

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

1	Arquitetura de sistema	1
1.1	Arquitetura de funcional	2
1.2	Arquitetura de componentes	3
1.3	Tabela de endpoints	3
1.4	Web scraper	5
1.5	falar com prof sobre como integrar isto	5

Lista de Figuras

1.1	Arquitetura do sistema	1
1.2	Arquitetura do funcional	2
1.3	Arquitetura de componentes	3

Lista de Tabelas

1.1	Tabela de endpoints	3
-----	-------------------------------	---

1. Introdução

Este projeto intitulado Install & Go, é um aplicativo para smartphone dedicado a todos os clientes e técnicos Motorline, com o propósito de agilizar todo o processo de resolução de problemas e de acesso aos produtos. Para alcançar estes objetivos a aplicação conta com um catálogo para todos os utilizadores e assim como também um fórum para os clientes e técnicos Motorline.

Visto que o projeto a desenvolver seria de grande dimensão este foi particionado em diferentes etapas sendo que a cada etapa o objetivo seria acrescentar novas funções e melhorar a etapa anterior até se alcançar o produto final desejado.

Visto que o projeto é dividido com mais uma colega, esta na primeira etapa ficou encarregue de desenvolver o catálogo de produtos. Sendo assim este projeto e as suas tarefas são partilhadas entre os dois elementos, mas com uma clara divisão de encargos entre catálogo e fórum.

O fórum da aplicação é uma plataforma que permite aos clientes e técnicos criar tópicos que podem conter perguntas e/ou problemas para a comunidade ou diretamente a técnicos Motorline. Estes tópicos podem então ser comentados pela comunidade, conseguindo também realizar uma comunicação através desta secção do tópico.

De forma a ser possível dar suporte a esta aplicação, foi desenvolvido também um backend, sendo apenas necessário para fórum, mas que tem ligação com os servidores de clientes e técnicos da Motorline, assim como também de produtos.

1.1 Objetivos

A plataforma de fórum do aplicativo deverá ser possível que a comunidade partilhe os seus tópicos, para isso o utilizador deverá conseguir criar tópicos indicando o problema e uma descrição do mesmo, conseguindo anexar imagens, indicar a categoria e subcategoria de produto, assim como também referenciar o produto, facilitando assim a identificação e resolução do problema.

A comunidade deverá também conseguir responder aos tópicos e comunicar na secção de comentários do tópico. O dono do tópico deverá conseguir colocar o seu tópico privado caso deseje que apenas técnicos Motorline respondam ao mesmo. Este deverá também conseguir indicar quando um tópico se encontra finalizado e qual a melhor resposta que obteve. A comunidade também deverá conseguir gostar de tópicos e comentários para destacar os mesmos perante a restante comunidade.

A comunidade deverá conseguir ver os tópicos em destaque, os mais recentes, os seus tópicos, os tópicos que não estão finalizados e os técnicos conseguirão também ver os tópicos privados existentes. A comunidade poderá também pesquisar por tópicos com assuntos específicos ou rapidamente através de um código QR, ou pesquisa por nome, poderá pesquisar por tópicos referentes a um produto. Para filtragem de pesquisa a comunidade deverá também conseguir seleccionar categoria e subcategoria de tópico.

1.2 Contexto

O projeto foi desenvolvido na empresa Motorline Eletrocelos S.A daqui em diante designada Motorline, esta empresa é especializada na produção e comercialização de automatismos para portas e portões, sistemas de controlo de acessos, sistemas de segurança, entre outros produtos relacionados com o setor da automação.

Este projeto vem por este meio resolver o problema de comunicação com o cliente, que atualmente para solucionar as suas questões tem de contactar a assistência técnica que por vezes pode estar sobrelotada, ou então os clientes têm de preencher um formulário para expor a sua questão.

Com esta solução os clientes e técnicos Motorline poderão procurar ou expor os seus problemas com a comunidade onde poderão ser respondidos por técnicos, ou outros clientes que já tiveram um problema igual e conseguem ajudar, agilizando assim o processo de resolução de problemas.

Esta plataforma de fórum enquadra-se numa aplicação móvel desenvolvida em conjunto com uma colega que estará encarregue de desenvolver a área de catálogo da aplicação.

1.3 Estrutura do documento

Este documento contém a descrição de todo o processo de engenharia de software, desenvolvimento e pensamento sobre o problema em mãos, sendo estes pontos divididos sobre diversos tópicos:

1. Análise do problema, onde é abordado o estado da arte sendo descrito o modelo de negócio.
2. Metodologia de trabalho, tópico que aborda a organização de tarefas do projeto.
3. Análise e modelação de software, onde é descrito o levantamento de requisitos, diagramas de caso de uso e diagramas de atividades, assim como também arquiteturas do projeto.
4. Implementação, em que é descrito as tecnologias utilizadas bem como a razão de utilização destas tecnologias.
5. Análise de resultados, onde é abordado testes de código em backend e frontend.
6. Conclusão

2. Metodologia de Trabalho

De forma a organizar todo o trabalho a desenvolver visto que este é dividido com mais uma colega, foi então utilizada a técnica de desenvolvimento ágil, conseguindo assim organizar todas as tarefas entre os elementos de desenvolvimento do projeto.

2.1 Ferramentas de trabalho utilizadas

De forma a ser possível a organização de todas as tarefas, foi utilizado Github Projects, esta ferramenta permite ligar um projeto a um repositório de Github, conseguindo também personalizar completamente todo o projeto e parâmetros das tarefas o que permite uma organização minuciosa destas.

O Microsoft Excel foi também utilizado para a engenharia de software onde foram descritos os requisitos do projeto, user stories e também a especificação de casos de uso. Esta ferramenta foi também utilizada para a organização de reuniões com o cliente e redação de tópicos a abordar e abordados nesta.

Para o desenvolvimento do design do software foi utilizado a ferramenta figma, que permite o design de todas as componentes do software tendo em conta as reais dimensões de um dispositivo. Esta ferramenta permite também criar uma apresentação interativa que consegue demonstrar o comportamento da aplicação como completamente desenvolvida.

O draw.io foi também utilizado para o desenho das arquiteturas do projeto tendo se revelado de grande auxílio visto que este permite uma grande liberdade no desenho. Esta ferramenta permite também conexão com github conseguindo assim facilmente guardar estes projetos de ter acesso a partir de qualquer dispositivo.

A engenharia de software foi realizada utilizando Visual Studio paradigm, esta é uma ferramenta muito completa contendo modelos e regras para a engenharia de software. Esta ferramenta tornou-se um grande auxílio no desenvolvimento da base de dados, pois é possível desenhar o modelo e exportar para um ficheiro de criação de base de dados.

2.2 Planificação do trabalho

De forma a obter uma visão geral do projeto e uma previsão de finalização foi realizada uma planificação expectável de tarefas. Visto que o supervisor de estágio propôs uma divisão do projeto por etapas, as tarefas de cada etapa eram definidas no início da etapa. O objetivo seria melhorar a solução ao fim de cada etapa implementando novas funções, pelo que a cada nova etapa todo o processo de engenharia de software teria de se repetir.

Sendo assim as etapas são:

- Etapa 1 - Desenvolvimento de fórum e catálogo de produtos - Figura ??.



Figura 2.1: Planeamento de sprint da etapa 1

2.3 Diagrama de sequência

Visto que a realização da autenticação, ativação e confirmação de conta requer passos extras e regras a seguir, foi necessário criar diagramas de sequência para especificar a sequência de interações do com o sistema.

2.3.1 Diagrama de sequência Login e ativação de conta

Através deste diagrama (Figura ??) é entende-se que assim que o técnico deseja realizar o login, primeiramente tem de verificar as credenciais, caso estas se encontrem incorretas, este receberá uma mensagem de erro, caso as credenciais estejam válidas e a conta esteja ativada o técnico ficará autenticado.

Caso o técnico coloque as credenciais corretas, mas a conta não esteja ativada, este irá realizar a ativação de conta, onde poderá enviar o código de ativação, caso esteja correto a sua conta será ativada, caso contrário este receberá uma mensagem de erro. Este poderá também cancelar a ativação de conta e pedir um novo email de ativação, onde será pedido novo código ao servidor, este será gerado e enviado.

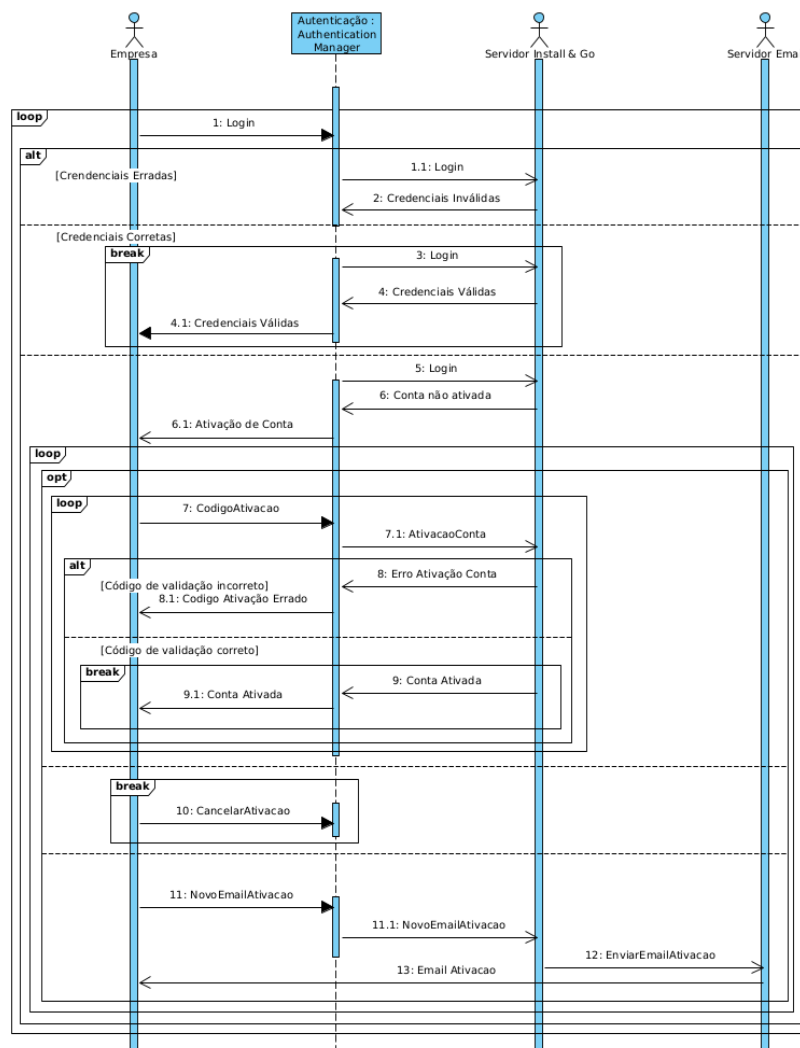


Figura 2.2: Diagrama de sequência de login e ativação de conta

2.3.2 Diagrama de sequência Registo e ativação de conta

Através do diagrama abaixo representado (Figura ??) é possível perceber que quando uma empresa realiza o registo este será enviado para o servidor, o qual registrará a empresa com uma conta não ativada, esta conta será então validada pela Motorline sendo de seguida gerado um código de ativação e enviado por email para o email de registo, após isto a empresa será encaminhado para a validação de conta, esta validação ocorre seguindo o mesmo processo mencionado no anteriormente.

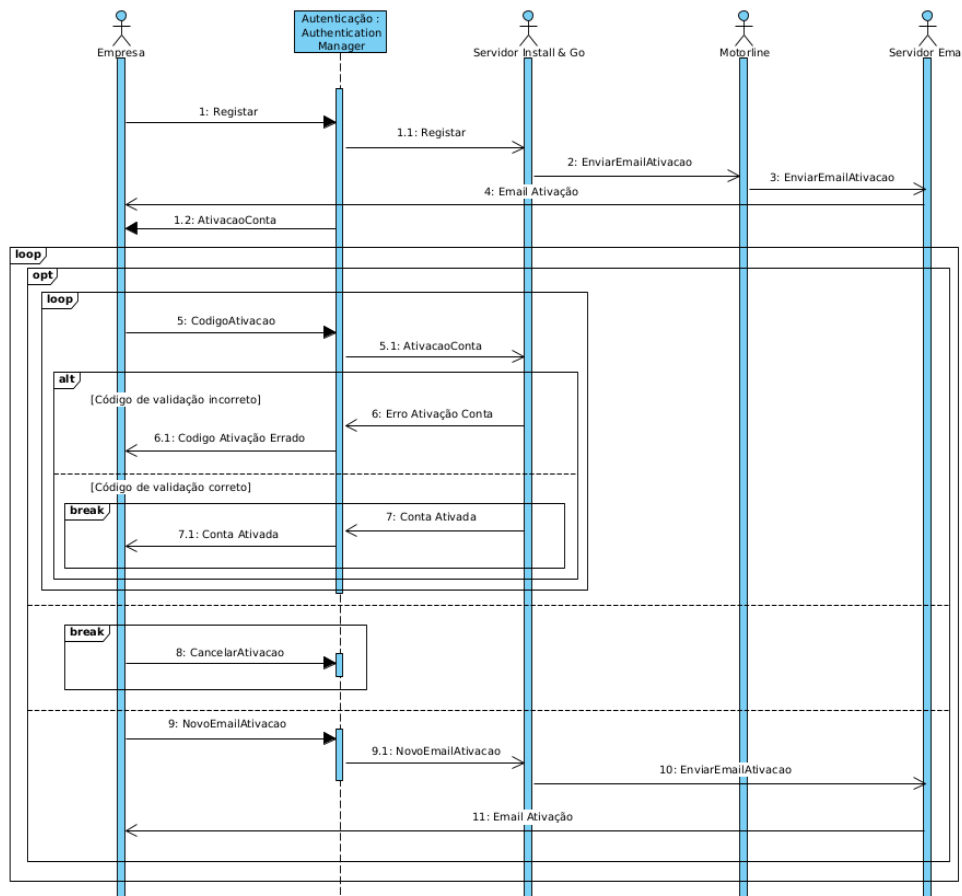


Figura 2.3: Diagrama de sequência de registo e validação de conta

2.3.3 Diagrama de sequência registro de técnicos

Através do diagrama abaixo representado (Figura ??) é possível perceber que quando uma empresa deseja registrar um técnico, esta introduzirá os seus dados, sendo a sua conta criada. Após isto, um código de ativação é gerado e enviado para o técnico ativar a sua conta.

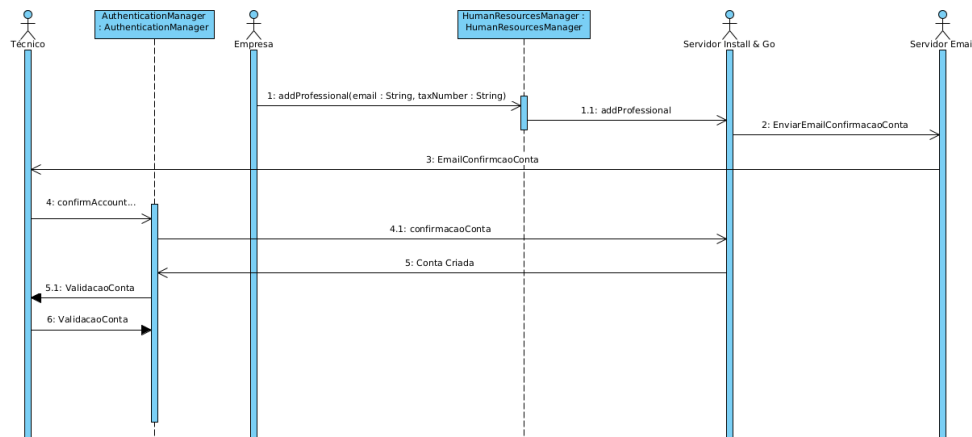


Figura 2.4: Diagrama de sequência de registro de técnicos

3. 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.

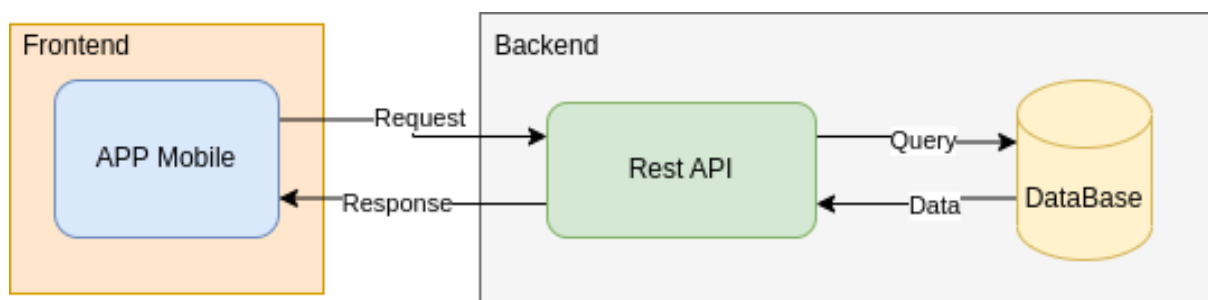


Figura 3.1: Arquitetura do sistema

3.1 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.

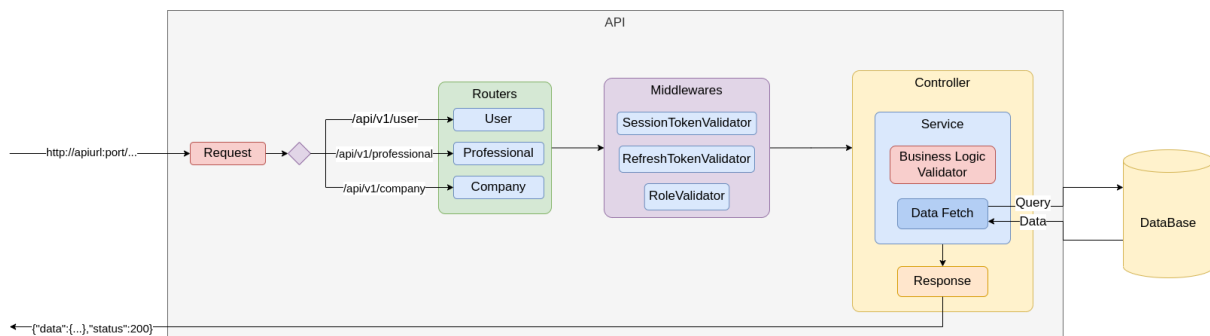


Figura 3.2: Arquitetura do funcional

3.2 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.

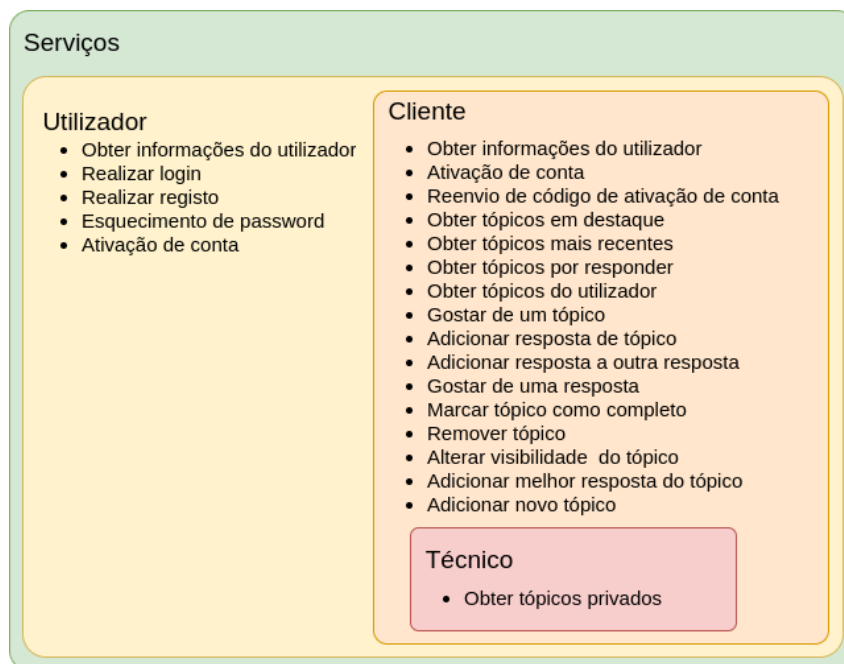


Figura 3.3: Arquitetura de componentes

3.3 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 3.1: 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 3.1: 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

3.4 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 disponibilizado pela empresa a estrutura do website a seguir para obter as informações da página web.

3.5 falar com prof sobre como integrar isto