

# **Documento de Políticas Nomenclatura**

Prueba Técnica Desarrollador Full Stack

Master

Daniel Stiven Londoño Ortega  
Developer Full Stack

2024

<b>Base de datos</b>	<b>3</b>
<b>Carpetas</b>	<b>5</b>
<b>Archivos</b>	<b>5</b>
<b>Variables</b>	<b>5</b>
<b>Funciones</b>	<b>6</b>
<b>Clases</b>	<b>6</b>
<b>Constantes</b>	<b>6</b>
<b>Git</b>	<b>7</b>

# Base de datos

**Tablas** → El nombre de las tablas debe estar en plural (Usuarios), cambiando espacios por underscore (Usuario\_Pedido), luego de cada underscore se debe iniciar con mayúscula, antecedido por el esquema al que pertenecen las tablas (Core.Usuario - Core.Usuario\_Pedido).

**Campos** → Se utiliza el nombre natural utilizado para describir el atributo y procurando que se pueda inferir el significado, cambiando los espacios por underscore (Usuario\_Id), luego de cada underscore se debe iniciar con mayúscula, con una longitud máxima de 14 caracteres.

**Procedimientos almacenados** → Los procedimientos almacenados deben iniciar con las siglas PR, cambiar los espacios por underscore, luego de cada underscore se debe iniciar con mayúscula, seguido del nombre de esquema, seguido por la descripción de la acción del procedimiento, con una longitud máxima de 50 caracteres.

PR\_Core\_EliminarUsuario

**Trigger** → Los trigger deben iniciar con las siglas TR, cambiar los espacios por underscore, luego de cada underscore se debe iniciar con mayúscula, seguido del nombre de esquema, seguido por la descripción de la acción del trigger con una longitud máxima de 50 caracteres.

TR\_Core\_EnviarEmail

**Vistas** → Las vistas deben iniciar con las siglas V, cambiar los espacios por underscore, luego de cada underscore se debe iniciar con mayúscula, seguido del nombre de esquema, seguido por la descripción del contenido de la vista con una longitud máxima de 50 caracteres.

V\_Core\_PedidosUsuarios

**Índices** → Los índices deben iniciar con las siglas I, cambiar los espacios por underscore, luego de cada underscore se debe iniciar con mayúscula, seguido del nombre de esquema, seguido por el nombre abreviado de la tabla. seguido del nombre del campo con una longitud máxima de 30 caracteres.

I\_Core\_Usu\_Nombre

**Constraints** → Los constraints deben iniciar con la abreviación de la tabla, seguido del nombre del campo y luego las siglas PK si es primaria y FK si es foránea, cambiar los espacios por underscore, luego de cada underscore se debe iniciar con mayúscula, campo con una longitud máxima de 30 caracteres.

Usuario\_Id\_PK

Usuario\_Nombre\_FK

# Carpetas

- Todos los nombres de carpetas deben estar en minúsculas
- Cuando un nombre de carpeta consista en varias palabras, usar guiones para separarlas.

## Ejemplos:

src/components/api-services

# Archivos

- Todos los nombres de archivos deben estar en minúsculas
- Cuando un nombre de archivo consista en varias palabras, usar guiones para separarlas
- Usar la extensión del archivo que vamos a crear .js, .jsx, .tsx

## Ejemplos:

index.js  
app.js  
usuario.jsx

# Variables

- Los nombres de las variables deben estar en camelCase.
- Elegir nombres que describan claramente el propósito de la variable.
- Evitar nombres de una sola letra a menos que se utilicen en un bucle corto o como variable temporal.

## Ejemplos:

```
let usuarioNombre;  
let isActive;  
const maxItems = 10;
```

# Funciones

- Los nombres de las funciones deben estar en camelCase.
- Elegir nombres que describan claramente el propósito de la función
- Usar verbos de acción para indicar lo que hace la función.

## Ejemplos:

```
function eliminarUsuario(){  
    //Código  
}
```

# Clases

- Los nombres de las clases deben estar en PascalCase.
- Elegir nombres que describan claramente el propósito de la clase

## Ejemplos:

```
class PerfilUsuario {  
    //Código  
}
```

# Constantes

- Los nombres de las constantes deben estar en letras mayúsculas con guiones bajos separando las palabras.
- Elegir nombres que describan claramente el propósito de la constante.

## Ejemplos:

```
const MAX_ITEMS = 100;  
const API_URL = 'https://api.github.com/search/repositories?q=user:google'
```

# Git

- El nombre de la rama debe tener el número relacionado a la tarea o bug a solucionar.
- El nombre de la rama debe tener el tipo de push que se va a realizar, ejemplo:

**add** - Agregar nuevas funcionalidades

**fix** - Corregir de bugs

**hotfix** - Solucionar bugs o agregar funcionalidades a producción (Riesgos)

- El nombre de la rama debe tener la descripción de la rama que debe ser escrita en camelCase.
- El nombre de la rama no debe tener espacios y deben ser reemplazados por underscore.
- El nombre de la rama no debe ser mayor de 30 caracteres.

## Ejemplos:

12345\_add\_calcularValorPedido\_Develop

12345\_hotfix\_calcularValorPedido\_Prod