

PADRÃO DE PROJETO SINGLETON

Gerenciamento Eficiente de
Conexões com Banco de Dados em
Java Usando o Padrão Singleton

Éder

EBOOK

Conceito



O que é o padrão Singleton?


O **Singleton** é um **padrão de projeto criacional** que garante: Apenas **uma única instância** de uma classe.

Um **ponto de acesso global** a essa instância.

Evita múltiplas criações de objetos que devem ser únicos — como uma conexão ao banco de dados.



Exemplo

 **Exemplo prático: Conexão ao banco de dados (sem threads)** Imagine que você quer evitar múltiplas conexões ao banco em uma aplicação simples. O Singleton resolve isso criando uma única conexão, reutilizada sempre que for necessária.



Conexão

```
public class ConexaoBD {

    // Instância única da própria classe
    private static ConexaoBD instancia;

    // Objeto de conexão do JDBC
    private Connection conexao;

    // Dados de conexão
    private static final String URL = "jdbc:mysql://localhost:3306/seu_banco";
    private static final String USUARIO = "root";
    private static final String SENHA = "senha";

    // Construtor privado impede criação fora da classe
    private ConexaoBD() {
        try {
            conexao = DriverManager.getConnection(URL, USUARIO, SENHA);
            System.out.println("Conexão estabelecida.");
        } catch (SQLException e) {
            System.out.println("Erro ao conectar: " + e.getMessage());
        }
    }

    // Método de acesso à instância única
    public static ConexaoBD getInstancia() {
        if (instancia == null) {
            instancia = new ConexaoBD();
        }
        return instancia;
    }

    // Getter para a conexão
    public Connection getConexao() {
        return conexao;
    }
}
```



Classe de teste

```
public class TesteConexao {

    public static void main(String[] args) {
        ConexaoBD conexao1 = ConexaoBD.getInstancia();
        ConexaoBD conexao2 = ConexaoBD.getInstancia();

        // Verificando se é a mesma instância
        if (conexao1 == conexao2) {
            System.out.println("Singleton funcionando: mesma instância.");
        } else {
            System.out.println("Erro: instâncias diferentes.");
        }

        // Obtendo a conexão
        Connection conn = conexao1.getConexao();
        // Aqui você pode usar a conexão para executar queries
    }
}
```



Vantagens

✓ Vantagens do Singleton neste cenário

Reutilização da conexão, economizando recursos.

Controle centralizado de acesso ao banco.

Fácil de implementar em aplicações pequenas ou testes locais.

