



Universidad Nacional de Colombia - sede Bogotá
Facultad de Ingeniería
Departamento de Sistemas e Industrial
Curso: Ingeniería de Software 1 (2016701)

Patrones de Diseño (Entrega 4) - Grupo 7 UNEmployed

Integrantes:

- ANGEL DAVID GÓMEZ PASTRANA
- DIEGO FELIPE CABRERA SALAMANCA
- PABLO LUNA GUZMÁN
- JAURI ESTEBAN CORTÉS CÁRDENAS

Singleton

Para este proyecto se eligió utilizar PostgreSQL como el motor de base de datos. Por defecto, PostgreSQL permite un máximo de 100 conexiones en simultáneo; aún así, tener múltiples conexiones puede llevar a problemas de sincronización y acceso a los datos en tiempo real. Por esto, se va a implementar el patrón **Singleton** para nuestra conexión a la base de datos.

En el backend ya tenemos implementado adicional al uso del ORM seleccionado, tenemos declarada una clase **Database** que cuenta con los siguientes métodos:

- `getInstance()` retorna la instancia del ORM ya conectada a la base de datos
- `initialize()` para inicializar la conexión en caso de que aún no lo esté

Factory Method

Como se indica en nuestros requerimientos y luego en los casos de uso, más específicamente el caso relacionado al requerimiento [RF_5](#) la aplicación cuenta con dos tipos de registro:

- Email y contraseña
- OAuth de Google y Facebook

Para manejar estas dos modalidades de autenticación de manera escalable, se implementa el **Factory Method**. Desde acá se define una interfaz común para la creación de usuarios dependiendo de su autenticación, delegando en las subclases la responsabilidad de instanciar el tipo específico de autenticación según corresponda.

En nuestro caso, el backend posteriormente debe definir una clase abstracta **AuthFactory** que declare el método `createAuthHandler()`. Cada implementación concreta de la fábrica se encarga de crear el objeto correspondiente al tipo de autenticación:

- EmailAuthHandler para el login con correo y contraseña.
- GoogleAuthHandler o FacebookAuthHandler para los logins mediante OAuth.

De esta forma, el sistema puede interactuar con un mismo punto de acceso para todos los tipos de autenticación sin conocer los detalles internos de cómo se manejan cada uno de ellos.