### FATEC DR ARCHIMEDES LAMMOGLIA

Caio Simonassi

TECNICAS DE PROGRAMAÇÃO II

### **Atividade: Padrão de Projeto Singleton com Conexão de Banco de Dados**

Orientador: Profº Simone Mendes da Silva

INDAIATUBA

SETEMBRO/2025

##### **1. Exemplo de Conexão com Banco de Dados**

O Padrão de Projeto Singleton garante que uma classe tenha apenas uma única instância, fornecendo um ponto de acesso global a ela.

Um dos seus usos mais comuns é para gerenciar recursos compartilhados, como uma conexão com um banco de dados, o padrão garante que, não importa quantas vezes o objeto de conexão seja solicitado, a mesma e única instância será retornada.

O construtor é privado, o que impede a criação de novas instâncias diretamente, um método estático é a única forma de obter a instância da classe e a conexão é criada apenas uma vez, dentro da lógica de getInstance().

public class DatabaseConnection {

private static DatabaseConnection instance; // A única instância da classe.

private MongoClient mongoClient; // Objeto de conexão com o banco.

private MongoDatabase database; // O banco de dados a ser usado.

private DatabaseConnection() {

try {

// Construtor privado: cria a conexão apenas uma vez.

this.mongoClient = MongoClients.create("mongodb://localhost:27017");

this.database = mongoClient.getDatabase("mydatabase");

System.out.println("Conexão com o banco de dados estabelecida.");

} catch (Exception e) {

e.printStackTrace();

}

}

public static DatabaseConnection getInstance() { // Ponto de acesso global.

if (instance == null) {

instance = new DatabaseConnection(); // Cria a instância se ainda não existir.

}

return instance; // Retorna a única instância.

}

public MongoDatabase getDatabase() {

return database; // Método para obter o objeto do banco de dados.

}

}

Como Acessar a Classe

Para usar a conexão, você apenas precisa chamar o método getInstance() de qualquer lugar do seu código.

public class Main {

public static void main(String[] args) {

// Acessa a única instância da classe de conexão, garantindo que o construtor só seja chamado uma vez.

DatabaseConnection dbConnection = DatabaseConnection.getInstance();

// Obtém a referência ao banco de dados e a uma coleção, para começar a trabalhar.

MongoCollection<Document> collection = dbConnection.getDatabase().getCollection("usuarios");

// Exemplo: Insere um novo documento na coleção.

Document user = new Document("nome", "Jhonatan").append("idade", 25);

collection.insertOne(user);

System.out.println("Documento inserido com sucesso!");

}

}

##### **2. Pontos Positivos e Negativos do Padrão Singleton**

Alguns Pontos Positivos são a Economia de Recursos que Evita a criação de múltiplos objetos de conexão, o que economiza memória e tempo, o Controle Centralizado faz com que toda a aplicação use a mesma e única instância de conexão que pode ser acessada facilmente de qualquer parte do código.

Ja na parte dos Pontos Negativos nos temos o Acoplamento Forte, o código que usa a classe Singleton fica diretamente dependente dela, dificultando a troca por outra implementação, outro ponto é a Dificuldade de Teste que deixa mais complicado testar uma classe que depende de um Singleton, pois não é fácil substituí-lo por uma versão de teste e não é a melhor prática com Spring Boot, em projetos Spring Boot, usar um Singleton manual vai contra a filosofia do framework, que já gerencia a criação de instâncias de forma mais flexível usando Injeção de Dependência.