



FACULDADE ESTACIO
CURSO TECNÓLOGO EM DESENVOLVIMENTO FULL STACK

RAFAEL VALVERDE FONSECA

MISSÃO PRÁTICA – NÍVEL 4 – MUNDO 3
RPG0017 - Vamos integrar sistemas

Serrinha - BA

2025

RAFAEL VALVERDE FONSECA

MISSÃO PRÁTICA – NÍVEL 4 – MUNDO 3
RPG0017 – Vamos integrar sistemas

Trabalho apresentado à disciplina Back-end
Vamos integrar sistemas do Curso Tecnólogo em
Desenvolvimento Full Stack, período 2024.4 Flex,
como requisito parcial do relatório de
acompanhamento.

Tutoria: Jhonatan Alves

OBJETIVOS

Este relatório apresenta a composição do trabalho proposto para o Nível 4: vamos integrar sistemas, o qual está contido no semestre letivo no período 2024, apresentando todos os códigos solicitados, resultados da execução desses códigos e descrição de avaliação sobre o tema abordado, respondendo perguntas propostas pelo tutor.

Palavras-chave: JPA, JEE, EJB, Servlets, JSP, SQL Server, Glassfish, JDBC.

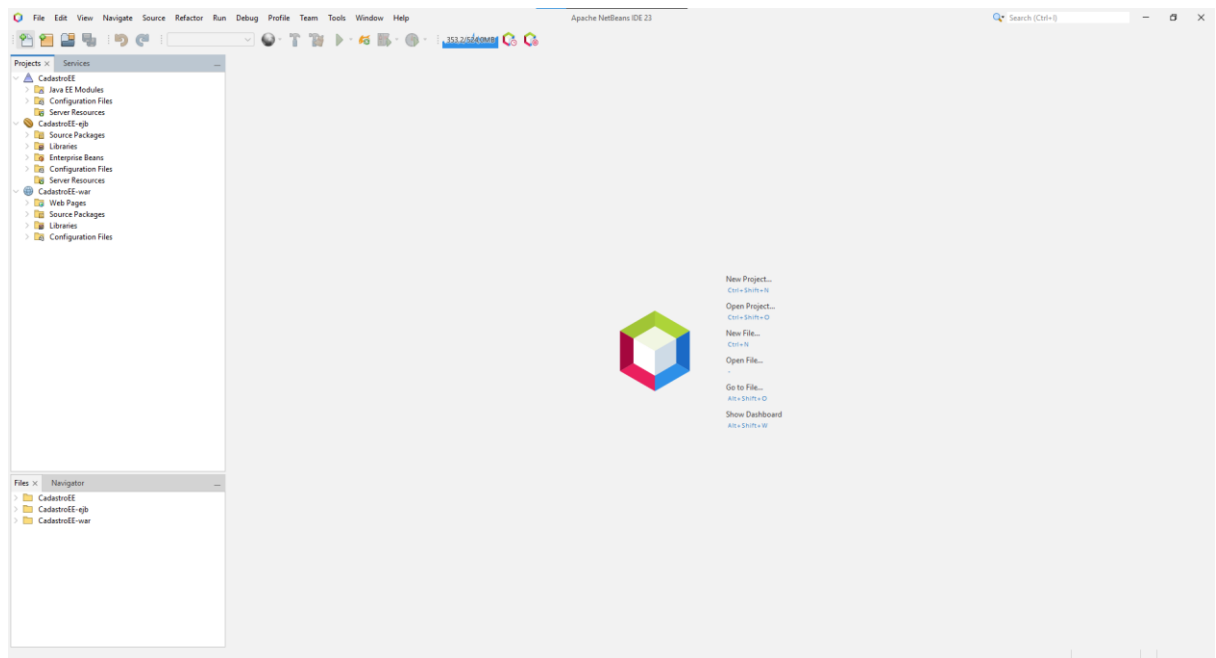
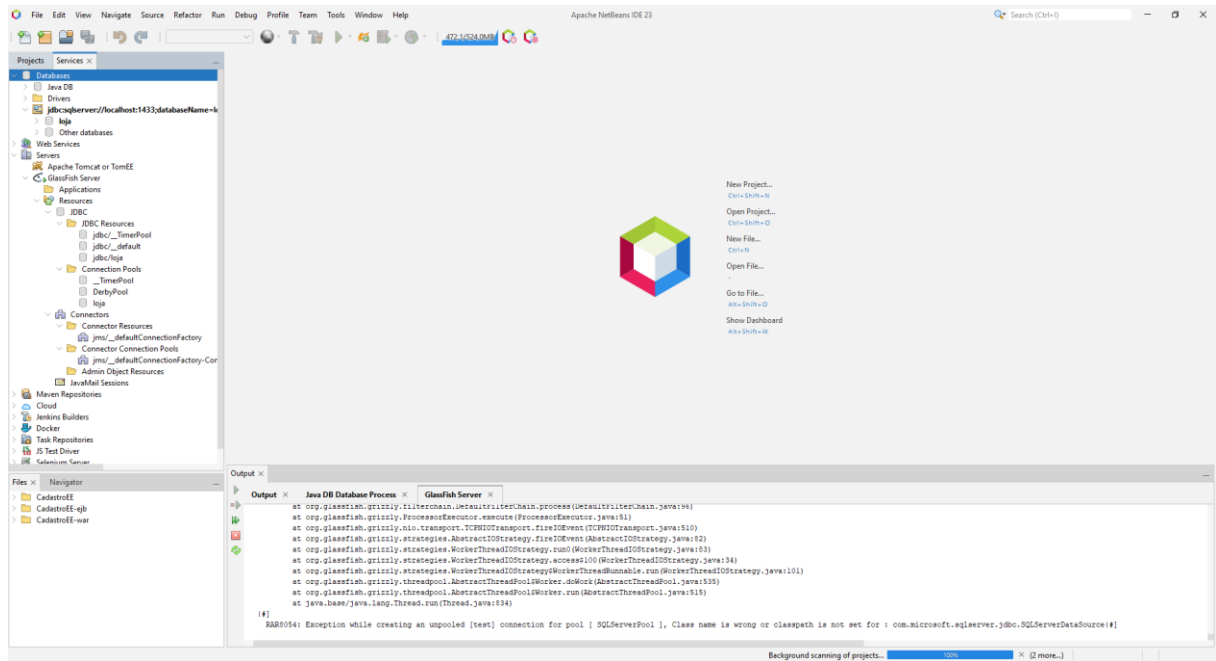
SUMÁRIO

1	RELATÓRIO	9
2	TELAS	9
3	ANÁLISE	11

1 RELATÓRIO

Esse trabalho tem o objetivo de implementar persistência em base em JPA, bem como regras de negócio na plataforma JEE, através de EJBs.

2 TELAS



The screenshot displays the Apache NetBeans IDE environment for a project named 'CadastroEE'. The interface is divided into several panes:

- Project Explorer (Left):** Shows the project structure, including 'CadastroEE' and its sub-projects like 'CadastroEE-war', 'CadastroEE-ear', and 'CadastroEE-ejb'. It also lists various libraries and configuration files.
- Source Editor (Top):** Displays the 'web.xml' file, which contains the following configuration:


```

<?xml version="1.0" encoding="UTF-8"
<web-app version="4.0" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  <servlet-class>cadastroee.servlets.ServletProduto</servlet-class>
  <servlet-mapping>
    <servlet-name>ServletProduto</servlet-name>
    <url-pattern>/ServletProduto</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
</web-app>
      
```
- Output Window (Bottom):** Shows the execution output of the 'CadastroEE' application. It indicates that the application was built successfully and is running on 'http://localhost:8080/CadastroEE-war'. The output also shows the initial deployment and the completion of the initial distribution of 'CadastroEE'.
- Run Window (Bottom):** Displays the 'BUILD SUCCESSFUL' message, indicating that the application was built successfully in 12 seconds.
- SQL Editor (Top):** Shows the SQL script for creating the database schema. The script includes the following statements:

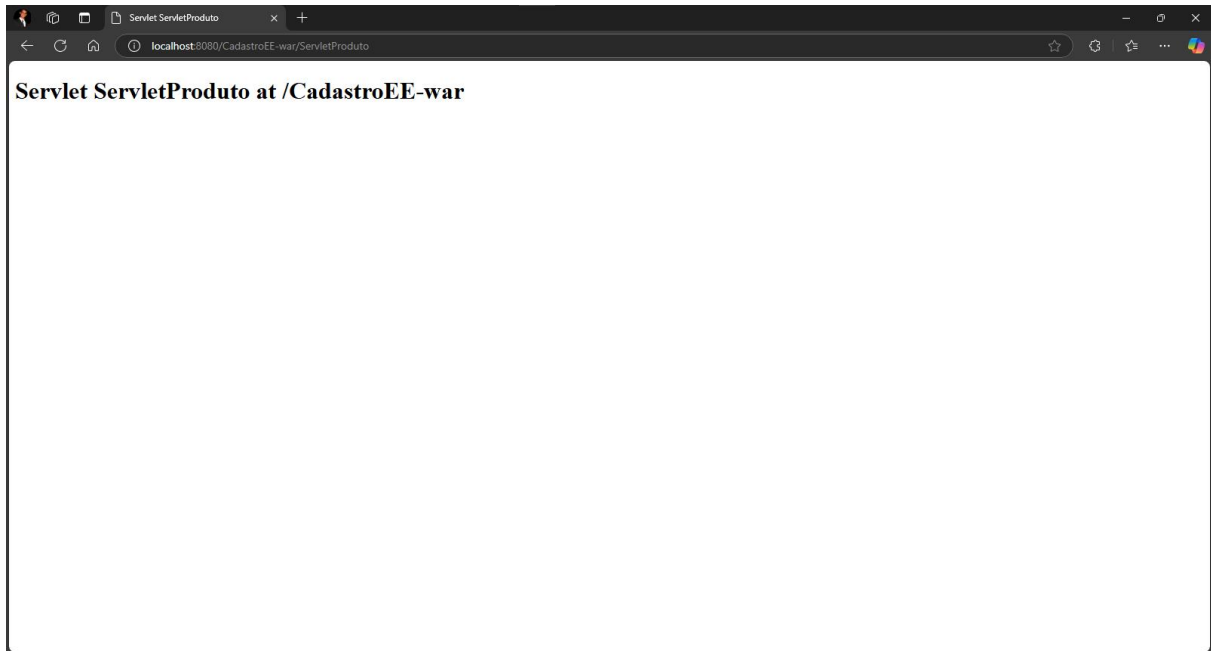

```

CREATE SEQUENCE sequencia
START WITH 1 -- Inicia a sequencia a partir de 1
INCREMENT BY 1 -- Incremento de 1 em 1
MINVALUE 1 -- Valor minimo permitido (1)
NO CYCLE; -- Não reinicia o valor

CREATE TABLE Pessoa (
  id INTEGER NOT NULL,
  nome VARCHAR(255),
  logradouro VARCHAR(255),
  cidade VARCHAR(255),
  estado VARCHAR(255),
  telefone VARCHAR(255),
  email VARCHAR(255),
  PRIMARY KEY (id));

CREATE TABLE Produto (
  idProduto INTEGER NOT NULL,
  nome TEXT,
  quantidade TEXT,
  preco_venda TEXT,
  PRIMARY KEY (idProduto));

CREATE TABLE Usuario (
  idUsuario INTEGER NOT NULL,
  login TEXT,
  senha TEXT);
      
```



3 ANÁLISE

Como é organizado um projeto corporativo no NetBeans?

No NetBeans, um projeto corporativo é estruturado de forma modular, o que permite que diferentes componentes sejam desenvolvidos e gerenciados de maneira eficiente. O projeto é separado em projeto principal, projeto EJB e projeto WER, onde:

Projeto Principal (ou Web):

Esse é o projeto que gerencia a interface com o usuário e os recursos do front-end.

Um projeto Web (projeto WAR - Web ARchive) que contém servlets, JSPs, recursos estáticos (como imagens e arquivos CSS/JS) e a configuração da aplicação web.

Projeto EJB (Enterprise JavaBeans):

O EJB contém a lógica de negócios e serviços do back-end.

Esse projeto terá um módulo JAR (Java ARchive). Ele gerencia a lógica de transações, segurança, persistência e pode interagir com bancos de dados.

Projeto WER (Web Services Enterprise/RESTful):

Esse projeto é onde você coloca Web Services (tanto SOAP quanto RESTful).

Um projeto WER é voltado para a exposição de APIs (REST ou SOAP) que podem ser consumidas por sistemas externos.

Ele pode estar incluído no projeto Web ou ser separado como um projeto independente (dependendo da arquitetura).

Esse projeto pode expor funcionalidades do EJB via Web Services, tornando-as acessíveis de forma distribuída.

Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?

JPA (Java Persistence API):

Função: Facilita a interação com o banco de dados, permitindo mapear objetos Java para tabelas de banco de dados e realizar operações de CRUD (Create, Read, Update, Delete).

Exemplo: Quando você precisa salvar ou recuperar dados de um banco, a JPA lida com a parte da persistência de forma simples e eficiente, usando anotações e frameworks como Hibernate.

EJB (Enterprise JavaBeans):

Função: Fornece uma camada de abstração para lógica de negócios distribuída e escalável. Oferece funcionalidades como gerenciamento de transações, segurança e concorrência, e pode ser acessado remotamente.

Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?

Com produtividade, O NetBeans fornece ferramentas automáticas, como geração de código, templates e integração com banco de dados, que tornam o desenvolvimento com JPA e EJB mais rápido e eficiente. Também com a facilidade de Uso: A IDE facilita tarefas complexas (como configurações e deploy), permitindo que o desenvolvedor se concentre mais na lógica do aplicativo e menos nas configurações técnicas.

O que são Servlets, e como o NetBeans oferece suporte à construção desse tipo de componentes em um projeto Web?

Servlets são componentes Java que processam requisições HTTP e geram respostas em aplicativos Web. Eles atuam no lado do servidor, sendo responsáveis por lidar com a lógica de negócios, interagir com bancos de dados e gerar dinamicamente o conteúdo a ser enviado ao cliente (navegador).

Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs?

A comunicação entre Servlets e Session Beans no pool de EJBs é feita através de injeção de dependência ou lookup. No Servlet, você pode usar a anotação @EJB para injetar uma instância do Session Bean diretamente. Já o Servlet pode fazer um lookup do EJB usando JNDI (Java Naming and Directory Interface) para localizar o Session Bean no servidor.