



INSTITUTO FEDERAL

São Paulo

Câmpus Bragança Paulista

Tecnologia em Análise e Desenvolvimento de Sistemas

Desenvolvimento Web Back-end (BRADWBK)

Prof. Luiz Gustavo Diniz de Oliveira Vêras

Orientações para o Desenvolvimento de Projetos Web Back-end - Semestre 2025/1

Introdução

Este documento descreve as especificações acerca do projeto a ser desenvolvido como requisito da disciplina de Desenvolvimento Web Back-end (BRADWBK). O foco é a construção de uma **API REST** com servidor web integrado a banco de dados e a realização de comparações de desempenho em ambiente com e sem balanceamento de carga.

Objetivo do Trabalho

Desenvolver competências técnicas no desenvolvimento de APIs RESTful robustas, com banco de dados relacional e implantação em ambiente com balanceamento de carga. O projeto também visa desenvolver habilidades de teste funcional e de desempenho com ferramentas especializadas.

Dinâmica do Trabalho

O sistema deverá ser desenvolvido por grupos de **1 até 5 alunos**.

Cada grupo deverá:

- Escolher um domínio (tema da aplicação);
- Definir pelo menos 4 entidades e envolvê-las em pelos menos um **relacionamento Um-para-muitos** e um **Muitos-para-Muitos**;
- Modelar e implementar uma API REST seguindo especificações HTTP. A API modelada deve considerar operações de CRUD para cada uma das entidades;
- Implantar a API em ambiente com **NGINX** como balanceador de carga;
- Realizar testes de performance com **JMeter**, comparando os resultados com e sem o balanceamento de carga, de acordo com o que foi apresentado em aula;
- Realizar testes funcionais com ferramentas como **JMeter**, **Postman**, **Thunder Client**, **Insomnia**, entre outras.

Documento de Especificação da API

Cada grupo deverá elaborar um documento simples de especificação da API, contendo as seguintes informações:

- Tema da aplicação (se for o mesmo de outra disciplina do curso, pode repeti-la aqui);
- Integrantes do grupo (nome e prontuário);
- Quatro entidades e a descrição do relacionamento entre elas;
- Tabela de modelagem da API REST, conforme exemplo:

Verbo HTTP	Path	Body de Requisição (se houver)	Body de Retorno (se houver)	Status HTTP de sucesso	Status HTTP de erro

Obs: A Tabela anterior pode ser substituída por modelagem em ferramenta como Swagger ou similar.

Tecnologias

As seguintes tecnologias são preferíveis, mas **podem ser substituídas**, desde que mantenham a aderência aos requisitos de projeto definidos aqui e a arquitetura REST:

- **Back-end:** Java com Spring Boot e Spring Web;
- **Banco de Dados:** PostgreSQL, MySQL, SQLite ou outro banco relacional. Se preferir por utilizar o banco de dados H2, que funciona de forma similar ao SQLite, porém é para Java;
- **ORM:** JPA/Hibernate para operações com banco de dados relacional.;
- **Balanceamento de Carga:** Servidor NGINX;
- **Teste de Performance:** Apache JMeter;
- **Teste Funcional:** JMeter, Postman, Thunder Client, Insomnia ou similar;
- **Versionamento de Código:** Git (preferencialmente GitHub, GitLab ou Bitbucket).

Entregas

O grupo deverá entregar:

- Documento de especificação do domínio, entidades e API Web em PDF;
- Código-fonte da API REST com suporte a CRUD das entidades em repositório Git;
- Scripts de configuração do NGINX para balanceamento de carga em repositório Git;

- Relatório comparativo de testes de performance com e sem balanceamento, usando JMeter em PDF;
- Evidência dos testes funcionais com alguma ferramenta especializada. Podem ser geradas imagens JPEG ou PNG anexadas ao documento PDF do item anterior.

Data de Entrega de todos os documentos e Apresentação: 26/06/2025.

Critérios de Avaliação

Código do Critério	Descrição	Pontuação Máxima
CR1	Documento de especificação do domínio, entidades e da API Web	1,0
CR2	CRUD da entidade 1 pela API Web integrada ao Banco de Dados com ORM	1,0
CR3	CRUD da entidade 2 pela API Web integrada ao Banco de Dados com ORM	1,0
CR4	CRUD da entidade 3 pela API Web integrada ao Banco de Dados com ORM	1,0
CR5	CRUD da entidade 4 pela API Web integrada ao Banco de Dados com ORM	1,0
CR6	Implementação de 1 relacionamento Um-para-Muitos	1,0
CR7	Implementação de 1 relacionamento Muitos-para-Muitos	1,0
CR8	Configuração e funcionamento do balanceamento com NGINX	1,0
CR9	Relatório de testes de performance com JMeter	1,0
CR10	Testes funcionais documentados com Postman ou equivalente	1,0
Total da pontuação		10,0

Um ótimo desenvolvimento a todos!