

DRONESETA

Professor: Paulo Roberto Farah

Disciplina: 55DSW

Aluno: Gabriel Antônio Bertoldi Perini

DETALHES DO PROJETO

Além das instruções de como executar o projeto, este breve documento também conta com os principais requisitos que o projeto atende.

EXECUÇÃO

- **BACKEND**

Pode-se abrir tanto em qualquer IDE java, como eclipse ou IntelliJ e rodar o projeto dessa maneira.

Outra maneira seria a partir do terminal acessar a pasta: *Trabalho - Droneseta\backend\droneseta\target* e por fim executar o jar do projeto: *dronseta-0.0.1-SNAPSHOT.jar*

(Obs.: note que *droneseta* está escrito incorretamente, foi um erro no momento de gerar a aplicação, mas é assim mesmo).

O SGBD escolhido para armazenar os dados foi o *postgreSQL*;

Para configurar o acesso ao banco de dados é necessário alterar o arquivo ***application-properties*** e especificar o ***usuário, senha e o banco de dados***, o nome atual do banco seria ***db-droneseta***.

Outro ponto importante para notar é o *spring.jpa.hibernate.ddl-auto = create-drop*. Escolhi o método *create and drop* pois há um seeder no sistema que povoa vários produtos, e também realiza alguns pedidos automaticamente. Além disso um usuário (cliente) padrão é criado, com endereço e cartão de crédito. Um usuário admin também é criado. Podes encontrar todas as informações na classe ***DatabaseSeeder***

Para facilitar o login do professor, tanto na tela no website quanto na parte administrativa aqui estão os e-mails:

Cliente: paulo.farah@udesc.br senha: 12345

Admin: admin@admin.com senha: admin12345

- **FRONTEND**

Abrir um terminal nas pastas

Trabalho - Droneset\frontend\website:

npm install para instalar as dependências

npm run para iniciar a aplicação

Trabalho - Droneset\frontend\admin:

npm install para instalar as dependências

npm run para iniciar a aplicação

- **FUNCIONALIDADES & ESPECIFICAÇÕES:**

Essas informações abaixo apenas explicam partes que podem não estar explícitas ou de fácil entendimento no sistema ou que julguei que ficaram confusas durante a usabilidade, mas não consegui arrumar a tempo.

Drone: Todo o processo envolvendo, seja o bin packing para decidir quais pedidos e produto serão levados e a checagem de tempo em que o drone faria as entregas está no backend em uma **ScheduledTask**, para comodidade, coloquei o tempo das entregas para um minuto, tanto para confirmar a entrega quanto o horário em que é salvo que a entrega será registrada no banco de dados, mas é possível alterar o tempo em que o “drone” checa se há pedidos para entregar sem maiores problemas

Cadastro de usuário: No cadastro de usuário há 3 etapas, onde os dados de usuário, cartão e endereço devem ser preenchidos, os dois últimos dados são utilizados para fazer a compra também, não é pedido ao usuário os dados novamente.

Carrinho e efetuar compra: A compra é efetuada no carrinho de compras, é necessário estar logado para comprar, a fim de ter as informações de endereço e cartão.

Lista de produtos: Ao dar hover na lista de produtos, apenas o ícone de lupa é funcional, as demais funcionalidades como uma wishlist não consegui implementar... mas ficaram os ícones pois achei que ficaria meio vazio sem eles... Além disso cada produto tem um tamanho específico, o que gerou vários produtos iguais na lista de produtos. Infelizmente só percebi muito tarde e prossegui desta maneira.