

Relatório de Atividades – Laboratórios de Sistemas Distribuídos

Disciplina: Sistemas Distribuídos

Professor: Rafael Braga

Aluno: André Alves de Freitas

Campus: UFC - Quixadá

1. Introdução

Durante as atividades práticas da disciplina de Sistemas Distribuídos, realizei a construção de dois laboratórios interligados:

- O primeiro consistiu no desenvolvimento de um WebService REST utilizando Spring Boot, simulando uma API de calculadora.

- O segundo envolveu a criação de um cliente em PHP para consumir os endpoints dessa API, utilizando interface web simples com HTML e cURL.

O objetivo geral foi compreender a comunicação entre sistemas heterogêneos (Java ↔ PHP) via WebService, reforçando conceitos de arquitetura cliente-servidor e protocolos HTTP.

2. Laboratório 1 – API REST com Spring Boot

2.1 Etapas Realizadas

1. Instalação das ferramentas:

- Instalei o Java 17 (OpenJDK) e o Maven no Ubuntu.

2. Criação do projeto com Maven:

- Utilizei o VSCode e a opção "New Maven Project" para gerar a estrutura do projeto.

3. Desenvolvimento da API:

- Implementei a classe Main.java com os métodos:

- /somar/{a}/{b}

- /subtrair/{a}/{b}

- /multiplicar/{a}/{b}

- /dividir/{a}/{b}

- Incluí tratamento de erro para divisão por zero.

4. Interface HTML:

- Criei um index.html básico com fetch() para consumir a API.

5. Execução:

- Rodei com mvn clean install e mvn spring-boot:run.

- A API respondeu corretamente via navegador em <http://localhost:8080/index.html>.

2.2 Dificuldades Encontradas

- Configuração do Java e Maven:

Tive dificuldade para verificar as versões e configurar corretamente o Maven. Resolvi seguindo tutoriais recomendados no roteiro.

- Estrutura do projeto Maven:

No início, cometi erros ao posicionar os arquivos nas pastas erradas. Precisei rever a estrutura padrão (src/main/java, static/, etc.).

- Porta 8080 ocupada:

3. Laboratório 2 – Cliente PHP para Consumo da API

3.1 Etapas Realizadas

1. Instalação do PHP e cURL:

- Instalei com sudo apt install php php-cli php-curl.

2. Criação do cliente:

- Montei a pasta cliente-php com os arquivos:

- index.php: interface HTML.
- calcular.php: envio da requisição à API com cURL.
- resultado.php: exibição do resultado.

3. Execução local:

- Rodei o servidor com `php -S localhost:8000`.
- Acessando via navegador, a calculadora funcionou conforme esperado.

3.2 Dificuldades Encontradas

- Requisição cURL com erro:

A extensão `php-curl` não estava habilitada por padrão. Resolvi com:
`sudo apt install php-curl`

- Falhas com números decimais:

Os campos `input type="number"` do HTML não aceitavam vírgula como separador decimal. Corrigi usando `floatval()` no PHP e padronizando o uso do ponto (.).

- Erro ao integrar com Spring Boot:

No início, os endpoints não respondiam ao PHP porque a API ainda não estava rodando corretamente. Apreendi a verificar se a porta 8080 estava aberta e se a API estava ativa.

4. Conclusão

A construção desses laboratórios foi desafiadora, principalmente por envolver tecnologias distintas (Java e PHP) e exigir configuração de ambiente. No entanto, foi uma experiência extremamente enriquecedora, pois consegui compreender na prática:

- Como expor serviços REST em Java com Spring Boot;
- Como consumir serviços REST com PHP;
- Como testar endpoints e tratar erros de integração.

Senti evolução tanto na parte de programação quanto na autonomia para configurar ferramentas e solucionar problemas típicos de ambientes distribuídos.

