

22. Fases do Ciclo de Vida do JSF

O **JSF (JavaServer Faces)** possui um ciclo de vida bem definido, dividido em **seis fases principais**, controladas pela classe `javax.faces.lifecycle.Lifecycle`.

Essas fases são executadas de forma sequencial e controladas pela servlet padrão do JSF, chamada **FacesServlet**. Ela é responsável por receber requisições HTTP e processar a interação entre a **interface de usuário (UI)** e o **modelo de dados** da aplicação.

O ciclo de vida do JSF gira em torno de um objeto chamado **FacesContext**, criado por uma **fábrica de contexto**. Esse objeto centraliza todas as informações sobre a requisição, os componentes, mensagens e estado da aplicação durante o ciclo.

Resumo das 6 Fases do Ciclo de Vida JSF

1. Restaurar Apresentação (Restore View)

- Início do ciclo de vida.
- O JSF cria (ou restaura) a **árvore de componentes UI** da página solicitada.
- Cada componente recebe um **ID único** dentro da aplicação.
- O **FacesContext** começa a ser povoado com os dados da requisição e da UI.

2. Aplicar Valores Requisitados (Apply Request Values)

- Os **valores do formulário** enviados pelo usuário são lidos.
- Cada valor é atribuído ao seu componente correspondente.
- Se necessário, ocorre a **conversão de tipos** (por exemplo, de **String** para **int**).
- **Erros de conversão** geram mensagens e impedem o avanço das fases seguintes.

3. Processar Validações (Process Validations)

- Os valores convertidos são **validados** de acordo com regras específicas (como obrigatoriedade, tamanho, faixas de valores).
 - Se a validação falhar:
 - Uma mensagem de erro é gerada.
 - O ciclo **pula para a fase de renderização da resposta**, sem continuar para as próximas fases.
 - Exemplo: Validar que a quantidade em um carrinho de compras esteja entre 1 e 10.
-

4. Atualizar Valores do Modelo (Update Model Values)

- Os valores validados são atribuídos às propriedades dos **beans gerenciados (Managed Beans)**.
 - Aqui, os **dados da interface são sincronizados com o modelo de negócio**.
 - Valores incorretos podem passar aqui, mas ainda podem ser **validados na próxima fase** com regras de negócio mais complexas.
-

5. Invocar Aplicação (Invoke Application)

- Ações do usuário (como clicar em um botão ou submeter um formulário) são processadas.
 - Métodos no bean são executados (ex: salvar dados, calcular preço).
 - Se a ação retornar uma **string de navegação**, o JSF usa essa string para determinar qual página deve ser exibida em seguida.
-

6. Renderizar Resposta (Render Response)

- Fase final: A **página é renderizada** e enviada ao cliente.
- Se for a primeira requisição, os componentes são adicionados à árvore de visualização agora.

- Se for um **postback** (requisição subsequente), os componentes já estavam na árvore.
 - **Mensagens de erro** são exibidas nos componentes `<h:message />` ou `<h:messages />`, caso existam.
-

Resumo Visual (em ordem):

1. Restaurar Apresentação →
 2. Aplicar Valores Requisitados →
 3. Processar Validações →
 4. Atualizar Valores do Modelo →
 5. Invocar Aplicação →
 6. Renderizar Resposta
-

Eventos Suportados pelo JSF

O JSF suporta três principais tipos de eventos que podem ser manipulados durante o ciclo de vida:

1. **ValueChangeEvent**
 - Disparado quando o valor de um componente é alterado (ex: alteração em um campo de formulário).
2. **ActionEvent**
 - Disparado por componentes de ação, como `<h:commandButton>`, após o clique do usuário.
3. **PhaseEvent**
 - Disparado ao **entrar e sair de cada fase** do ciclo de vida.

- Pode ser usado para **auditoria**, **logging** ou **execução condicional** em fases específicas.