

# Política de Nomenclatura para el Equipo de Desarrollo

## Objetivo

El propósito de este documento, es que se establezca un conjunto de reglas que permitan la consistencia, legibilidad y mantenibilidad del código en todo el equipo de desarrollo.

## Ámbito de Aplicación

Esta política se aplica a todas las bases de datos, variables, funciones, clases, y uso de Git en el equipo de desarrollo de la compañía.

### 1. Variables

#### 1.1 Variables en Lenguajes de Programación

- Formato: camelCase para variables locales y snake\_case para variables globales o constantes.
- Ejemplo: userName, order\_discount

#### 1.2 Constantes

- Formato: UPPER\_SNAKE\_CASE
- Ejemplo: MAX\_VALUES, DEFAULT\_VALUE

### 2. Funciones y métodos

#### 2.1 Nombres de funciones

- Formato: camelCase
- Ejemplo: calculateTotal(), fetchUserData()

#### 2.2 Nombres de métodos y clases

- Formato: camelCase
- Ejemplo: getUserInfo(), setOrderDetails()

## 2.3 Funciones privadas

- Formato: camelCase con un guion bajo inicial
- Ejemplo: `_calculateDiscount()`, `_fetchInternalData()`

## 3. Clases y objetos

### 3.1 Nombres de clases

- Formato: PascalCase
- Ejemplo: `UserProfile`, `OrderManager`

### 3.2 Nombres de objetos

- Formato: camelCase
- Ejemplo: `userProfile`, `orderManager`

## 4. Bases de datos

### 4.1 Nombres de bases de datos

- Formato: snake\_case
- Ejemplo: `customer_management`, `inventory_system`

### 4.2 Nombres de tablas

- Formato: snake\_case
- Prefijo: Si es necesario, se debe usar un prefijo que indique el módulo o la funcionalidad. También siempre algo que indique plural en el nombre de la tabla.
- Ejemplo: `user_accounts`, `order_details`

### 4.3 Nombres de columnas

- Formato: snake\_case
- Prefijo: Deben ser lo más descriptivos posible y, si es necesario, incluir el nombre de la tabla.
- Ejemplo: `user_id`, `order_date`

#### 4.4 Indices

- Formato: snake\_case
- Prefijo: idx\_
- Ejemplo: idx\_user\_email, idx\_order\_date

### 5. Git

#### 5.1 Nombres de ramas

- Formato: feature/, bugfix/, hotfix/, release/
- Ejemplo: feature/add-auth, bugfix/fix-order-detail, hotfix/security-patch, release/v1.1.0

#### 5.2 Mensajes de commit

- Estructura: [Tipo] Descripción corta
- Tipos: feat (nueva funcionalidad), fix (corrección de errores), refactor (refactorización de código), docs (cambios en la documentación), style (cambios de estilo), test (añadir/modificar tests)
- Ejemplo: [feat] Implement login functionality, [fix] Correct total calculation error

#### 5.3 Tags

- Formato: vX.Y.Z
- Ejemplo: v1.0.0, v1.1.3

### 6. Otros

#### 6.1 Nombres de archivos

- Formato: snake\_case y archivos de configuración
- Indicativo: Incluir un sufijo que indique el tipo de archivo.
- Ejemplo:
  - Controladores: user.controller.ts
  - Modelos: user.model.ts
  - Servicios: user.service.ts

## 6.2 Directorios

- Formato: kebab-case (guiones medios)
- Ejemplo: src, config, tests