ESTáNDAR De base de datos

El estándar de base datos presente en este informe será el que el equipo de Scrum llevara a cabo a lo largo de su proyecto “Punto de Venta” en la materia de Integradora I.

Proyecto: “Punto de Venta”

# Objetivo del Informe

El siguiente informe establece los estándares de codificación y naming conventions que seguiremos para asegurar la consistencia y facilidad de mantenimiento en nuestra bases de datos MySQL que estaremos utilizando en el proyecto Punto de Ventas en la materia de Integradora I. Se enfoca en la capitalización de palabras reservadas y la convención de nombres de tablas, columnas, procedimientos almacenados, triggers, vistas y funciones para que sea más legible el código y el diseño de nuestra base de datos.

# Estándares Recomendados

#### Palabras Reservadas y Convenciones de Capitalización:

* + Las palabras reservadas de SQL, como SELECT, INSERT, UPDATE, deben estar en mayúsculas.
  + Los nombres de tablas,y otros elementos deben seguir una convención de capitalización.

#### Nombres de Tablas:

* + Utilizar: La primera letra del nombre de la tabla debe ir en mayúscula seguido de minúscula, si es más de una palabra se separa por un guión bajo al igual que la primera palabra comenzara con mayúscula seguida de minúsculas.
  + Ejemplo: Usuarios\_Registrados, Pedidos\_Cliente.

#### Nombres de Columnas:

* + Comenzar con minúscula, al igual que usar guion medio y mayúscula al iniciar cada nueva palabra.
  + Ejemplo: id\_Usuario, nombre\_Cliente, fecha\_Registro.

#### Nombres de Procedimientos Almacenados:

* + Utiliza el prefijo **proc** para reconocer los Procedimientos almacenados.
  + El prefijo será escrito en minúsculas y cada palabra nueva se debe separar por un guión bajo y comenzar con mayúscula seguido de minúsculas.
  + Ejemplo: sp\_Obtener\_Usuario\_PorId, SP\_ActualizarEstadoPedido.

#### Nombres de Triggers:

* + Utiliza el prefijo **trg** para reconocer los triggers.
  + El prefijo será escrito en minúsculas y cada palabra nueva se debe separar por un guion bajo y comenzar con mayúscula seguido de minúsculas.
  + Ejemplo: trg\_Actualizar\_Stock\_Producto, trg\_Audit\_Log.

#### Nombres de Vistas:

* + Utiliza el prefijo **vw** para reconocer las vistas.
  + El prefijo será escrito en minúsculas y cada palabra nueva se debe separar por un guión bajo y comenzar con mayúscula seguido de minúsculas.
  + Ejemplo: vw\_Resumen\_Ventas\_Anuales, vw\_Productos\_Agotados.

#### Convenciones de Codificación:

* + Separar claramente las secciones de código con comentarios descriptivos.
  + Seguir un estilo de escritura de código coherente entre los desarrolladores.

Beneficios de la Aplicación de Estándares

La implementación de estos estándares nos ofrecerá varios beneficios a lo largo de nuestro proyecto “Punto de Ventas”, a continuación las enlisto.

* *Consistencia:* Todos los elementos de la base de datos tienen nombres coherentes y fáciles de entender, facilitando la comprensión y el mantenimiento.
* *Claridad:* Los nombres descriptivos ayudan a entender rápidamente la estructura y el propósito de cada objeto en la base de datos.
* *Facilidad de Mantenimiento:* La uniformidad en los nombres y convenciones reduce el tiempo necesario para realizar cambios o correcciones.

### Ejemplos de Implementación

A continuación, se presentan ejemplos con código SQL que siguen los estándares que usaremos en este proyecto Punto de Ventas.

## Creación de una tabla

Texto

Descripción generada automáticamente

## Creación de Procedimiento almacenado

Texto

Descripción generada automáticamente

## Creación de Trigger con nombre

Texto

Descripción generada automáticamente

## Creación de Vistas

Texto

Descripción generada automáticamente

## Consulta SELECT con palabras reservadas en mayúsculas

Texto

Descripción generada automáticamente

## Actualizar datos

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## Conclusión

Adoptar y aplicar consistentemente estos estándares de codificación y naming conventions en bases de datos MySQL es fundamental para mejorar la eficiencia del desarrollo, asegurar la calidad del código y facilitar la colaboración entre nuestro equipo de desarrollo de scrum y gestión de bases de datos. Esto resulta en sistemas más robustos, mantenibles y fáciles de gestionar a largo plazo.