

Ejercicio 6: Nomenclatura

Crea un documento de políticas de nomenclatura para el equipo de desarrollo de una compañía. La política debe incluir nomenclatura de: bases de datos, variables, funciones, clases, git, etc.

Políticas de Nomenclatura:

A continuación, se establecen las políticas de nomenclatura que se seguirán en el proyecto. Estas están agrupadas en cuatro pilares: Base de datos, Código, Archivos y Git.

Antes de continuar con las políticas, es importante tener en cuenta que estas van de la mano con el uso de buenas prácticas que deben ser consideradas transversalmente, tales como: mantener los nombres lo más descriptivos posible, evitar abreviaturas extrañas, y tener siempre actualizado el repositorio.

Nomenclatura de Bases de Datos

- Nombre de la Base de Datos: Utilizar snake_case para el nombramiento, intentando ser lo más descriptivos posible. Ejemplos: banco_bogota_bd, seguros_alfa_bd.
- Nombre de las Tablas: Utilizar snake_case para el nombramiento, intentando ser lo más descriptivos posible y en singular. Ejemplos: usuario, producto, pedido.
- Nombre de las Columnas: Utilizar snake_case para el nombramiento, intentando ser lo más descriptivos y claros posible. Ejemplos: usuario_id, fecha_alta, pedido_id.

Nomenclatura de Código

- Variables: Usar camelCase para nombrar las variables, siempre intentando reflejar el propósito de la variable. No usar prefijos como lista o array. Para referirse a un arreglo de objetos, usar el nombre en plural. Ejemplo: en lugar de listaProducto, usar productos.
- Constantes: Usar UPPER_SNAKE_CASE para nombrarlas, siendo siempre descriptivos y claros. Ejemplos: VEL_MAX, API_URL, ERROR_404.
- Funciones: Usar camelCase para nombrarlas, siempre intentando ser lo más descriptivo posible y comenzando con un verbo. Ejemplos: calcularImpuesto(), crearProducto(), responder().

- Clases: Usar PascalCase para nombrarlas, siempre intentando ser lo más descriptivo posible. Ejemplos: Producto, Usuario, ProcesosError.

Nomenclatura de Archivos y Directorios

- Archivos: Usar kebab-case para nombrarlos, siempre intentando ser lo más descriptivo posible y reflejar el contenido del archivo. Después del nombre, usar una pre-extensión de archivo para clasificar si este es un .service, .controller, etc. Ejemplos: producto.service.ts, servicio-compra.controller.js.
- Directorios: Usar kebab-case para nombrarlos, siempre intentando ser lo más descriptivo posible y reflejar el propósito del directorio. Ejemplos: productos, servicios.

Nomenclatura de Git

- Ramas: Usar kebab-case para nombrarlas, utilizando prefijos para indicar el tipo de rama:
 - feature/ para nuevas características.
 - bugfix/ para corrección de errores.
 - hotfix/ para correcciones urgentes en producción.
 - release/ para preparar una versión.
 - Ejemplos: feature/user-authentication, bugfix/order-total-calculation.
- Commits: Usar una descripción breve al inicio seguida del comentario. Este debe ser siempre de índole local, intentando no mezclar en los comentarios otros proyectos.
 - feat: Una nueva característica.
 - fix: Una corrección de errores.
 - docs: Cambios en la documentación.
 - style: Cambios de formato, comas, etc.
 - refactor: Refactorización del código.
 - test: Agregar o corregir tests.
 - chore: Tareas de mantenimiento.
 - Ejemplo: feat: add user login feature, fix: correct order total calculation.