos sin tener que adivinar.

# No usar espacios ni caracteres especiales

- **Qué significa:** En lugar de espacios, usa *guion bajo* o escribe en **camelCase**.
- **Ejemplo mal:** Nombre del Cliente
- **Ejemplo bien:** nombre\_cliente o nombreCliente

 Evita usar acentos, ñ, signos de interrogación, etc.

# Nombres en singular para atributos y plural para tablas

- **Qué significa:** Las tablas representan conjuntos de datos, los campos representan un solo dato.
- **Ejemplo bien:**
  - Tabla: Clientes
  - Atributos: id\_cliente, nombre, telefono

Es más lógico y mantiene una estructura coherente.

# Usar prefijos o sufijos si es necesario para mayor claridad

- **Qué significa:** Puedes agregar algo al nombre para indicar su función o relación.
- **Ejemplo bien:**
  - id\_cliente (clave primaria)
  - fk\_producto (clave foránea de producto)

Esto ayuda a entender relaciones entre tablas.

# Ser consistente

- **Qué significa:** Usa el mismo estilo de nombres en toda la base.
- Si decides usar guion\_bajo, úsalo siempre.
- Si decides escribir en minúsculas, hazlo en todo.

La consistencia hace que todo sea más legible y fácil de mantener.

# Ejemplo completo:

**Tabla:** Estudiantes

**Campos:**

- id\_estudiante
- nombre
- apellido
- correo
- fecha\_nacimiento

# Recomendaciones extra

- Empieza los nombres de las tablas y campos con una letra, no con número.
- Usa inglés o español, pero **no mezcles idiomas**.
- Usa minúsculas por compatibilidad (aunque SQL no distingue entre mayúsculas y minúsculas).



# ¿Cuándo usar mayúsculas y guiones bajos en nombres de campos/tablas?

## 1. snake\_case (con guiones bajos)

- **Cómo se escribe:** Todo en minúsculas y palabras separadas con guiones bajos.
- **Ejemplo:** nombre\_alumno, fecha\_nacimiento, id\_cliente
- **Ventajas:**
  - Muy usado en bases de datos relacionales (SQL).
  - Fácil de leer.
- **Recomendado para:** SQL (MySQL, PostgreSQL, SQLite), Python, PostgreSQL.

# ¿Cuándo usar mayúsculas y guiones bajos en nombres de campos/tablas?

## 2. camelCase

- **Cómo se escribe:** La primera palabra en minúscula y cada palabra siguiente empieza con mayúscula.
- **Ejemplo:** nombreAlumno, fechaNacimiento, idCliente
- **Ventajas:**
  - Muy común en lenguajes de programación como JavaScript, Java.
  - Útil si la base de datos se va a usar junto con código en esos lenguajes.
- **Recomendado para:** Programación orientada a objetos, APIs.

# ¿Cuándo usar mayúsculas y guiones bajos en nombres de campos/tablas?

## 3. PascalCase (también llamado UpperCamelCase)

- **Cómo se escribe:** Cada palabra empieza con mayúscula, incluso la primera.
- **Ejemplo:** NombreAlumno, FechaNacimiento, IdCliente
- **Ventajas:**
  - Muy usado para nombrar clases o entidades en programación.
- **Recomendado para:** Nombres de **tablas** si se usa estilo Pascal.

# Recomendación

Para bases de datos en SQL (lo más común en clases):

Elemento	Estilo sugerido	Ejemplo
Nombres de tablas	PascalCase o plural	Alumnos, Productos
Nombres de campos	snake_case	nombre_alumno, fecha_nacimiento

# Reglas para usar mayúsculas o minúsculas

- **En bases de datos SQL**, se recomienda usar **minúsculas** para evitar problemas con sistemas que distinguen mayúsculas y minúsculas (como PostgreSQL).
- Evita usar mayúsculas si no es necesario, especialmente en SQL.
- Usa mayúscula solo al **inicio de cada palabra** si decides usar camelCase o PascalCase.

# Tipos de campos en bases de datos (tipos de datos)

## 1. INT (Integer / Entero)

- **Qué guarda:** Números enteros (sin decimales).
- **Ejemplo:** 5, 2023, -15
- **Uso común:** Edad, número de control, cantidad de productos.
- **Ejemplo de campo:** edad INT

# Tipos de campos en bases de datos (tipos de datos)

## 2. DOUBLE o FLOAT (Decimal)

- **Qué guarda:** Números con decimales.
  - **Ejemplo:** 3.14, 1500.75, -0.01
  - **Uso común:** Precios, medidas, porcentajes.
  - **Ejemplo de campo:** precio DOUBLE
- DOUBLE guarda más precisión que FLOAT.

# Tipos de campos en bases de datos (tipos de datos)

- **3. VARCHAR(n)**

- **Qué guarda:** Texto o cadenas de caracteres, con **longitud variable** hasta un máximo n.
- **Ejemplo:** 'Juan', 'México', 'correo@gmail.com'
- **Uso común:** Nombres, apellidos, correos, direcciones.
- **Ejemplo de campo:** nombre VARCHAR(50)  
VARCHAR(50) guarda hasta 50 caracteres.



# Tipos de campos en bases de datos (tipos de datos)

## 4. CHAR(n)

- **Qué guarda:** Texto con longitud **fija** de n caracteres.
- **Ejemplo:** 'MX', 'AB' (si CHAR(2))
- **Uso común:** Códigos postales, abreviaciones, claves fijas.
- **Ejemplo de campo:** codigo CHAR(5)

Se usa cuando siempre tendrá el mismo tamaño.

# Tipos de campos en bases de datos (tipos de datos)

## 5. DATE

- **Qué guarda:** Fechas en formato AAAA-MM-DD.
- **Ejemplo:** 2025-06-06
- **Uso común:** Fechas de nacimiento, registro, eventos.
- **Ejemplo de campo:** fecha\_nacimiento DATE

# Tipos de campos en bases de datos (tipos de datos)

## 6. DATETIME

- **Qué guarda:** Fecha y hora completas.
- **Ejemplo:** 2025-06-06 14:30:00
- **Uso común:** Registro de actividad, hora de entrada/salida.
- **Ejemplo de campo:** fecha\_registro DATETIME

# Tipos de campos en bases de datos (tipos de datos)

## 7. BOOLEAN

- **Qué guarda:** Solo dos valores: TRUE o FALSE (1 o 0).
- **Ejemplo:** 1 (sí), 0 (no)
- **Uso común:** Activo/inactivo, aprobado/no aprobado.
- **Ejemplo de campo:** es\_activo BOOLEAN

# Consejos finales

- No uses VARCHAR para todo. Escoge el tipo adecuado según el dato.
- Usa INT para números que no necesitan decimales.
- Usa DECIMAL para montos de dinero.
- Usa DATE, TIME y DATETIME correctamente.
- Evita espacios y mayúsculas inconsistentes en los nombres de los campos.