

Tecnologias de programação web
2023/2024

RECEITAS

Mariana Perna, 108067
Rafaela Dias, 108782
Vânia Moraes, 102383

Índice:

Links e Utilizadores disponíveis

Introdução ao Conceito

Estrutura do Trabalho

Funcionalidades da Aplicação

Base de Dados

Deploy dos Ambientes

Referências

Links e Utilizadores disponíveis

O trabalho desenvolvido encontra-se disponível nos seguintes links:

1. Client-Side - proj2tpw.pythonanywhere.com
2. Server-Side - para correr deve entrar no repositório /delicious e fazer o comando “npm install”, de seguida correr “ng build” e “ng serve”.

Basta aceder à aplicação no client-side e efetuar o login para começar a explorar. De forma a poder explorar e utilizar o nosso site, preparamos alguns utilizadores:

1. Utilizador normal

- a. username - vania
- b. password - vanines05

2. Admin

- a. username - mariana
- b. password - 1234

As informações sobre os utilizadores e o deploy serão explicadas de seguida.

Introdução ao Conceito

O conceito por trás deste projeto é proporcionar aos utilizadores uma plataforma dedicada ao compartilhamento de receitas culinárias, com o intuito de facilitar a criação, partilha e descoberta de novos pratos. O projeto visa enriquecer a experiência culinária, tornando o processo de cozinhar mais acessível e exploratório.

Ao longo do desenvolvimento, optamos por adotar uma abordagem que separa claramente dois ambientes distintos e intuitivos. O primeiro ambiente concentra-se no armazenamento, tratamento e pesquisa de dados no lado do servidor (server-side). Utilizamos o Django REST Framework (DRF) para criar elementos fundamentais no back-end, incluindo métodos de autenticação, validação, pesquisa e roteamento.

No segundo ambiente, focamo-nos na interação do utilizador e na apresentação dos dados armazenados, utilizando tecnologias client-side. Implementamos o Angular Framework, baseado em TypeScript (TS), para desenvolver interfaces de utilizador interativas e exploráveis. Além disso, para simplificar e agilizar o desenvolvimento da interface, optamos pelo uso do Tailwind CSS. Este framework de estilo utilitário oferece uma abordagem altamente configurável, permitindo a criação de designs responsivos e estilizações consistentes. A integração do Tailwind CSS no lado do cliente proporciona uma manipulação eficiente e dinâmica dos dados apresentados aos utilizadores, aproveitando as classes utilitárias para uma estilização coesa e padronizada.

Este projeto visa oferecer uma experiência integrada e intuitiva, combinando tecnologias avançadas para criar uma plataforma robusta e atraente para os entusiastas da culinária.

Estrutura do Trabalho

A estrutura da aplicação foi concebida com uma clara separação em dois ambientes que se comunicam através de métodos HTTP Request e Response.

No âmbito do back-end, utilizamos diversas tecnologias suportadas pelo Django REST Framework (DRF). Especificamente, empregamos a estrutura base de um projeto Django e adicionamos mecanismos para atender às necessidades específicas do nosso projeto. Criamos nove modelos distintos, representando os diferentes elementos do sistema, e implementamos Serializers para validar e facilitar o acesso aos objetos armazenados. Estes Serializers desempenham um papel crucial em todas as interações com os objetos do sistema, incluindo operações de pesquisa, criação, alteração e eliminação.

Durante o desenvolvimento em Django, observamos atentamente a criação de endpoints da API, estabelecendo as principais rotas para interações específicas. Esses endpoints são essenciais para a integração e comunicação eficaz entre o back-end e o front-end da aplicação. Ao longo do processo, garantimos que cada modelo é devidamente representado e interligado através das Serializers, promovendo uma coesão estrutural e facilitando a manipulação de dados.

A seguinte lista apresenta uma visão geral dos endpoints da API desenvolvidos no contexto do Django REST Framework:

```

path('api-auth/', include('rest_framework.urls')),

path('ingredientes/', views.ingredientes),
path('criarIngrediente/', views.create_ingredient),
path('ingredientes/<int:id>', views.get_ingredient_detail),
path('update_ingredient/<int:id>', views.update_ingredient),
path('delete_ingredient/<int:id>', views.delete_ingredient),

path('receitas/', views.receitas),
path('criarReceita/', views.create_receita),
path('detail_receita/<int:id>', views.receita_detail),
path('update_receita/<int:id>', views.update_receita),
path('delete_receita/<int:id>', views.delete_receita),
path('receitas_filtradas/', views.filtrar_receitas),

path('categorias/', views.categorias),
path('criarCategoria/', views.create_categoria),
path('categorias/<int:pk>', views.categoria_detail),
path('categorias/<int:pk>', views.edit_categoria),
path('categorias/<int:pk>', views.delete_categoria),

path('categoria-ingredientes/', views.all_categoria_ingredientes),
path('create_categoria_ingredient/', views.create_categoria_ingredient),
path('categoria-ingredientes/<int:pk>', views.categoria_ingredient_detail),
path('edit_categoria-ingredientes/<int:pk>', views.edit_categoria_ingredient),
path('delete_categoria-ingredientes/<int:pk>', views.delete_categoria_ingredient),

path('receita_ingredientes/', views.receita_ingredientes),
path('create_receita_ingredientes/', views.create_receita_ingredientes),
path('detail_receita_ingredientes/<int:pk>', views.detail_receita_ingredientes),
path('update_receita_ingredientes/<int:pk>', views.update_receita_ingredientes),
path('delete_receita_ingredientes/<int:pk>', views.delete_receita_ingredientes),

path('avaliacoes/', views.avaliacoes),
path('criarAvaliacao/', views.create_avaliacoes),
path('detail_avaliacoes/<int:pk>', views.avaliacao_detail),
path('update_avaliacoes/<int:pk>', views.update_avaliacao),
path('remove_avaliacoes/<int:pk>', views.delete_avaliacao),

path('frigorificos/', views.frigorificos),
path('add_frigorifico/', views.create_frigorifico),
path('detail_frigorifico/<int:pk>', views.frigorifico_detail),
path('remove_frigorifico/<int:pk>', views.delete_frigorifico),

path('favoritos/', views.favoritos),
path('add_favoritos/', views.create_favoritos),
path('detail_favoritos/<int:pk>', views.favoritos_detail),
path('remove_favoritos/<int:pk>', views.delete_favoritos),

path('listas_compras/', views.listas_compras),
path('add_lista_compras/', views.create_listas_compras),
path('listas_compras/<int:pk>', views.lista_compras_detail),
path('remove_lista_compras/<int:pk>', views.delete_lista_compras),

```

Esta estruturação no back-end proporciona uma base sólida para a manipulação e gestão eficiente dos dados, garantindo consistência e segurança nas interações do utilizador.

Para o front-end da nossa aplicação usamos Angular Framework, criando um total de 16 componentes, muitos deles encapsulados dentro de outros componentes, que são invocados a partir de chamadas para qualquer das routes (RouterModule). De forma a salvar o estado atual da utilização recorremos a EventEmitters, com os quais guardamos a autenticação e autorização do utilizador atual. Para facilitar a troca de dados com a API, desenvolvemos web services que invocam qualquer um dos base models e tratam de efetuar o HTTP Request.

Em termos de componentes Angular, em todas as páginas existem 2 grandes componentes, a barra de navegação, o componente principal da página. A barra de navegação faculto o acesso aos routes disponíveis para cada utilizador. As páginas da aplicação serão explicadas nas funcionalidades do sistema.

Funcionalidades da aplicação

Utilizador Não Autenticado

Ao explorar a aplicação pela primeira vez, o utilizador encontra-se numa experiência não autenticada. Durante este período inicial e até à realização do registo, o utilizador tem a liberdade de navegar pela página principal (Home) onde são mostradas todas as receitas e explorar páginas adicionais com conteúdo informativo, como About . Além disso, na barra de navegação presente em todas as páginas, são visíveis os botões de Login e SignUp.

O botão de Login oferece a possibilidade ao utilizador de autenticar-se através de uma conta já existente no sistema. Contudo, caso o utilizador ainda não possua uma conta na aplicação, pode recorrer ao SignUp para efetuar o registo. Para um registo bem-sucedido, são necessários um nome, uma conta de email única (que não esteja associada a nenhuma conta existente) e uma palavra-passe segura. Após o SignUp bem-sucedido, o utilizador é redirecionado para a página de Login, onde pode autenticar-se com as credenciais recém-criadas.

Utilizador autenticado

Numa experiência autenticada na nossa aplicação, um utilizador normal desfruta de uma série de funcionalidades que tornam a sua interação mais personalizada e envolvente. Este utilizador tem o poder de criar receitas exclusivas, com a liberdade de alterar ou apagar aquelas que adicionou previamente. Além disso, pode construir e gerir a sua lista de compras e frigorífico, garantindo um controlo eficaz sobre os ingredientes disponíveis.

A funcionalidade de Receitas Favoritas permite ao utilizador autenticado manter uma coleção personalizada das suas receitas preferidas, com a capacidade de executar operações CRUD (Criar, Ler, Atualizar, Apagar) sobre elas. Para enriquecer a comunidade, o utilizador pode partilhar a sua opinião através de comentários nas receitas, com a flexibilidade de apagar ou atualizar os seus comentários conforme necessário.

Além destas capacidades exclusivas, o utilizador autenticado mantém acesso a todas as funcionalidades disponíveis para utilizadores não autenticados, proporcionando uma experiência completa e abrangente na nossa aplicação.

Admin

O administrador, detentor de privilégios especiais, tem a capacidade de realizar ações exclusivas que vão além das permissões dos utilizadores comuns. Dentre essas capacidades destacam-se a exclusão de qualquer receita, comentário ou elemento presente nas listas de frigorífico, compras ou favoritos.

Adicionalmente, o administrador possui acesso a uma página dedicada onde pode gerenciar dinamicamente a estrutura da aplicação. Nessa área privilegiada, é possível adicionar novas categorias, ingredientes, categorias de ingredientes ou até mesmo remover qualquer elemento que se torne desnecessário para a otimização do sistema. Este controle refinado permite ao administrador moldar e aprimorar a experiência dos utilizadores de forma eficiente.

No entanto, tivemos problemas na implementação do update de um receita e do registro de um novo utilizador. A lista de compras não tivemos tempo de implementar no frontend.

Base de Dados

Para o armazenamento dos dados, foi feito o recurso a uma base de dados SQLite, diretamente ligada ao Django, que facilita a recolha e o fornecimento de dados. Para a criação da base de dados, recorreu-se a migrações de tabelas/modelos desenvolvidos em Django e executadas pelo terminal com recurso a um ficheiro Bash, criado para esse mesmo efeito.

Deploy dos Ambientes

O ambiente do server encontra-se online, utilizamos PythonAnywhere, assim como aconselhado pelo professor e para o client-side tentamos utilizar Firebase, pois trata-se de um sistema de hospedagem que oferece inúmeras vantagens/funcionalidades para quem pretende dar deploy da sua Angular Application, no entanto, não conseguimos. Antes também experimentamos dar deploy no heroku mas também não tivemos sucesso.

Referências

Recorremos aos materiais disponibilizados pelo professor na página da cadeira, assim como sites com a documentação das tecnologias usadas:

- <https://www.django-rest-framework.org/>
- <https://angular.io/docs>
- <https://material.angular.io/>