

# ProSTOCK

Projeto de Gerenciamento de Estoque  
Por Lucas Andrade de Oliveira Reis

## 1 DECISÕES ARQUITETURAIS

Ao longo do seu desenvolvimento, o projeto foi subdividido em camadas de modo que as responsabilidades fossem melhor distribuídas pelo sistema, permitindo assim uma melhor organização e potencial de escalabilidade.

- **ProStock.Domain:** Camada responsável por armazenar as entidades e regras de negócio.
  - O sistema possui apenas uma entidade (a de Produto) porém, caso o sistema necessite ser escalado, o desenvolvimento será um pouco menos oneroso.
- **ProStock.Repository:** Camada responsável pela persistência no dados no banco de dados
  - Foi utilizado o Microsoft.EntityFrameworkCore, pela facilidade do uso da ferramenta e pela fluência do desenvolvedor na mesma;
  - Foi utilizado Sqlite como banco de dados por ser simples e por não haver necessidade de robustez para este projeto.
- **ProStock.WebAPI:** Responsável pela disponibilização das APIs (via controllers);
- **prostock-app:** Diretório onde foram armazenados os arquivos do frontend.
  - Construído em angular;
  - Utiliza principalmente o bootstrap e alguns componentes do ngx-bootstrap, para construção de interfaces simples e visualmente agradáveis;
  - Modelos e serviços isolados para permitir reusabilidade.

## 2 POSSÍVEIS MELHORIAS OU ALTERAÇÕES

- Adição de arquivos de tradução para strings do sistema;
- Troca do banco de dados para um mais robusto;
- Criação de novos menus e rotas;
- Necessita que haja tratativa mais restritiva no Startup.cs, o qual hoje permite requisição cruzada (CORS) proveniente de qualquer origem, método ou header do protocolo http;
- Uso de ferramentas mais sofisticadas de estilização (não apenas bootstrap);
- Implementação de paginação na lista de produtos para otimizar o carregamento em caso de muitos produtos;