

O objetivo do projeto visa implementar um projeto AEM com o conhecimento de Java e AEM adquirido durante nosso programa.

## PROJETO AEM

1- Estruturar o projeto AEM com Sling Models, Sling Servlets, Service, DAO, Impl e tratamento de exceções, conforme visto nos módulos anteriores e práticos com AEM. Neste projeto, implementar API que tenha capacidade de consultar, inserir, atualizar e deletar dados através de uma requisição. Iremos criar uma API de Catalogo de produtos, onde devemos ter as seguintes operações básicas com persistência em banco de dados e estrutura:

Produto: id, nome, categoria, preço

Cliente: id, nome

Notas fiscais: numero, idproduto, idcliente, valor;

### GET –

permite consultar dados através de **parâmetros ou sem parâmetros**;

Implementar servlet com filtros, por id de produto, ordenar por preço mais baixo, categoria e por palavras chaves(produto que contém “X” palavra);

Implementar servlet que consulta as compras que um determinado cliente fez através da nota fiscal;

### POST –

Permite receber um payload com um **produto ou vários**;

Implementar cadastro de um **cliente ou vários** no mesmo payload;

DELETE – permite deletar **um ou vários produto/clientes**

PUT - permite atualizar um produto ou cliente;

Exemplificando, imagine uma URL: <http://localhost:4502/bin/projectexample/product?id=2>

Considerando essa URL, o servlet retornaria um JSON com o produto através do id que foi passado na requisição.

Essa mesma URL poderia ser uma consulta por id, inserir, alterar ou deletar um produto, por exemplo.



**Relatório**

*product/user/report*

Criar um Sling Servlet que retorna um PDF ou uma pagina HTML com o relatório de produtos comprados por um determinado cliente.

Espera-se que o Sling Servlet seja robusto suficiente para receber CRUD de registros, emita um relatório simples, implementação com tratamentos, boas práticas, **dentre outros**.

Espera-se que a aplicação tenha dentro da sua implementação pelo menos um filtro.

**Obrigatório:**

Seguir estrutura de projeto AEM conforme demonstrado;

Estruturar o padrão MVC;

Criar Service para separar a lógica do controlador;

Criar DAO;

Utilizar o MySQL como banco de dados;

Divisão de pacotes e boas práticas de POO;

Tratamento de exceções;

Retorno de JSON personalizado em casos de erros;

Enviar link do repositório git;

**Extra:**

Realizar alguma implementação de segurança (EXTRA);

**Não aceitaremos a entrega após o prazo.**

Prazo de entrega:

**17/08/2022 – 13h00**

Enviar link do github para os e-mails:

andre.lima@compasso.com.br; renan.acosta@compasso.com.br;

thauany.martins@compasso.com.br

