



**Estácio**

## **DESENVOLVIMENTO BACK-END CORPORATIVO COM JAVA E CLOUD**

**Pedro Martins Gomes de Santana / 202407343538**

### *RELATÓRIO DE PRÁTICA /*

#### *DESENVOLVIMENTO BACK-END CORPORATIVO COM JAVA E CLOUD*

*Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.*

#### *Objetivo da Prática*

*O objetivo desta prática foi implementar um sistema cadastral corporativo, aplicando os conceitos de arquitetura em camadas utilizando a plataforma Jakarta EE (JEE).*

*As metas específicas incluíram:*

- *Implementar persistência de dados utilizando JPA (Java Persistence API);*
- *Encapsular a lógica de negócio através de EJBs (Enterprise JavaBeans);*
- *Desenvolver a interface do usuário utilizando Servlets e JSPs;*
- *Aplicar o framework Bootstrap para melhoria do design e responsividade*
- *Documentação: Este arquivo (PDF) foi armazenado no repositório.*

---

#### *1º Procedimento / Camadas de Persistência e Controle*

##### *Relato da Execução*

*As etapas de Configuração de Conexão com o SQL Server e Estrutura Modular do Projeto foram concluídas com sucesso, demonstrando o domínio na configuração de ambiente para aplicações corporativas baseadas em Jakarta EE.*

*Sucesso na Configuração de Persistência (SQL Server e GlassFish):*

- Conexão JDBC no NetBeans: O driver mssql-jdbc-12.2.0.jre8.jar foi adicionado e configurado com sucesso. A conexão com o banco de dados loja no SQL Server foi estabelecida e testada com êxito no NetBeans.*
- Pool de Conexões no GlassFish: O Pool de Conexões (SQLServerPool) e o Recurso JNDI (jdbc/loja) foram criados e, o teste de conexão (ping-connection-pool SQLServerPool) obteve sucesso, garantindo que o servidor de aplicação pudesse gerenciar a fonte de dados.*

*Estrutura Modular:*

- Criação da estrutura modular do projeto Enterprise Application (CadastroEE):*
  - CadastroEE-ejb (Modelo e Controle - JPA e EJBs)*
  - CadastroEE-war (Visualização - Servlets e JSPs)*

*Bloqueio na Fase de Geração de Código:*

*Apesar da infraestrutura de persistência estar pronta, na fase de geração automática de código (Entidades JPA e Session Beans EJB) ocorreram falhas críticas no IDE que impediram a finalização da camada EJB.*

**✗ Justificativa da Paralisação**

*(Código e resultados não obtidos devido a problemas no NetBeans)*

<i>Categoría</i>	<i>Detalhe</i>
<i>ID do Erro</i>	<i>Wizard Failure</i>
<i>Descrição</i>	<i>Cannot invoke "org.netbeans.api.java.source.JavaSource.runModificationTask(...)" because "source" is null</i>
<i>Bloqueio Final</i>	<i>Erro de Compilação: CriteriaQuery sem referência à JPA</i>

Categoria	Detalhe

*Mesmo com orientação da IA e tentativas de correção manual, o erro persistiu e bloqueou o avanço.*

### *Análise e Conclusões do Procedimento 1*

#### *a) Organização de Projeto Corporativo no NetBeans*

- *EAR (Projeto Principal): Responsável por empacotar todos os módulos para o GlassFish.*
- *EJB Module (CadastroEE-ejb): Contém as Entidades (JPA) e as Regras de Negócio (EJBs).*
- *WAR Module (CadastroEE-war): Contém os Servlets e JSPs (Camada de Visualização).*
- *Conclusão: Representa a arquitetura em camadas padrão da plataforma Java EE.*

#### *b) Papel do JPA e EJB*

- *Tecnologia: JPA*
  - *Função: Persistência de dados utilizando Mapeamento Objeto-Relacional (ORM).*
- *Tecnologia: EJB*
  - *Função: Implementação da lógica de negócio com suporte do servidor (gerenciamento de transações, segurança, etc.).*

#### *c) Produtividade no NetBeans*

*O NetBeans oferece alta produtividade através de:*

- *Geração automática de Entidades JPA e Session Beans/Facades CRUD.*
- *Inserção facilitada de anotações essenciais (@PersistenceContext, @EJB).*

#### *d) O que são Servlets e Suporte do NetBeans*

- *Definição de Servlet: Processa requisições HTTP recebidas do cliente e encaminha a resposta, geralmente para uma página JSP.*

- *Suporte do NetBeans: Automatiza a criação do componente, configuração e roteamento das URLs.*

**e) Comunicação Servlet  $\rightleftarrows$  EJB**

- *Método: Feita via Injeção de Dependência utilizando a anotação @EJB.*
  - *Gerenciamento: O Container EJB é responsável por gerenciar as instâncias e transações automaticamente.*
- 

 **2º Procedimento / Interface Cadastral com Servlet e JSPs**

**Relato da Execução**

*Procedimento não iniciado.*

 **Justificativa da Paralisação**

*O desenvolvimento dependia da compilação bem-sucedida do módulo EJB do Procedimento 1. Sem EJB funcional, não foi possível realizar o deploy e, consequentemente, desenvolver a Camada Web.*

**Análise e Conclusões do Procedimento 2**

**a) Padrão Front Controller**

- *Funcionamento: Um único Servlet centraliza todas as requisições de entrada, agindo como ponto de controle.*
- *Arquitetura: Sua implementação é fundamental para o padrão MVC (Model-View-Controller), delegando as ações para o negócio e a visualização para os JSPs.*

**b) Diferenças e Semelhanças: Servlet x JSP**

Característica	Detalhe sobre Servlet	Detalhe sobre JSP
Foco	Lógica de controle e processamento.	Apresentação e view.
Código	Java com código HTML gerado.	HTML com código Java embutido (Scriptlets/EL/JSTL).

<i>Característica</i>	<i>Detalhe sobre Servlet</i>	<i>Detalhe sobre JSP</i>
<i>Execução</i>	<i>Compilado como uma Classe Java.</i>	<i>Convertido internamente para um Servlet antes de ser executado.</i>

c) *Redirect x Forward*

<i>Método</i>	<i>Funcionamento</i>	<i>URL</i>	<i>Uso Recomendado</i>
<i>sendRedirect()</i>	<i>Gera uma nova requisição HTTP por parte do cliente.</i>	<i>Muda para o novo endereço.</i>	<i>Utilizado, tipicamente, após operações de POST (pós-inclusão/alteração) para evitar o reenvio do formulário.</i>
<i>forward()</i>	<i>É uma troca interna de recurso gerenciada pelo servidor.</i>	<i>Não muda no navegador.</i>	<i>Utilizado para enviar dados (via RequestDispatcher) e controle para um JSP (Camada View).</i>

📌 *3º Procedimento / Design Responsivo com Bootstrap*

*Relato da Execução*

*Não iniciado devido aos bloqueios anteriores.*

*Análise e Conclusões do Procedimento 3*

*Tópicos sobre Bootstrap*

- *Uso do Bootstrap:* É um framework de front-end aplicado através de classes pré-definidas no código HTML do projeto.
- *Independência Estrutural:* O design é aplicado por meio de classes CSS ao invés de depender de seletores complexos como IDs e hierarquias específicas do HTML.

- *Responsividade: É garantida pelo Sistema de Grid (baseado em 12 colunas) e pela filosofia Mobile-First, adaptando o layout para diferentes tamanhos de tela.*
- 

## Conclusão Geral

A prática demonstrou:

- ✓ *Domínio prático da configuração de persistência (SQL Server + GlassFish JNDI).*
- ✓ *Domínio teórico da arquitetura Java EE em camadas.*
- ✓ *Capacidade de configuração de servidor e estrutura de módulos.*
- ✗ *Conclusão inviabilizada por erros estruturais do NetBeans.*
- ✗ *EJB não compilou → Camada Web não pôde ser desenvolvida.*

*Apesar do esforço contínuo, o erro de dependência do JPA persistiu e bloqueou o avanço do desenvolvimento do código dentro do tempo disponível.*