Tayná Crisllen José De Santana
Matrícula: 201905556
Arquitetura de Software - 2021/1 (ES)
Prof. Guilherme Silva Marques

COMÉRCIO BRASIL

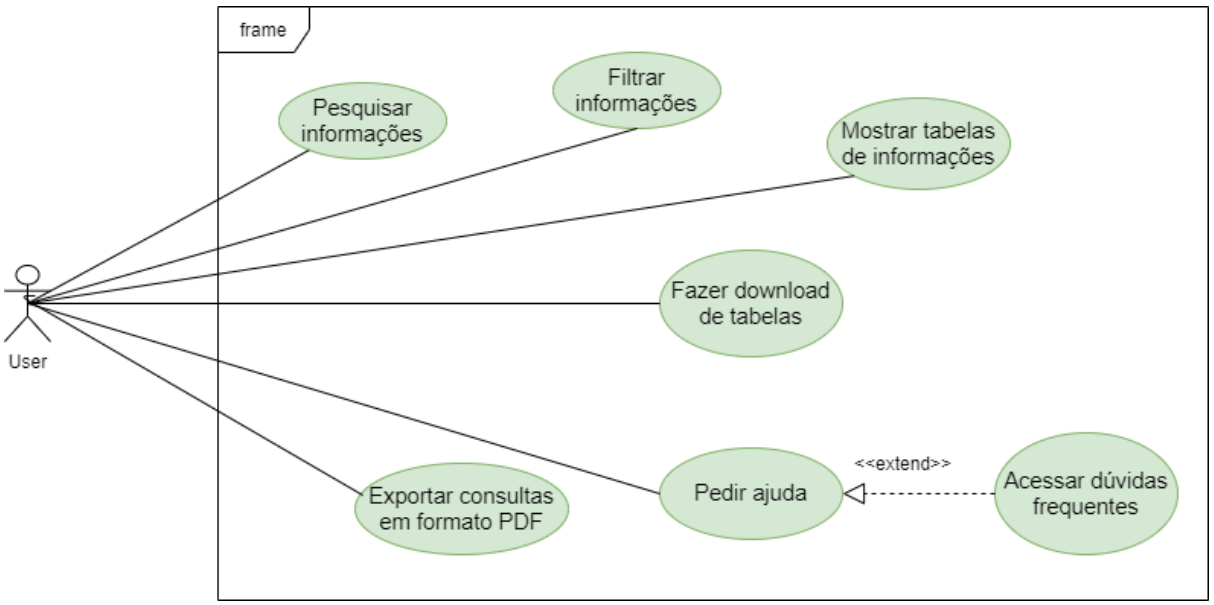
Escopo do projeto

Construção de um sistema Web que auxilia na visualização de dados referentes ao comércio internacional de produtos do Brasil. As funcionalidades do sistema Comércio Brasil abrangem as seguintes capacidades:

- O sistema deverá ser intuitivo ao usuário;
- Ter um API de pesquisa de fácil uso;
- Desenvolvimento de filtros de pesquisa;
- Disponibilizar para download tabelas de resultados de dados, tais dados serão fornecidos pelo banco de dados do Ministério da Indústria.

Diagrama de casos de uso

- Pesquisar informações ;
- Filtrar informações;
- Mostrar tabelas de informações;
- Fazer download de tabelas ;
- Pedir ajuda;
- Acessar dúvidas frequentes;
- Exportar consultas em formato PDF.



Definição Arquitetural

1. Tecnologias que serão utilizada

Através de discussões com os clientes, foram selecionadas as tecnologias a serem utilizadas .

1.1 GraphQL

O GraphQL foi desenvolvido para tornar as APIs mais rápidas, flexíveis e intuitivas para os desenvolvedores.

Alguns pontos positivos:

- Permite aos desenvolvedores construir solicitações que extraem os dados de várias fontes em uma única chamada de API.
- Proporciona aos profissionais responsáveis pela manutenção das APIs flexibilidade para adicionar ou preterir campos, sem afetar as consultas existentes.
- Os desenvolvedores podem criar APIs com o método que quiserem, pois a especificação do GraphQL assegura que elas funcionem de maneira previsível para os clientes.

1.2 Python (Django)

Django é um framework web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção. O Django cuida de grande parte do trabalho de

desenvolvimento web. É gratuito e de código aberto, tem uma comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago.

Django será responsável pelo back-end do projeto pois apresenta uma estrutura base para projetos, sendo a maior comunidade e tendo uma excelente aceitação sobre os frameworks web Pytthon, e ainda ajuda no aprendizado da equipe.

1.3 Angular

O Angular é um framework JavaScript que simplifica não apenas a construção da interface de usuário, mas também o desenvolvimento de aplicações client-side diferenciadas, sejam elas para a web, mobile ou desktop.

Como é um framework completo e seus componentes estão disponíveis para várias necessidades de aplicação , o Angular vai ser utilizado como front-end do projeto.

1.4 SQL Server

O SQL Server é um sistema gerenciador de Banco de dados relacional (SGBD).

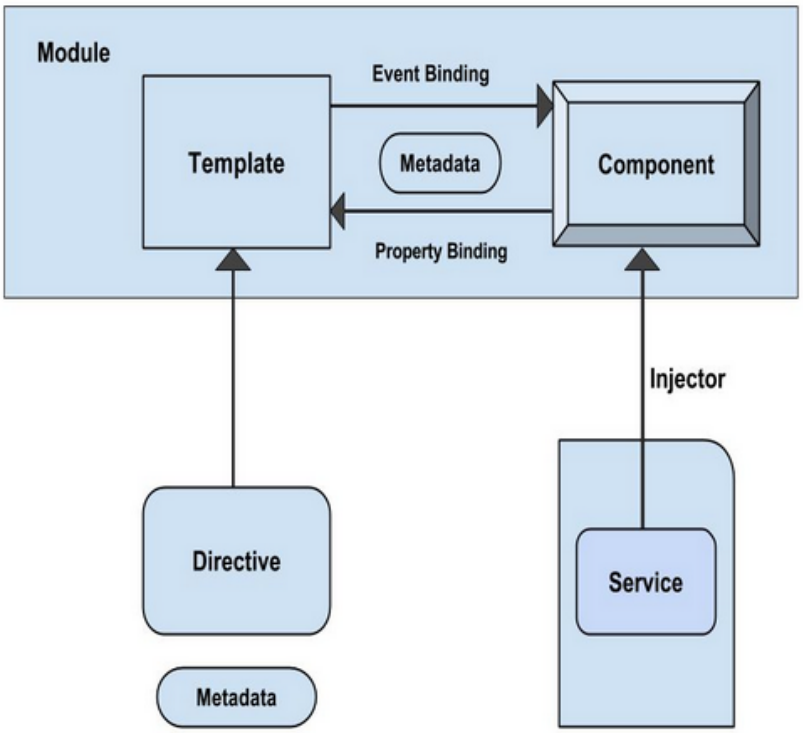
Como um Banco de dados, é um produto de software cuja principal função é a de armazenar e recuperar dados solicitados por outras aplicações de software, seja aqueles no mesmo computador ou aqueles em execução em outro computador através de uma rede (incluindo a Internet).Ele permite a criação de tabelas relacionadas, evitando a necessidade de armazenar dados redundantes em vários locais dentro de um banco de dados.

O SQL será utilizado pelos desenvolvedores, pois ele tem muitas formas para proteger e gerenciar os dados do projeto.

2. Visão da arquitetura

O Comércio Brasil será uma aplicação web desenvolvida a partir do framework Django, escrito em Python, em conjunto com o framework Angular, que será utilizado no desenvolvimento do front-end, sua linguagem de consulta é a bases de dados GraphQL, que facilitará os meios de comunicação com o banco de dados, pois tem uma grande capacidade como linguagem de consulta dentro do contexto de Data Science.

O Angular possui uma arquitetura orientada a componentes, como no diagrama:



Cada componente constrói através de Templates em HTML uma View, na qual é projetado o conteúdo visto pelo usuário, o Event e Propery Biding unem os elementos do Template nas funções do componente, o primeiro tipo conectando eventos e o segundo propriedades. Os serviços por sua vez, que são inseridos através das Dependency Injections , são tarefas que fogem do aspecto da interação entre Templates e Views, as Directives possuem outras instruções para a renderização das Templates em Views, podendo ser estruturais ou de atributo. Tudo isso forma um Módulo, que ao se juntar com outros módulos constrói a aplicação em Angular.

O Django, que será utilizado no back-end, por sua vez segue o padrão MVC de perto, no entanto, ele usa sua própria lógica na implementação. Como o *"Controller"* é manipulado pelo próprio framework e a maior parte do entusiasmo no Django acontece em modelos, templates e views, o Django é frequentemente chamado de framework da MTV. No padrão de desenvolvimento da MVT:

Model, a camada de acesso a dados. Essa camada contém tudo e qualquer coisa sobre os dados: como acessá-lo, como validá-lo, quais comportamentos ele possui e as relações entre os dados.

View, a camada de lógica de negócios. Essa camada contém a lógica que acessa o modelo e requisita a respectiva resposta. Você pode pensar nisso como a ponte entre modelos e templates.

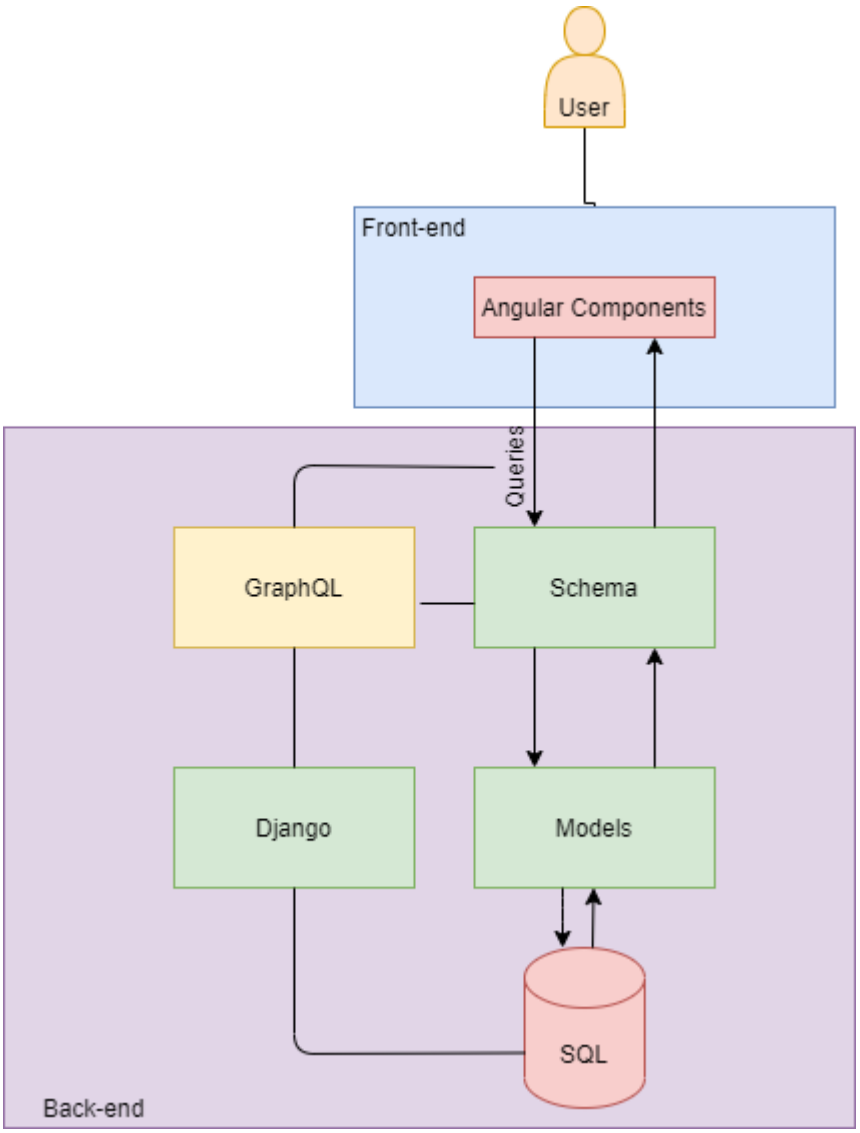
Template, a camada de apresentação. Essa camada contém decisões relacionadas à apresentação: como algo deve ser exibido em uma página da Web ou outro tipo de documento.

Desta forma, a *view* do Django é mais parecida com o *controller* no MVC, e a *view* do MVC é na verdade um *Template* no Django.

Logo o sistema não será desenvolvido utilizando arquitetura MVT do Django de forma pura, mas pela adaptação utilizada no Angular, dando ao projeto uma nova face arquitetural, o Django continuará fazendo seu papel de classes de domínios e interface com o banco de dados, o Angular estará na construção front-end, no lugar da função que seria feita pelas templates do Django. O GraphQL, pelo Django através da biblioteca Graphene, será a linguagem e ferramenta utilizada para realizar as consultas ao banco, ficando como interface entre o back-end e o front-end.

O usuário fará requisições por meio do front-end em Angular, que enviará uma ordem de pesquisa para a API através de um query parametrizado. Pelos métodos de pesquisa e filtragem definidos no Schema e pela utilização dos modelos de dados definidos nas models, será feita com o GraphQL uma consulta ao banco de dados . Após isso, o front-end receberá em formato JSON uma resposta com os resultados da pesquisa, que serão enfim mostrados ao usuário pelos Components do Angular.

Veja o diagrama abaixo:



3. Referências

O que é GraphQL? (redhat.com)
Introdução ao Django - Aprendendo desenvolvimento web | MDN (mozilla.org)
Guia Completo de Angular: Primeiros Passos no Angular (devmedia.com.br)
Microsoft SQL Server – Wikipédia, a enciclopédia livre (wikipedia.org)
Documento de Arquitetura · Comex Stat (fga-eps-mds.github.io)