App Install & Go

Roberto Filipe Manso Barreto
(nrº 21123, regime diurno)

Orientação de
Luís Gonzaga Martins Ferreira

Licenciatura em Engenharia em Sistemas Informáticos
Escola Superior de Tecnologia
Instituto Politécnico do Cávado e do Ave

Identificação do aluno

Roberto Filipe Manso Barreto
Aluno número 21123, regime diurno
Licenciatura em Engenharia em Sistemas Informáticos

Orientação

Luís Gonzaga Martins Ferreira

Informação sobre o Estágio

Motorline Eletrocelos S.A
Travessa do Sobreiro, 29 Rio Côvo (Sta. Eugénia) 4755-474 Barcelos
Eng. Helder Remelhe

Resumo

Resumo do trabalho realizado. Deve ser sucinto, e cobrir todo o relatório: uma introdução ao problema que se pretendeu resolver, um pequeno resumo da abordagem realizada, e algumas conclusões do trabalho atingido.

Poderão ser criados vários parágrafos, até para que cada um corresponda às três fases de introdução, desenvolvimento e conclusão.

Não é relevante colocar no resumo o local de estágio ou a referência ao curso. Essa informação já consta da capa.

Abstract

This is the translation of the previous text. It should say the exact same thing. Please do not use directly Google Translator.

Agradecimentos

[A secção de agradecimentos é a parte pessoal do documento, e o único sítio onde o aluno pode escrever de forma menos formal, usando o tipo de linguagem que lhe parecer adequado para as pessoas a quem agradece.]

Conteúdo

Lista de Figuras

Lista de Tabelas

1. Arquitetura de sistema

Na Figura 1.1 é possível visualizar a arquitetura do sistema que indica os principais componentes deste software. Entre estes componentes é possível visualizar a aplicação frontend, onde esta realiza pedidos a uma aplicação backend e espera respostas. A aplicação backend é composta de uma api rest que receberá os pedidos e responderá aos mesmos, este backend é composto também por uma base de dados a qual vai receber queries e devolver dados para a aplicação api rest.

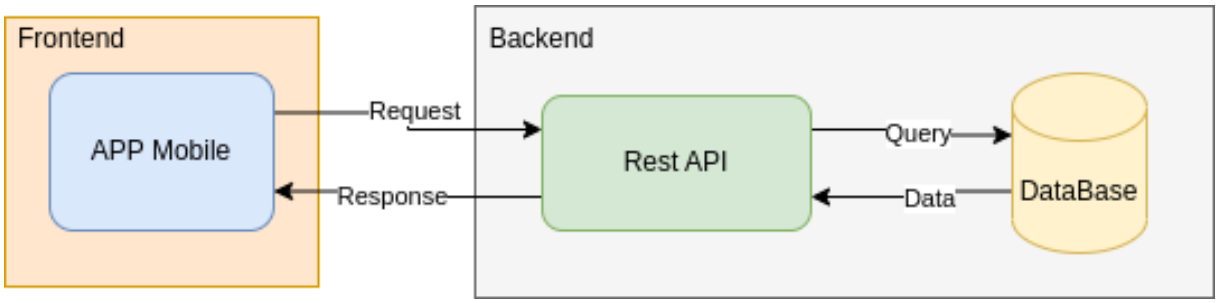

Figura 1.1: Arquitetura do sistema

1.1 Arquitetura de funcional

Para especificar a implementação da api rest foi então criada uma arquitetura de backend(Figura 1.2), nesta arquitetura é possível visualizar que sempre que a api recebe um request este é redirecionado primeiramente para o router, o router tem como função identificar a rota a ser pedida e redirecionar para os respetivos middlewares.

Os middlewares tem como função realizar todo o código necessário antes de proceder à execução do código de rota, os middlewares existentes são o SessionTokenValidator, este middleware tem como função validar a sessão do utilizador a realizar o pedido, de forma similar o middleware RefreshTokenValidator, valida a sessão principal do utilizador, por fim o middleware RoleValidator, tem como função validar se o utilizador que realiza o pedido tem cargos suficientes . Caso o pedido não seja impedido por nenhum middleware este é então direcionado para o controller.

O controller tem como função principal extrair os dados do pedido, validar os dados, verificando se os dados obrigatórios existem e encaminhar o pedido para o serviço, procedendo depois à formação da resposta e devolução da mesma. No serviço serão primeiramente aplicadas as regras de negócio para validar o conteúdo do pedido, caso o pedido não seja impedido por nenhuma das validações de regras de negócio, este então, em caso de necessidade, irá proceder à interação com base de dados, podendo esta realizar diversas interações como, obter dados, atualizar dados, apagar dados e inserir dados. Por fim a resposta é formada e devolvida como resposta ao pedido recebido.

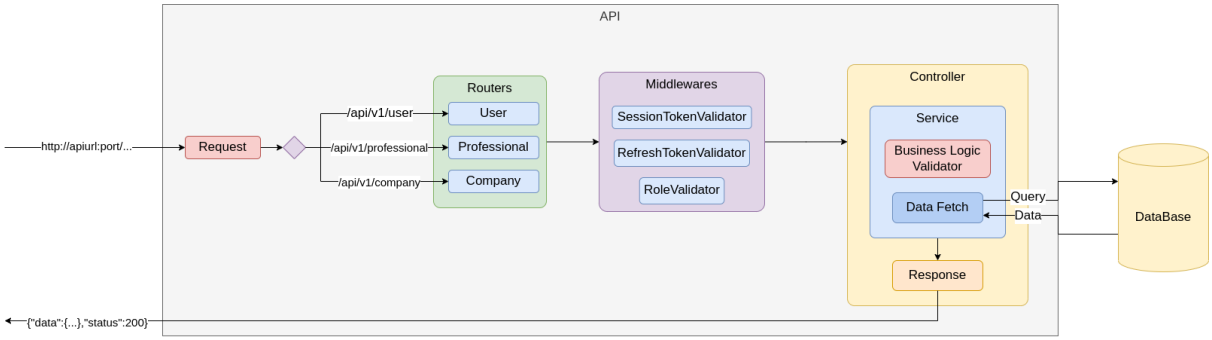

Figura 1.2: Arquitetura do funcional

1.2 Arquitetura de componentes

Após a perceção de todas as necessidades da aplicação do frontend, foi então desenvolvida a arquitetura de componentes na qual estão contidos todos os serviços que deverão ser implementados na api frontend, identificando também qual ator poderá realizar estes pedidos.

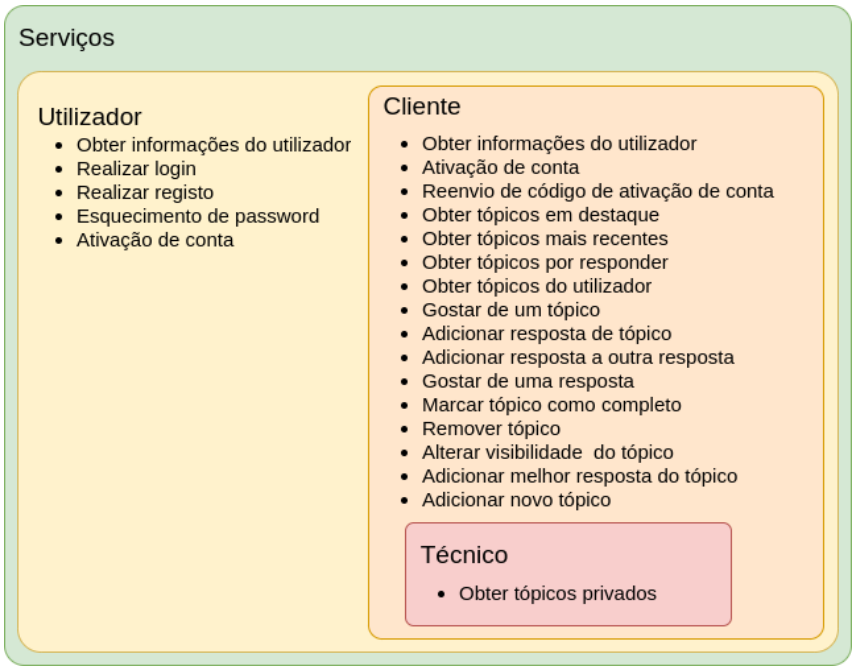

Figura 1.3: Arquitetura de componentes

1.3 Tabela de endpoints

De forma a evitar colisões de endpoints durante a implementação dos mesmos, foi então desenvolvida a tabela de endpoints que contém uma estrutura semelhante à arquitetura de componentes, mas que contém para cada serviço a rota e o método a utilizar.

Tabela 1.1: Tabela de endpoints

Serviço	Ator	Rota	Método
Obter informações do utilizador	Cliente	baseurl/client/:uid	GET
Realizar login	Utilizador	baseurl/login	POST
Realizar registo	Utilizador	baseurl/register	POST
Esquecimento de password	Utilizador	baseurl/forgot-password	GET
Ativação de conta	Cliente	baseurl/client/:uid/activate	POST

Continued on next page

Tabela 1.1: Tabela de endpoints (Continued)

Reenvio de código de ativação de conta	Cliente	baseurl/client/:uid/new-code	GET
Obter tópicos em destaque	Cliente	baseurl/client/topics/featured	GET
Obter tópicos mais recentes	Cliente	baseurl/client/topics/latest	GET
Obter tópicos por responder	Cliente	baseurl/client/topics/to-answer	GET
Obter tópicos do utilizador	Cliente	baseurl/client/topics	GET
Obter tópicos privados	Técnico	baseurl/professional/topics/private	GET
Gostar de um tópico	Cliente	baseurl/client/topics/:topicId/like	PUT
Adicionar resposta a tópico	Cliente	baseurl/client/topics/:topicId/answer	POST
Adicionar resposta a outra resposta	Cliente	baseurl/client/answers/:answerId/	POST
Gostar de uma resposta	Cliente	baseurl/client/answers/:answerId/like	PUT
Marcar tópico como completo	Cliente	baseurl/client/topics/:topicId/completed	PUT
Remover Tópico	Cliente	baseurl/client/topics/:topicId/	DELETE
Alterar visibilidade do tópico	Cliente	baseurl/client/topics/:topicId/visibility	PUT
Adicionar melhor resposta do tópico	Cliente	baseurl/client/topics/:topicId/answers/ :answerId/best-answer	PUT
Adicionar novo tópico	Cliente	baseurl/client/topics/	POST

1.4 Web scraper

Após uma reunião com o cliente foi percebido que o catálogo de produtos Motorline não se encontra em um servidor, esta informação encontra-se apenas diretamente no website da empresa, sendo assim viu-se a necessidade de criar um web scraper.

Web scraping é uma terminologia dada para o processo de ler uma página web com o objetivo de obter informações desta página, geralmente utilizando bots. O grande problema com web scraping é que se não for realizado com cuidado é possível sobrelotar o servidor da página web.

Sendo assim o scraper irá correr apenas 1 vez por mês visto que o catálogo não é atualizado de forma regular. Para agilizar a realização do scraper foi disponibilidado pela empresa a estrutura do website a seguir para obter as informações da página web.

1.5 falar com prof sobre como integrar isto