



Implementação Avançada de CRUD em Java com Swing, JSP, JSF e Spring Boot

O padrão CRUD (Create, Read, Update, Delete) é fundamental em sistemas de gerenciamento de dados, e sua implementação em Java pode ser realizada de diferentes maneiras, dependendo da arquitetura e dos frameworks escolhidos. Neste artigo técnico, detalhamos a implementação de CRUD utilizando as tecnologias Swing, JSP, JSF e Spring Boot, aproveitando as capacidades do Java 17 para aplicações modernas.

Cada uma dessas tecnologias atende a diferentes necessidades de projeto e arquitetura, desde aplicações desktop até soluções web e microservices, sendo importante conhecer suas particularidades para selecionar a melhor abordagem.

1. CRUD com Swing

Swing é uma biblioteca gráfica robusta, adequada para aplicações desktop. Embora seu uso esteja em declínio, sua integração com JDBC ou ORM como Hibernate permite o desenvolvimento de sistemas desktop capazes de realizar operações CRUD de forma eficaz.

Estrutura do CRUD em Swing:

- **Create:** Captura de dados com componentes como JTextField e JButton, persistindo os dados via JDBC.
- **Read:** Exibição de dados em uma JTable com um TableModel customizado.
- **Update:** Edição de dados em campos pré-populados, seguida de atualização via SQL.
- **Delete:** Exclusão de registros da JTable e do banco de dados via operações SQL.



Exemplo de integração com JDBC:

```
JButton saveButton = new JButton("Salvar");

saveButton.addActionListener(e -> {

    String nome = nomeTextField.getText();

    String email = emailTextField.getText();

    Try {

        Connection connection = DriverManager.getConnection(DB_URL, USER,
PASS);

        String query = "INSERT INTO usuario (nome, email) VALUES (?, ?)";

        PreparedStatement preparedStatement =
connection.prepareStatement(query);

        preparedStatement.setString(1, nome);

        preparedStatement.setString(2, email);

        preparedStatement.executeUpdate();

        connection.close();

    } catch (SQLException ex) {

        Ex.printStackTrace();

    }

});
```



2. CRUD com JSP (JavaServer Pages)

JSP é utilizado para a construção de páginas web dinâmicas e ainda é visto em sistemas legados. Ele permite a mesclagem de HTML com código Java para operações CRUD, mas a separação de responsabilidades é limitada, o que pode dificultar a manutenção e evolução da aplicação.

Estrutura do CRUD em JSP:

- **Create:** Formulários HTML capturam os dados, processados em um Servlet ou diretamente no JSP.
- **Read:** Exibição de registros em tabelas HTML através de ResultSet em loop.
- **Update:** Similar à criação, mas com dados pré-populados.
- **Delete:** Botões ou links que invocam um Servlet para exclusão.

Exemplo de exibição de dados em JSP:

Jsp

```
<%@ page import="java.sql.*" %>
```

```
<%
```

```
    Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
```

```
    Statement stmt = conn.createStatement();
```

```
    ResultSet rs = stmt.executeQuery("SELECT * FROM usuario");
```

```
%>
```

```
<table>
```

```
    <tr><th>ID</th><th>Nome</th><th>Email</th></tr>
```

```
    <% while (rs.next()) { %>
```

```
        <tr>
```

```
            <td><%= rs.getInt("id") %></td>
```

```
            <td><%= rs.getString("nome") %></td>
```



```
<td><%= rs.getString("email") %></td>

</tr>

<% } %>

</table>
```

3. CRUD com JSF (JavaServer Faces)

JSF é um framework orientado a componentes que oferece um modelo robusto para o desenvolvimento de interfaces web. Sua principal vantagem é a separação de responsabilidades e o gerenciamento de estado, tornando-se adequado para aplicações web corporativas de grande porte.

Estrutura do CRUD em JSF:

- **Create:** Formulários JSF vinculados a Managed Beans, com controle de estado e persistência.
- **Read:** Exibição de listas com <h:dataTable> e integração com coleções em Managed Beans.
- **Update:** Formulários com campos pré-preenchidos, permitindo a edição de dados.
- **Delete:** Exclusão de registros diretamente da interface, invocando métodos do Managed Bean.

Exemplo de formulário de inserção com JSF:

Xml

```
<h:form>

<h:inputText value="#{usuarioBean.nome}" />

<h:inputText value="#{usuarioBean.email}" />

<h:commandButton value="Salvar" action="#{usuarioBean.salvar}" />

</h:form>
```



4. CRUD com Spring Boot

Spring Boot é a escolha moderna para aplicações Java, especialmente em arquiteturas de microservices e web. Com Spring Data JPA, as operações CRUD são abstraídas, permitindo foco na lógica de negócios sem se preocupar com detalhes de persistência.

Estrutura do CRUD em Spring Boot:

Create: Dados são capturados e enviados para um controlador que utiliza POST para persistir via Spring Data JPA.

Read: Dados são recuperados pelo controlador via repositórios JPA e exibidos na interface.

Update: Dados atualizados são enviados para o controlador e persistidos no banco.

Delete: Um método HTTP DELETE remove registros do banco de dados.

Exemplo de implementação CRUD com Spring Boot:

Java

```
@RestController
```

```
@RequestMapping("/usuarios")
```

```
Public class UsuarioController {
```

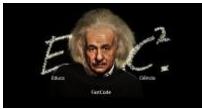
```
    @Autowired
```

```
    Private UsuarioRepository usuarioRepository;
```

```
    @GetMapping
```

```
    Public List<Usuario> listar() {
```

```
        Return usuarioRepository.findAll();
```



```
}
```

```
@PostMapping
```

```
Public Usuario criar(@RequestBody Usuario usuario) {
```

```
    Return usuarioRepository.save(usuario);
```

```
}
```

```
@PutMapping("/{id}")
```

```
Public Usuario atualizar(@PathVariable Long id, @RequestBody Usuario usuario) {
```

```
    Return usuarioRepository.findById(id)
```

```
        .map(u -> {
```

```
            u.setNome(usuario.getNome());
```

```
            u.setEmail(usuario.getEmail());
```

```
            return usuarioRepository.save(u);
```

```
        }).orElseThrow(() -> new ResourceNotFoundException("Usuário não encontrado"));
```

```
}
```

```
@DeleteMapping("/{id}")
```

```
Public void deletar(@PathVariable Long id) {
```

```
    usuarioRepository.deleteById(id);
```

```
}
```

```
}
```



Considerações Finais

A escolha da tecnologia ideal para a implementação de um sistema CRUD em Java depende do contexto do projeto, requisitos de escalabilidade, complexidade e usabilidade.

Swing é eficaz para aplicações desktop, mas está em declínio, sendo mais adequado para sistemas legados ou soluções offline.

JSP ainda é amplamente utilizado em aplicações legadas, mas sua abordagem misturando lógica de negócios com a camada de visualização já foi superada por frameworks mais modernos.

JSF oferece uma abordagem robusta para aplicações web empresariais, com forte suporte a componentes e integração com APIs padrão do Java EE.

Spring Boot é a escolha mais moderna, oferecendo facilidade de configuração, uma estrutura modular e uma integração eficiente com Spring Data JPA para persistência de dados.

Com a evolução contínua das tecnologias Java e o suporte a novos paradigmas como arquiteturas baseadas em microservices, é essencial que desenvolvedores dominem diferentes abordagens de CRUD e saibam quando aplicá-las para maximizar a eficiência e escalabilidade de suas soluções.

Por EducaCiência FastCode, esperamos que este guia técnico tenha oferecido insights valiosos para a implementação de sistemas robustos em Java, utilizando o padrão CRUD como base para manipulação de dados.