**Droneseta**

O projeto se resume a um e-commerce de venda de camisetas em parceria com a empresa Alemão Drones. O projeto pode ser dividido, para uma mlehor explicação, em duas partes: back-end e front-end. Abaixo será explicado a implementação de cada uma das partes.

**Beck-end:** Para trabalhar do lado do servidor, foi construída uma API em Java Rest que manipula todo o trafego de informações no banco de dados. Inicialmente, o back-end conta com quatro classes de entidade: Endereco, Pedido, Produto e Usuario. Essas classes são necessárias para o bom andamento da aplicação. Porém, como há uma relação “muitos-para-muitos” entre Pedido e Produto, a própria API, automaticamente, criará mais uma tabela que irá lidar com essa comunicação entre estas classes de entidade. Para a escrita e leitura dos dados, três classes Web Services foram criadas: *PedidoFacadeREST*, que trabalhará com os pedidos, ProdutoFacadeREST, que irá lidar com os produtos e UsuarioFacadeREST que se encarregará da parte dos usuários.

**Sobre o servidor:** O servidor utilizado no projeto foi o *GlassFish* na sua versão 4.0.

**Sobre o banco de dados**: O Banco de Dados utilizado na manipulação dos dados do projeto, foi o *PostgreSQL*, em sua versão 9.6.

**Front-end:** Para a visualização das informações e navegação entre o projeto, o sistema conta com a implementação do React. O React trabalha com a construção da interface por meio de arquivos JavaScript e dessa forma, insere eles na interface dentro de um único arquivo HTML, ou seja, sem atualizar a página. Foi utilizado o framework Bootstrap para a estilização dos elementos HTML no projeto. Cada arquivo JavaScript é uma tela que será posta na tela principal do sistema, logo, por conta disso, não é necessário a atualização da página para alterações visuais no projeto. Também há, em meio ao arquivo JavaScript, as requisições POST e GET para o envio e coleta de informações no servidor. Mas como isso é feito? O JS possui um método de envio de requisições chamado **fetch()**, nele, você passa o endereço do seu servidor, o método de requisição e o tipo de dado que será enviado para o servidor, que no caso do projeto, será um JSON. Para