Documentacion de proyecto singleton: conexión a bases de datos

Johan Sebastián Villalobos Pelaez

# Modelo Singleton

El **Singleton** es un patrón de diseño creacional que garantiza que una clase tenga solo una instancia y proporciona un punto de acceso global a ella. Se utiliza comúnmente para:

* Controlar el acceso a recursos compartidos (como conexiones a bases de datos)
* Configuraciones globales
* Cuando se necesita exactamente una instancia de una clase

**Características principales:**

* Constructor privado para evitar instanciación directa
* Método estático público para obtener la instancia
* Instancia almacenada en una variable estática privada
* Creación perezosa (lazy initialization) de la instancia

# Cómo se aplicó el patrón Singleton

1. **Control de instanciación**:
   * El constructor privado previene la creación de instancias mediante new Conexion()
   * La única forma de obtener la instancia es a través de getInstancia()
2. **Acceso global**:
   * Cualquier clase en la aplicación puede acceder a la conexión mediante Conexion.getInstancia()
3. **Creación perezosa**:
   * La instancia se crea solo cuando se solicita por primera vez (if(instancia == null))
   * No se crea hasta que sea necesaria, optimizando recursos
4. **Reutilización**:
   * Todas las llamadas posteriores a getInstancia() devuelven la misma instancia creada

# Beneficios en este contexto

1. **Optimización de recursos**: Solo existe una conexión a la base de datos en todo el sistema
2. **Consistencia**: Todas las partes de la aplicación trabajan con la misma conexión
3. **Control centralizado**: La gestión de la conexión (conectar/desconectar) está en un solo lugar
4. **Facilidad de acceso**: No es necesario pasar la conexión como parámetro entre clases