"Quem ouve, esquece. Quem vê, lembra. Quem faz, aprende."
Provérbio chinês.

FIAP – Faculdade de Informática e Administração Paulista

Curso de Tecnologia em Análise em Desenvolvimento de Sistemas (TDS)

Professor: Dr. Marcel Stefan Wagner

Checkpoint 4

Parte 1 (API e *Deploy*) + Parte 2 (MVC e *Deploy*)

Parte 1 – Exercício de Revisão de API

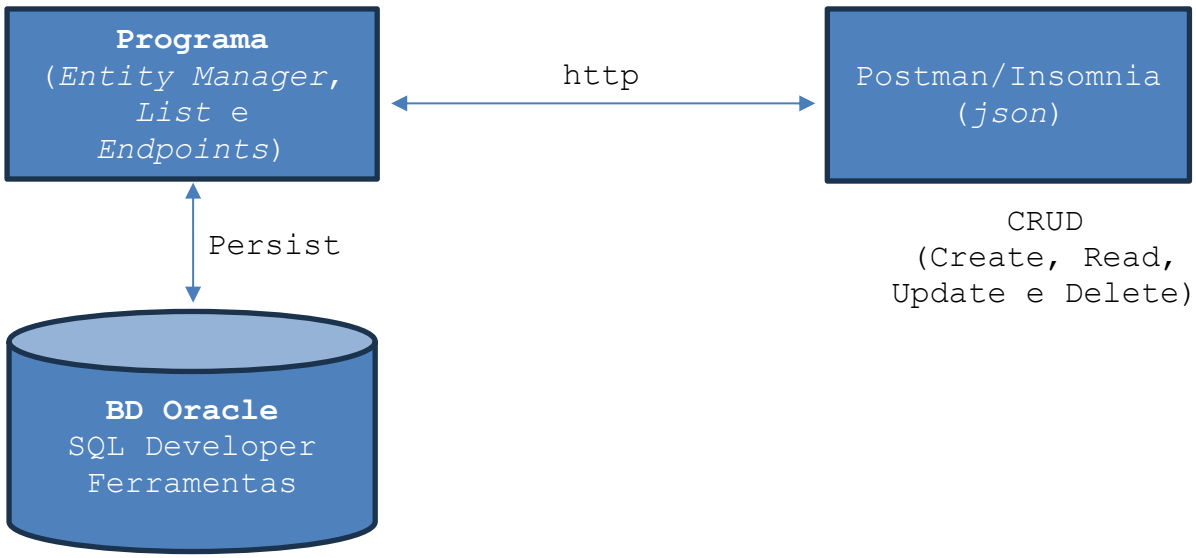
Spring com Persistência, Lombok, HATEOAS e Deploy

- A entrega deverá ser feita via **Teams** com o envio do arquivo diretamente ao professor (chat direto), **não** via chat da disciplina, informando a sua turma e curso;
- Deve-se entregar o Projeto em formato **.zip**, somente **um(a) integrante por grupo**, contendo:
    - Um arquivo **.txt** com o **nome e RM de todos(as) os(as) integrantes do grupo**. Neste mesmo arquivo, o grupo deve **fornecer o link do GitHub que possua um ReadMe contendo toda a descrição do Projeto** (incluindo imagens e explicações, principalmente do CRUD e com exemplos) e **indicar qual o IDE utilizado para elaboração do projeto** (IntelliJ, Eclipse ou NetBeans);
    - A **pasta toda do Projeto** com todas as pastas, subpastas e arquivos.
    - Um ***print* da tela com a configuração final do *Spring Initializr*** e respectivas dependências em **.jpg**, **.jpeg** ou **.png**.

Para tanto, desenvolva o seguinte aplicativo: Faça um Programa para uma empresa de **ferramentas e acessórios** (por exemplo: alicates, martelos, furadeiras, etc.) com base no *framework Spring Boot* configurado para o tipo *Maven* em linguagem Java, com as respectivas dependências, incluindo Lombok. Para tanto, faça os testes *Web* de *endpoints* via http usando o *software* Postman ou Insomnia, e contemplando todos os aspectos básicos *Create*, *Read*, *Update* e *Delete* de CRUD. Toda esta parte deve estar documentada no Relatório ou no *ReadMe*

do GitHub com *prints* de tela e respectivas explicações. Ao se consultar um determinado brinquedo via GET para o *endpoint* "/ferramentas" no Postman ou Insomnia, o programa deve consultar uma Tabela (por exemplo: TDS_TB_Ferramentas) no banco de dados ORACLE_FIAP do SQL Developer (com uma configuração básica em um arquivo persistence.xml da pasta META-INF ou via *application.properties*) para então, retornar ao Postman ou Insomnia o resultado da consulta com as informações do brinquedo solicitado, que deve ser mostrado no Postman ou Insomnia como resultado da consulta. Considerar:

- As seguintes colunas na Tabela do BD: Id, Nome, Tipo, Classificacao, Tamanho e Preco.
- No teste do Postman ou Insomnia deverá ser utilizado o endereço "*localhost*" e a porta **8081** do Servidor Tomcat.
- Para o POST, PUT e PATCH mostrar como fica a estrutura JSON, sugere-se enviar estes dados para uma lista no Programa desenvolvido, para então recuperar estas informações e depois enviar ao BD Oracle SQL Developer para o *Commit* com a inserção de dados no BD.
- Para o DELETE, deve-se realizar a exclusão do BD pelo ID.
- Deve-se utilizar obrigatoriamente o Lombok.
- Utilizar o padrão de retorno de informações HATEOAS (nível de maturidade 3 de projeto).
- Fique à vontade para criar os respectivos *endpoints*.
- Faça o *Deploy* em alguma plataforma que você queira e disponibilize o link de conexão via GitHub.



Parte 2 – *Spring Web*

Spring MVC e *Deploy*

- A entrega deverá ser feita via **Teams** com o envio do arquivo diretamente ao professor (chat direto), **não** via chat da disciplina, informando a sua turma e curso;
- Deve-se entregar, somente **um(a) integrante por grupo**, contendo:
    - Um arquivo **.txt** com o **nome e RM de todos(as) os(as) integrantes do grupo**. Neste mesmo arquivo, o grupo deve **fornecer o *link* do *GitHub* que possua um ReadMe contendo toda a descrição do Projeto** (incluindo imagens e explicações, principalmente do CRUD e com exemplos) e **indicar qual o IDE utilizado para elaboração do projeto** (IntelliJ, Eclipse ou NetBeans);
    - No mesmo .txt além do ***link* do *GitHub* do Projeto Spring MVC da parte II que seja separado da parte I**, **deve-se ter o *link* do *Deploy* e indicação de qual o sistema que foi utilizado para o *Deploy***.
    - Um ***print* da tela com a configuração final do *Spring Initializr*** e respectivas dependências em **.jpg**, **.jpeg** ou **.png**.

Para tanto, desenvolva o seguinte aplicativo: Dentro do mesmo tema da Parte 1, faça um Programa para uma empresa de **ferramentas e acessórios** (por exemplo: alicates, martelos, furadeiras, etc.) com base no *framework Spring Boot* configurado para o tipo *Maven* em linguagem Java, com as respectivas dependências, de preferência incluindo o *Lombok*. Para tanto, implemente uma interface *Web* e os respectivos *endpoints*, contemplando todos os aspectos básicos *Create*, *Read*, *Update* e *Delete* de CRUD, que devem aparecer na interface *Web* como *links* e/ou botões. Toda esta parte deve estar documentada no *ReadMe* do *GitHub* com *prints* de tela e respectivas explicações. Por exemplo, ao se consultar um determinado brinquedo, o programa deve consultar uma Tabela (por exemplo: TDS_MVC_TB_Ferramentas) no banco de dados usado na Parte I (com uma configuração básica em um arquivo persistence.xml da pasta META-INF ou via *application.properties*) para então, retornar o resultado da consulta com as informações do brinquedo solicitado na tela do navegador com o *Deploy* realizado. Com base nesse contexto, deve-se considerar:

- Utilize **obrigatoriamente** o *Thymeleaf*.
- As colunas na Tabela do BD ficam abertas para personalização conforme o grupo desejar.
- Para o CREATE, READ, UPDATE e DELETE, sugere-se utilizar o mesmo BD para o *Commit*.
- Pode-se utilizar o *Lombok* nesta parte também.
- Fique à vontade para criar os respectivos *endpoints*.
- **O *layout* do *front-end* será levado em consideração**.
- Faça o *Deploy* em alguma plataforma que você queira (por exemplo: *GitHub pages*, *Render* e *Fly.io*) e disponibilize o *link* de produção via *GitHub*.
- Faça um **vídeo de aproximadamente 5 minutos** mostrando as funcionalidades e *endpoint* com interface *Web* desenvolvida.

Tranquilo!

Bons estudos!

Atenção

- **Data de entrega do CP4 Parte II: 07/09/2025 (domingo) até 23h59.**
- **Entregas atrasadas serão consideradas como nota zero para a atividade.**