

EDITAL DO ENTREGÁVEL FINAL

1) O QUE FAZER:

- Desenvolver uma API RESTful (back-end) utilizando Springboot para **Gestão de Loja de Roupas**.
- Entidades:
 - Cliente: id, nome, cpf, idade e telefone.
 - Funcionário: id, nome, idade e matrícula.
 - Venda: id, endereço da entrega e valor total.
 - Produto: id, nome e valor.
- Relacionamentos:
 - Um cliente pode realizar várias compras, ou seja, pode estar vinculado a várias vendas, mas uma venda só pode estar associada a um cliente.
 - Um funcionário pode realizar várias vendas, mas uma venda só pode estar associada a um funcionário.
 - Uma venda pode ter vários produtos e um produto pode estar vinculado em várias vendas.
- Regras de negócio e validações:
 - A entidade principal é Venda. O sistema deve permitir salvar a venda com o cliente, o funcionário e os produtos em uma única requisição (cascade).
 - Todos os atributos das entidades são obrigatórios, exceto endereço de entrega (validation).
 - Além de métodos básicos de CRUD (findAll, findById, delete, save e update), implementar ao menos 3 filtros (findBy...), sendo 01 com JPQL e 02 com métodos automáticos para cada repository de cada uma das 4 entidades.
 - Todas as entidades devem possuir endpoints, então, haverá controller, service, repositon para cada entidade.
- **Se o grupo preferir fazer a solução de outro problema qualquer, também é possível, desde que atenda a todos os requisitos técnicos, que contenham no mínimo 4 entidades, relacionamentos 1-N, N-1 e N-N e todas as demais regras de negócio solicitadas.**

2) REQUISITOS QUE SERÃO AVALIADOS PELO PROFESSOR:

- Padrões organizacionais do Spring Boot: organização e nomenclatura dos packages classes; padronização de nomenclatura dos métodos; uso de annotations corretas nas classes.

- Persistência com Spring Data JPA: annotations para mapeamento de relacionamentos e atributos nas classes de entidade; validations nas classes de entidades; uso do JPA consultas customizadas com métodos automáticos e JPQL.
- Estrutura de requisições HTTP: endpoints padronizados com os métodos de CRUD; verbos HTTP corretos; JSON no body; request params, etc.
- Funcionamento das requisições pelo POSTMAN: consumo de todos os serviços da API sem erros.
- Solução do problema (interpretação): leiam atentamente e façam exatamente o que está sendo pedido no enunciado. Mesmo que funcione, se não estiver atendendo o que está sendo pedido, a nota poderá ser prejudicada.

Obs.: Dependências necessárias no POM: SPRING WEB, SPRING DATA JPA, SPRING VALIDATION, MYSQL CONECTOR J e LOMBOK.

3) FORMA DE ENTREGA E PRAZOS:

APRESENTAÇÕES:

- Mostrar e explicar a solução e o código de acordo com os requisitos do Item 2 deste documento.
- Rodar as requisições pelo POSTMAN.
- Todos os membros devem falar/explicar algo (a avaliação é individual).
- Tempo médio de apresentação: entre **10 e 15 minutos**.
- **Data: 26/08 e 27/08/2024** (a ordem será via sorteio).

ENTREGA APÓS APRESENTAÇÕES:

- Repositório Git: Apenas um do grupo envia a URL do repositório por meio da BlackBoard, desde que este liste os nomes de todos os membros do grupo.
- Máximo de membros: 03.
- Mínimo de membros: 01.
- **Data: 26/08/2024 às 19h (ENVIO DO LINK NA BLACK)**.
- **Data: 26/08/2024 às 23h59 (commits aceitos no repositório) (improrrogável)**.

4) ORIENTAÇÕES:

- **Dias de apresentações não são dias para tirar dúvidas ou corrigir bugs!** Portanto, iniciem desde já o desenvolvimento e tirem suas dúvidas até quinta-feira (22/08).
- **Todos deverão estar preparados para apresentar no dia 26/08**, pois haverá sorteio de ordem das apresentações.