

# Controle de estoque

## Descrição

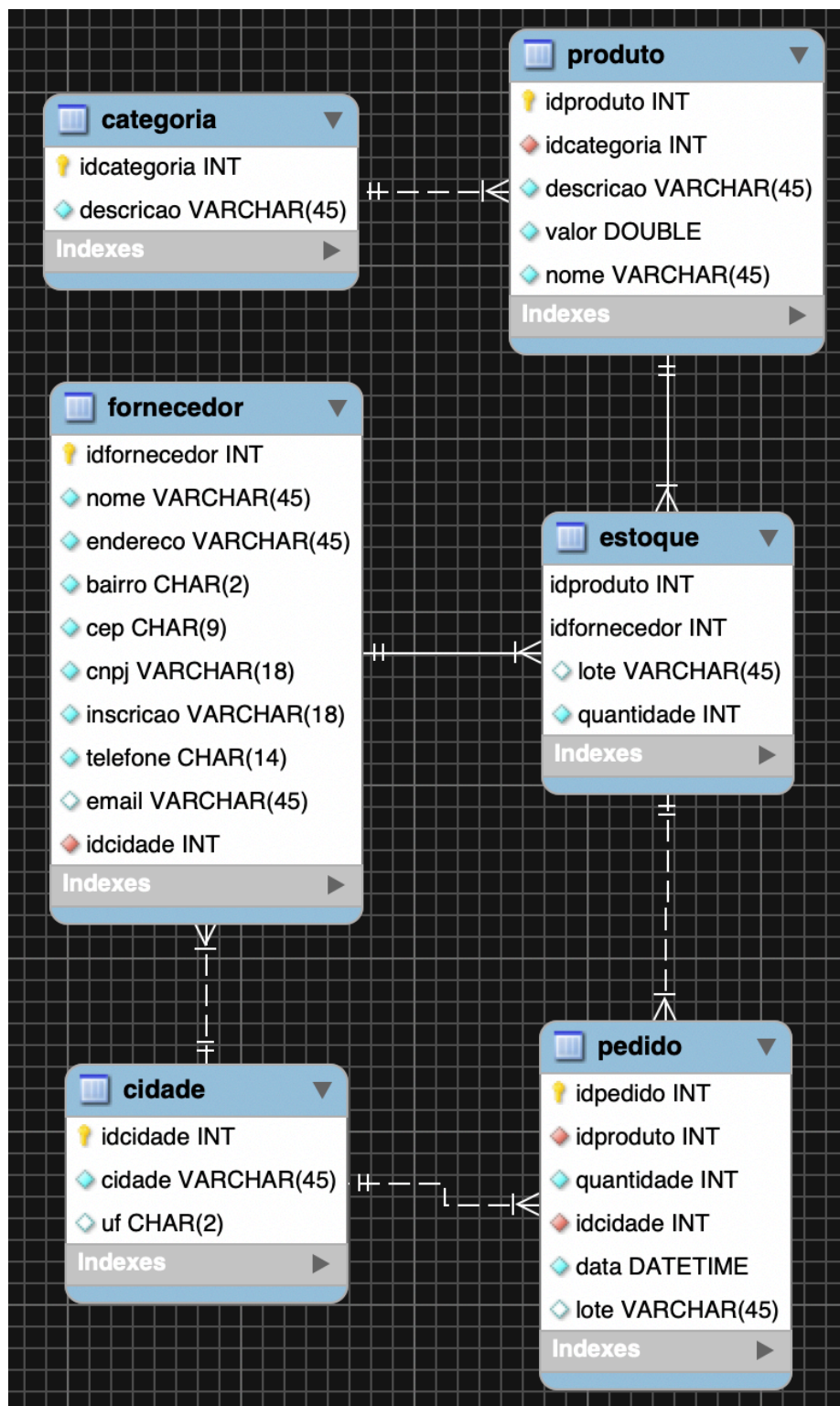
A implementação inicial proposta, baseia-se na premissa do usuário apenas adicionar ou excluir produtos, alterar nome, valor e quantidade.

As camadas de desenvolvimento permitem a expansão do projeto para um controle de estoque completo. Modificações nos controllers e models do backend, e na parte de acesso aos serviços do frontend permitem alterar as funcionalidades da aplicação sem retrabalho.

Para o controle de estoque completo, foi proposta a seguinte modelagem a seguir. A mesma contempla desde a entrada do produto no estoque até sua saída.

- **Categoria**  
Cada produto possuirá sua própria categoria, para que a utilização de filtros e relatórios seja possível.
- **Fornecedor**  
Um mesmo produto poderá ser recebido de diversos fornecedores, gerando um estoque único para o mesmo.
- **Produto**  
Cada produto possuirá um valor, descrição e uma categoria.
- **Estoque**  
Controlará a quantidade de cada produto por fornecedor. Isto também facilita a saída do produto no caso de entrega, que poderá ser entregue pelo fornecedor mais próximo de quem fez o pedido.
- **Pedido**  
Se refere ao pedido do produto, quantidade, fornecedor e data.

## Modelo de dados



## Configuração da stack

O banco de dados utilizado foi MySQL com a configuração padrão.

Foi utilizado o Visual Studio Code com a extensão Spring boot para criar a parte REST, utilizando Spring JPA.

Para a utilização na aplicação foi utilizada apenas uma tabela (dti.produto), mas para o sistema completo está sendo considerado o diagrama acima.

A connection string está em resources/application.properties.

A aplicação REST está no endereço <http://localhost:8080/service/produto> (DemoApplication.java) e a aplicação frontend no endereço <http://localhost:4200/home> (app-site).

A pasta app-site/node\_modules não foi incluído no GIT. (utilizei o comando npm i npm-install-all para atualizar as dependências).