# Aplicación del Patrón Factory Method en CRUD de Usuarios

## 1. Introducción

En este documento se explica la implementación del patrón de diseño creacional Factory Method aplicado a un CRUD (Create, Read, Update, Delete) de usuarios en Node.js con PostgreSQL. El objetivo principal es desacoplar la lógica de negocio de los controladores y centralizar la creación de objetos mediante una factoría.

## 2. Estructura del Proyecto

La estructura de carpetas utilizada es la siguiente:

backend/  
│  
├── index.js # Punto de entrada del servidor  
├── repositories/  
│ └── UsuarioRepository.js # Repositorio con la lógica de acceso a datos  
├── factories/  
│ └── UsuarioRepositoryFactory.js # Factoría para instanciar el repositorio de usuarios

## 3. Patrón Factory Method

El patrón Factory Method permite delegar la creación de objetos a una clase factoría, en lugar de instanciarlos directamente en el código. Esto facilita el mantenimiento, la extensibilidad y la prueba del sistema.

### Ejemplo aplicado

En nuestro caso, en lugar de crear el repositorio de usuarios directamente en index.js, se utiliza la factoría UsuarioRepositoryFactory para obtener una instancia. De esta forma, si en el futuro queremos cambiar la base de datos (por ejemplo, de PostgreSQL a MongoDB), solo necesitamos modificar la factoría y el repositorio, sin tocar el resto del código.

## 4. Componentes

### 4.1 UsuarioRepository

Contiene la lógica de acceso a datos de la tabla usuarios en PostgreSQL. Implementa métodos como listar(), obtenerPorId(), crear(), actualizar() y eliminar().

### 4.2 UsuarioRepositoryFactory

Se encarga de crear una única instancia del repositorio de usuarios. Esto asegura el uso del patrón Singleton junto con Factory Method, evitando múltiples conexiones innecesarias a la base de datos.

## 5. Beneficios

- Desacoplamiento entre controladores y lógica de acceso a datos.  
- Facilidad para cambiar la implementación del repositorio.  
- Código más limpio, organizado y mantenible.  
- Reutilización de lógica en diferentes partes del proyecto.

## 6. Conclusión

La aplicación del patrón Factory Method en el CRUD de usuarios permite una arquitectura más robusta y escalable. Gracias a este patrón, separamos responsabilidades y facilitamos la evolución futura del sistema.