

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	Introdução	1
1.1	Objetivos	2
1.2	Contexto	2
1.3	Estrutura do documento	3
2	Estado da arte	5
2.1	Ferramentas de trabalho utilizadas	5
2.2	Tecnologias utilizadas	6
2.2.1	Serviços Backend	6
2.2.1.1	Serviços RestFull e SOAP	6
2.2.1.2	TypeScript	7
2.2.1.3	Logs e Logging	7
2.2.1.4	Morgan	7
2.2.1.5	Envio de emails	8
2.2.1.6	Agendamento de tarefas	8
2.2.1.7	Encriptação de passwords	8
2.2.1.8	Cifragem de configurações do servidor	9
2.2.1.9	Firebase	9
2.3	Frontend	10
2.3.1	Desenvolvimento cross-platform	10
2.3.2	Flutter	11
2.3.3	Dart	11
2.3.4	Processo de aprendizagem	12
2.3.5	Links de aplicações	12
2.3.6	Qualidade de código	13
2.3.6.1	Design patterns	13
2.3.6.2	Documentação	13
2.3.6.3	Typedoc	13

2.3.6.4	Swagger	14
2.3.6.5	Testes de código	14
2.4	Planificação do trabalho	15
3	Analise e especificação	17
3.1	Modelo de negócio	17
3.2	Objetivos de negócio	17
3.3	Estado da arte	18
3.4	Domínio de aplicação do sistema	18
3.5	Operações a realizar no sistema	19
3.6	Descrição dos intervenientes	22
3.7	Partes Interessadas	22
3.8	Condições Específicas	23
3.9	Esquematização do conteúdo das páginas	24
3.9.1	Autenticação e página Inicial	25
3.9.2	Fórum	26
3.9.3	Criar novo tópico	26
3.9.4	Detalhes de tópico	27
3.9.5	Pesquisa de tópicos	27
3.9.6	Notificações	28
3.9.7	Perfil	28
3.9.8	Gestão de recursos humanos	29
3.10	Histórias de Utilizador	30
3.11	Casos de uso	33
3.11.1	Especificação de casos de uso	35
3.11.1.1	Especificação de caso de uso de listagem de tópicos	35
3.11.1.2	Especificação de caso de uso de criar novo tópico	36
3.11.1.3	Especificação de caso de uso de pesquisar tópicos por escrito	37
3.11.1.4	Especificação de caso de uso de ver finalizar tópico	37
3.11.1.5	Especificação de caso de uso de selecionar melhor resposta	38
3.11.1.6	Especificação de caso de uso de eliminar tópico	39
3.11.1.7	Especificação de caso de uso de alterar visibilidade de um tópico	39
3.11.1.8	Especificação de caso de uso gostar de um tópico	40
3.11.1.9	Especificação de caso de uso gostar de um comentário	40
3.11.1.10	Especificação de caso de uso de comentar tópico	41

3.11.1.11 Especificação de caso de uso ativar conta	42
3.11.1.12 Especificação de caso de uso configurar notificações	43
3.11.1.13 Especificação de caso de uso registar técnico	43
3.11.2 Diagramas de casos de uso	44
3.11.2.1 Casos de uso Fórum	44
3.11.2.2 Casos de uso de pesquisar tópicos	44
3.11.2.3 Casos de uso ver detalhes de tópico	45
3.11.2.4 Casos de uso ver comentários	45
3.11.2.5 Casos de uso ativação de conta	46
3.11.2.6 Casos de uso perfil	46
3.11.2.7 Casos de uso notificações	47
3.11.2.8 Casos de uso gestão de recursos humanos	47
3.12 Diagrama Entidade Relação	48
3.12.1 Escolha de diagrama de entidade relação	49
3.12.2 Dicionário de termos	50
3.13 Diagrama de Classes	54
3.14 Mockups	55
3.14.1 Página Inicial	55
3.14.2 Autenticação - Login e Registo	56
3.14.3 Autenticação - Ativação e Confirmação de conta	57
3.14.4 Página inicial fórum	58
3.14.5 Página de detalhes de um tópico	59
3.14.6 Página de criação de um tópico	60
3.14.7 Página de notificações	61
3.14.8 Página de perfil de utilizador	62
3.14.9 Página de gestão de recursos humanos	63
3.14.10 Página de perfil de técnico registado	64
3.14.11 Página de registo de novo técnico	65
3.15 Diagramas de atividades	66
3.15.1 Diagrama de atividades página inicial	66
3.15.2 Diagrama de atividades página de perfil	66
3.15.3 Diagrama de atividades página inicial do fórum	67
3.15.4 Diagrama de atividades página de criação de tópico	68
3.15.5 Diagrama de atividades página de detalhes do tópico	69
3.15.6 Diagrama de atividades páginas de autenticação	70
3.15.7 Diagrama de atividades registrar técnico	71

3.15.8 Diagrama de atividades confirmar conta	71
3.16 Diagramas de estados	72
3.16.1 Diagrama de estados criação de tópico	72
3.16.2 Diagrama de estados responder a tópico	73
3.16.3 Diagrama de estados autenticação e validação de conta	74
3.17 Diagrama de sequência	75
3.17.1 Diagrama de sequência Login e ativação de conta	75
3.17.2 Diagrama de sequência Registo e ativação de conta	76
3.17.3 Diagrama de sequência registo de técnicos	77
3.18 Arquitetura de sistema	78
3.18.1 Arquitetura de funcional	79
3.18.2 Arquitetura de componentes	80
3.18.3 Tabela de endpoints	81
4 Trabalho desenvolvido	83
4.1 Web scraper	83
4.1.1 Implementação web scraper	84
4.1.1.1 Implementação no website	89
4.1.1.2 Melhoria de implementação	90
4.1.1.3 Armazenamento de dados	94
4.2 Backend	95
4.2.1 Organização do projeto	95
4.2.2 Definição de rotas base	96
4.2.3 Middlewares	96
4.2.3.1 Linguagem	96
4.2.3.2 Autenticação	97
4.2.3.3 Validação de Papel	98
4.2.4 Controllers	98
4.2.4.1 Estruturação dos controllers	98
4.2.4.2 Execução da lógica de negócio	99
4.2.4.3 Validação dos dados	99
4.2.4.4 Formulação da resposta	99
4.2.4.5 Processamento de erros	100
4.2.4.6 Envio de emails	100
4.2.4.7 Logging de erros	101
4.2.4.8 Logging de pedidos com morgan	101

4.2.4.9	Agendamento de tarefas	101
4.2.4.10	Cifragem de passwords	102
4.2.4.11	Cifragem de configurações do servidor	102
4.2.4.12	Documentação Typedoc	102
4.2.4.13	Documentação Swagger	103
4.3	Frontend	104
4.3.1	Organização do projeto	104
4.3.2	Extensions	105
4.3.3	Handlers	105
4.3.4	Providers	105
4.3.5	Helpers	105
4.3.6	Gestão de utilizadores	106
4.3.7	Fórum	107
4.3.7.1	Filtragem de tópicos	108
4.3.7.2	Carregamento de tópicos	109
4.3.7.3	Detalhes de tópico	110
4.3.8	Firestorage	111
4.3.9	Apresentação de Imagens	112
4.3.10	Apresentação de Imagens em carrossel	113
4.3.11	Carregamento de Imagens	113
4.3.12	Videos	114
4.3.13	Links	114
4.3.14	Notificações	115
4.3.15	Permissões	116
4.3.16	Ios	117
5	Análise de resultados	119
5.1	Tarefas	119
5.2	Base de dados	120
5.3	Opinião do cliente	121

1. Introdução

Este projeto intitulado *Install & Go*, trata-se uma aplicação para *smartphone* dedicada 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 este projeto é dividido com mais uma colega, esta ficou encarregue de desenvolver o *frontend* do 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 os dois.

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. Estes tópicos podem então ser comentados, sendo também possível realizar uma comunicação através desta secção do tópico.

O suporte da aplicação foi realizado através de uma *api backend*, sendo inicialmente apenas necessário para fórum, mas necessitando também posteriormente de dar suporte para o catálogo de produtos.

1.1 Objetivos

A plataforma de fórum do aplicativo deverá permitir que a comunidade partilhe os seus tópicos, para isso o utilizador deverá conseguir criar tópicos indicando o problema e uma descrição do mesmo, com a possibilidade de anexar imagens, assim como também referenciar o produto, facilitando 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 Motorline 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 selecionar o tipo 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, pelo que deverão 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 têm a possibilidade de ser respondidos, 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. Estado da Arte, onde é abordado as ferramentas utilizadas, tecnologias exploradas e planificação de trabalho.
2. Análise e especificação, onde é abordado o estado da arte, descrito o modelo de negócio e apresentada toda a engenharia de *software*
3. Trabalho desenvolvido, onde é descrito todo o processo de desenvolvimento do projeto.
4. Analise de resultados, onde é discutido os resultados obtidos.
5. Conclusão e trabalho futuro, onde é abordada a conclusão sobre o projeto e também futuras implementações que poderiam ser realizadas.

2. Estado da arte

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

A organização de todas as tarefas, foi realizada na ferramenta *Github Projects*, esta 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 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, dando também suporte à implementação.

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 e ter acesso a partir de qualquer dispositivo.

A engenharia de *software* foi realizada através da utilização *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 Tecnologias utilizadas

2.2.1 Serviços Backend

De forma a realizar a integração entre a aplicação *frontend* e os dados, foi necessário desenvolver uma API para dar suporte a todos os serviços necessários para a aplicação. API sigla para *Application Programming Interface exposes a set of data and functions to facilitate interactions between computer programs and allow them to exchange information.* (Masse, 2011). Estas ferramentas apesar de serem *designed to work with other programs, they're mostly intended to be understood and used by humans writing those other programs* (Jin et al., 2018).

2.2.1.1 Serviços RestFull e SOAP

Os serviços RestFull *expose data as resources and use standard HTTP methods to represent Create, Read, Update, and Delete (CRUD) transactions against these resources* (Jin et al., 2018), sendo que a sua resposta é realizada em JSON *due to its simplicity and ease of use with JavaScript, JSON has become the standard for modern APIs.* (Jin et al., 2018).

Já SOAP é *nothing more than a simple XML-based envelope for the information being transferred, and a set of rules for translating application and platform-specific data types into XML representations*(Snell et al., 2002) sendo que a sua resposta é realizada em XML o que leva a que *It relies heavily on XML standards like XML Schema and XML Namespaces for its definition and function*(Snell et al., 2002). A utilização de XML permite que *two applications, regardless of operating system, programming language, or any other technical implementation detail, may openly share information using nothing more than a simple message encoded in a way that both applications understand.*(Snell et al., 2002)

Por fim foi decidido utilizar Rest devido a ser *much lighter compared to SOAP. It does not require formats like headers to be included in the message, like it is required in SOAP architecture.* Outra maior valia é que *it parses JSON, a human readable language designed to allow data exchange and making it easier to parse and use by the computer. It is estimated to be at around one hundred times faster than XML.*(Halili & Ramadani, 2018).

2.2.1.2 Typescript

O typescript *is a bit unusual as a language in that it neither runs in an interpreter (as Python and Ruby do) nor compiles down to a lower-level language (as Java and C do).*(Vanderkam, 2019) isto porque este *compiles to another high-level language, JavaScript.*(Vanderkam, 2019), isto faz com que o Typescript seja visto como *a superset of JavaScript in a syntactic sense.*

Todos os programas JavaScript *are TypeScript programs, but the converse is not true... This is because TypeScript adds additional syntax for specifying types..* O sistema de tipagens do TypeScript tem como objetivo *to detect code that will throw an exception at runtime, without having to run your code. ... The type checker cannot always spot code that will throw exceptions, but it will try.*(Vanderkam, 2019).

Esta linguagem de programação foi escolhida para o backend devido à sua capacidade de assegurar as tipagens, o que leva a um maior nível de segurança em termos de lidar com dados recebidos, assim como também a agilização do processo de programação devido à sua capacidade de prever a maioria dos erros de código.

2.2.1.3 Logs e Logging

Logs *are a very useful source of information for computer system resource management (printers, disk systems, battery backup systems, operating systems, etc.), user and application management (login and logout, application access, etc.), and security*(Chuvakin et al., 2012). Um Log é *what a computer system, device, software, etc. generates in response to some sort of stimuli. What exactly the stimuli are greatly depends on the source of the log message*(Chuvakin et al., 2012), ou seja perante um estímulo desejado, um log poderá ser gerado. Os dados dos logs são *the intrinsic meaning that a log message has*(Chuvakin et al., 2012), significando isto que estes contêm apenas dados relevantes ao objetivo do log. Logging é o nome que se dá ao processo de geração de logs.

Neste contexto logging poderá ser utilizado para realizar a monitorização de pedidos e erros. Estas informações poderão até auxiliar na tomada de decisões sobre o software e em quais funcionalidades deste software colocar mais atenção.

2.2.1.4 Morgan

Morgan é uma biblioteca que permite extrair dados de um pedido, assim como também a criação de logs, este atua como um middleware do servidor, recebendo qual o tipo de log a ser escrito, sendo estes tipos definidos pela biblioteca. Os principais dados obtidos pela biblioteca são a data e hora do pedido, o tipo de pedido, o serviço pedido, os dados recebidos, a resposta devolvida e também a descrição do sistema utilizado para realizar o pedido. Com estes dados é possível saber que plataforma é mais utilizada no software, quais as horas de maior utilização e quais os serviços mais executados, estes dados permitem direcionar mais recursos para uma indicada plataforma e/ou serviço, assim como também escolher os melhores horários de manutenção dos servidores.

2.2.1.5 Envio de emails

Para o envio de emails para os utilizadores, foi utilizada a biblioteca nodemailer, que permite a utilização de um servidor de SMTP para o envio de emails. Esta ferramenta foi escolhida devido a ser uma das mais utilizadas para esta tipo de necessidade, o que permite que exista mais informação sobre a mesma facilitando a resolução e identificação de erros.

Para desenvolver o conteúdo dos emails foi utilizada a ferramenta Tabular Email, esta ferramenta permite realizar o design do conteúdo de um email, sendo possível de seguida exportar o mesmo para html, a dificuldade desta ferramenta é que não permite a utilização de acentuação e visto que o html é gerado por uma máquina este torna-se complicado de navegar e traduzir.

2.2.1.6 Agendamento de tarefas

Um requisito para este projeto é o envio de emails com relatório diário de notificação todos os dias ao final do dia. Para realizar este primeiramente foi pesquisado que ferramentas existem para realizar este tipo de ações, pelo que foram encontradas o cronetab e o node-cron. A grande diferença entre estas duas ferramentas é, o cronetab funciona a nível de servidor sendo que sempre que se encontra na hora programada, este executa um comando indicado, este comando poderá por exemplo executar um código para enviar emails. Já o node-cron trata-se de uma biblioteca de NodeJs que trabalha com base no crontab, este permite o fácil agendamento de tarefas de forma programática assim como também a indicação do código a ser executado sem necessidade de criar comandos de execução de código.

Visto que a hora de execução do código de envio de emails poderá variar e necessitar de reprogramação foi então optado pela utilização do node-cron devido à sua facilidade de utilização, agilizando assim o processo de reprogramação de horas de envio de relatório.

2.2.1.7 Encriptação de passwords

De forma a garantir a segurança das passwords dos utilizadores é necessário encriptar estas, a encriptação poderá ser feita manualmente ou com o auxílio de ferramentas, a grande diferença é que manualmente poderá não se obter uma cifra tão segura como com o auxílio de uma ferramenta, sendo assim foi decidido utilizar uma ferramenta para encriptar passwords, a ferramenta escolhida foi bcrypt, esta foi escolhida devido a ser vastamente utilizada e até ao momento sem problemas em relação à sua cifra sendo que não existem registos de ataques bem sucedidos a esta ferramenta. Esta ferramenta oferece um conjunto de métodos sendo tendo sido utilizados os métodos de cifra e de comparação. O método de cifra permite através de um valor, chamado salt, indicar a complexidade a aplicar sobre a cifra, sendo de seguida devolvida a password cifrada. O método comparação permite comparar uma password cifrada com uma password sem cifra, devolvendo verdadeiro ou falso conforme as passwords sejam iguais ou não.

2.2.1.8 Cifragem de configurações do servidor

De forma a garantir um nível de segurança maior foram realizadas pesquisas sobre as principais falhas de segurança no NodeJs, pelo que foi descoberto que as principais formas de ataque a esta ferramenta é o desenvolvimento de bibliotecas de malware e o ataque às bibliotecas com o objetivo de obter dados de acesso a servidores que se encontram nos ficheiros de configuração.

Por norma todas as configurações de servidores são colocadas num ficheiro env, este ficheiro no momento de iniciar o servidor é utilizado para carregar todas as variáveis para o ambiente do mesmo, sendo assim qualquer um com acesso ao ficheiro ou às variáveis de ambiente poderá ver todas as configurações do servidor.

A solução mais indicada para este problema é a cifragem do ficheiro env e das variáveis de ambiente. A biblioteca mais utilizada para este objetivo é a secur-env, esta permite realizar a cifragem de um ficheiro indicando uma password. A password deverá ser indicada no processo de inicialização do servidor de forma a ser possível ao mesmo decifrar o ficheiro, sendo que a gestão das variáveis cifradas passa então a estar encarregue desta biblioteca.

Mesmo com esta solução existem possibilidades de ataque, pois é possível ver o histórico do terminal do servidor, pelo que é possível obter a password escrita no mesmo, para resolver este problema é indicada a biblioteca readline, pois esta possui o modo de password que apaga o histórico do terminal sempre que utilizado, esta contém a vertente assíncrona, readline e a vertente síncrona, readline-sync. Para o projeto foi utilizada a versão síncrona da biblioteca visto que o objetivo é o servidor apenas iniciar após a indicação da password, sem nenhum serviço a correr em simultâneo.

2.2.1.9 Firebase

Firebase é uma framework comprada pela google em 2014, o seu objetivo é *to provide the tools and infrastructure that you need to build great apps*(Moroney, 2017), esta alcança este objetivo através da oferta de serviços pré configurados, sendo que *many of the technologies are available at no cost*(Moroney, 2017).

Firestorage também conhecida como cloud storage, é um serviço que dispõe *a simple API that is backed up by Google Cloud Storage*(Moroney, 2017), que permite guardar e transmitir até 1GB de ficheiros de forma gratuita.

Cloud messaging é também um serviço do firebase que permite *to reliably deliver messages at no cost*(Moroney, 2017). Este garante que *Over 98% of connected devices receive these messages in less than 500ms*(Moroney, 2017). Cloud messaging também permite a utilização de diversas formas de envio de notificações como *driven by analytics to pick audiences, or using topics or other methods*(Moroney, 2017).

Dynamic links é um serviço do firebase que permite a criação de *links to an app that contain context about what you want the end user to see in the app*(Moroney, 2017)..

Esta ferramenta foi escolhida devido à sua capacidade de fornecer serviços pré configurados de forma gratuita, evitando assim o gasto monetário e de tempo durante o desenvolvimento.

2.3 Frontend

Um dos requisitos do projeto é o desenvolvimento do frontend utilizando a ferramenta Flutter, visto que a empresa já utiliza esta ferramenta, sendo assim necessário o aprendizado desta ferramenta e da sua linguagem de programação o dart.

2.3.1 Desenvolvimento cross-platform

O desenvolvimento de aplicações cross-platform ou multi-plataforma, conssite no desenvolvimento de uma aplicação para diversas plataformas e este pode ser realizado de diversas formas, mas as principais formas conhecidas são WebView, nativo e outras abordagens.

A frameworks nativas são "*the most stable choice for mobile application development*"(Mainkar & Giordano, 2019) e dispõem de um grande comunidade e leque de aplicações desenvolvidas. O que torna estas frameworks estáveis é o facto de "*the app in this framework talks directly to the system*"(Mainkar & Giordano, 2019), todo o desenho no ecrã é realizado através de o que é chamado de *OEM components* que são disponibilizados pela framework mas não permitem customização total. A grande desvantagem desta abordagem é o facto de que se o objetivo do projeto é o desenvolvimento para ios e android então "*you need to learn two different languages*"(Mainkar & Giordano, 2019), porque estas são utilizadas para "*write two different apps with the same functionalities*"(Mainkar & Giordano, 2019) o que significa que "*Every modification must be duplicated on both platforms*"(Mainkar & Giordano, 2019).

Uma outra abordagem para o desenvolvimento para diversas plataformas através de uma única base de código é o WebView. "*Cordova-, Ionic-, PhoneGap-, and WebView-based frameworks in general are good examples of cross-platform frameworks*"(Mainkar & Giordano, 2019), mas o grande problema desta abordagem é a "*lack in performance*"(Mainkar & Giordano, 2019) pois esta é composta por um processo intermédio chamado WebView que renderiza código HTML, isto significa que "*the app is basically a website*"(Mainkar & Giordano, 2019). Esta abordagem acrescenta também o componente de ponte que realiza o "*switch between JavaScript to the native system*"(Mainkar & Giordano, 2019) para obter acesso aos serviços nativos.

Um concorrente à tecnologia mencionada no ponto seguinte é o React Native, este assim como as frameworks nativas "*heavily relies on OEM components*"(Mainkar & Giordano, 2019), este "*expands the bridge concept in the WebView systems, and uses it not only for services, but also to build widgets*"(Mainkar & Giordano, 2019), isto leva a grandes problemas em termos de performance devido a que "*a component may be built hundreds of times during an animation, but due to the expanded concept of the bridge, this component may slow down to a great extent*"(Mainkar & Giordano, 2019).

2.3.2 Flutter

Flutter é uma framework desenvolvida pela google, de inicio "*lutter was an experiment, as the developers at Google were trying to remove a few compatibility supports from Chrome, to try to make it run smoother.*", por acabaram por descobrir que "*hey had something that rendered 20 times faster than Chrome did and saw that it had the potential to be something great*". Em suma google desenvolveu "*a layered framework that communicated directly with the CPU and the GPU in order to allow the developer to customize the applications as much as possible*".

Para o flutter tudo é um widget, "*Orientation, layout, opacity, animation... everything is just a widget*", isto permite que os utilizadores "*choose composition over inheritance, making the construction of an app as simple as building a Lego tower*". Todos estes widgets oficiais estão catalogados no catalogo de widgets do flutter. Como tudo no flutter é composto por widgets "*the more you learn how to use, create, and compose them, the better and faster you become at using Flutter*".

A abordagem ao cross-platform realizada pelo flutter é baseada em "*AOT (Ahead Of Time) instead of JIT (Just In Time) like the JavaScript solutions*" mostradas anteriormente. Esta também permite a conversação direta com o cpu sem necessidade de ponte e "*does not rely on the OEM platform*". Esta permite "*custom components to use all the pixels in the screen*", o que significa que "*the app displays the same on every version of Android and iOS*". Esta também utiliza "*Platform Channels to use the services*", o que permite que "*if you need to use a specific Android or iOS feature, you can do it easily*".

2.3.3 Dart

Dart é a linguagem de programação utilizada pela framwork flutter, esta é "*a general purpose programming language*"(Bracha, 2015) que foi desenhada para ser "*familiar to the vast majority of programmers*"(Bracha, 2015). Esta linguagem é "*purely object-oriented*" o que significa que "*all values a Dart program manipulates at run time are objects*"(Bracha, 2015), até tipos básicos como números e boleanos, esta é também "*class-based, optionally typed*", esta é opcionalmente tipada, o que significa que a decisao de utilizar tipagens cai sobre o programador, mas no caso de flutter, na sua versão mais recente é recomendado a utilização de tipagens de variáveis, por fim esta "*supports mixin-based inheritance and actor-style concurrency*"(Bracha, 2015).

2.3.4 Processo de aprendizagem

De forma a realizar a aprendizagem da ferramenta foi então procedido para a desenvolvedora da mesma, a google, esta dispõe de uma lista de workshops e projetos a seguir de forma a aprender as bases da ferramenta, sendo a lista a seguinte:

1. MDC-101 no Flutter: noções básicas dos componentes do Material Design | Google Codelabs
2. MDC-102 no Flutter: estrutura e layout do Material Design | Google Codelabs
3. MDC-103 Flutter: temas do Material Design com cores, formas, elevação e tipo | Google Codelabs
4. MDC-104 Flutter: componentes avançados do Material Design | Google Codelabs
5. Apps adaptáveis no Flutter | Google Codelabs

Após a realização destes workshops, foi possível entender como funciona a base desta ferramenta e também encontrar fontes para pesquisa de widgets da comunidade a nível gráfico e funcional, provando estes serem de grande auxílio no desenvolvimento da aplicação frontend.

2.3.5 Links de aplicações

Existem diferentes tipos de links sendo que para o a utilização em mobile é utilizado os app links, deep links e os dynamic links.

Os app links são "*web links that use the HTTP and HTTPS schemes*"(Developers, 2023), estes possuem também um atributo extra chamado autoVerify, este atributo permite a uma aplicação "*to designate itself as the default handler of a given type of link*"(Developers, 2023), isto permite que "*app opens immediately if it's installed*"(Developers, 2023). O grande problema é que estes links não permitem o redirecionamento do utilizador para uma parte específica da aplicação e é necessário dispor de um domínio próprio.

Os deep links são "*URIs of any scheme that take users directly to a specific part of your app*"(Developers, 2023), o grande problema deste tipo de links é que se os utilizadores não disporem da aplicação instalada no dispositivo, este irá falhar, nem permite a costumização de comportamento.

Já os dynamic links desenvolvidos pela firebase assim como os deep links "*if a user opens a Dynamic Link on iOS or Android, they can be taken directly to the linked content in your native app*"(Firebase, 2023), mas para além disto, este permite também que "*if a user opens the same Dynamic Link in a desktop browser, they can be taken to the equivalent content on your website*"(Firebase, 2023), ou seja este permite a costumização de comportamento de links para diversas situações e em caso do utilizador não dispor da aplicação instalada, este permite que "*the user can be prompted to install it; then, after installation, your app starts and can access the link*"(Firebase, 2023). Visto que este é o comportamento desejado pelo cliente da aplicação, então foi decidido utilizar esta abordagem.

2.3.6 Qualidade de código

A qualidade do código desenvolvido é de extrema importância de forma a possibilitar e facilitar a manutenção da solução desenvolvida, assim como também a melhoria da segurança da mesma. Esta permite a percepção de objetivo de código assim como também a estruturação do mesmo de acordo com normas estabelecidas perante a comunidade. A qualidade de código tem como objetivo também diminuir a complexidade do mesmo, pois nem sempre documentação e estruturação é o suficiente para o código ser de qualidade, sendo que este deverá ser simplificado de forma a ser interpretável. Todos estes pontos poderão ser implementados através do uso de design patterns, documentação e testes de código.

2.3.6.1 Design patterns

Os design patterns "*names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design*". Estes identificam "*the participating classes and instances, their roles and collaborations, and the distribution of responsibilities*".

O MVC consiste em "*three kinds of objects*", o Model que é "*the application object*", a View que é a "*screen presentation*" do Model e por fim o Controller que "*defines the way the user interface reacts to user input*". Antes do MVC "*user interface designs tended to lump these objects together*", o MVC "*decouples them to increase flexibility and reuse*", através de "*esta blising a subscribe/notify protocol between them*".

2.3.6.2 Documentação

De forma a ser possível manter todo o projeto desenvolvido foi criada documentação a diferentes níveis, sendo esta desenvolvida a nível de serviços explicando o objetivo do serviço e que dados este recebe, como também a nível de código explicando o código desenvolvido em cada script existente. Para estes níveis de documentação foram utilizadas diferentes ferramentas, para documentação de serviços, foi utilizada a ferramenta swagger e para a documentação foi utilizado typedoc.

2.3.6.3 Typedoc

Typedoc é uma ferramenta que faz utilização de comentários de código para gerar a sua documentação, esta documentação utiliza chaves específicas para detetar as informações de documentação, estas permitem também criar categorias para melhor organizar toda a documentação. Esta documentação permite também a interligação entre si mesma permitindo ao visualizador desta seguir todo o processo. Após a realização de geração de documentação, a ferramenta gera um website onde é possível navegar por toda a documentação gerada.

2.3.6.4 Swagger

Swagger é uma ferramenta que permite gerar documentação a nível de serviços, esta documentação é acessível a partir do mesmo servidor indicando uma rota para o mesmo, evitando assim outro servidor para hospedar a documentação, a base de toda a documentação encontra-se em um ficheiro no formato json. Esta documentação poderá ser gerada a partir de comentários a nível de código ou a partir de um ficheiro em formato json como mencionado anteriormente. Este ficheiro em formato json poderá ser mantido manualmente ou automaticamente.

2.3.6.5 Testes de código

Aquando o fim do desenvolvimento de cada serviço é necessário testar este de forma a verificar se a funcionalidade se encontra de acordo com o desejado e/ou se existem erros de código. Para realizar estes testes poderão ser utilizadas ferramentas de auxílio ou então poderão ser realizados manualmente. O grande problema de testes manuais é que são exaustivos devido a serem longos e propensos a erros, pelo que estes testes são realizados em menor escala do projeto. Para os testes deste projeto foram utilizadas ferramentas de auxílio, sendo as ferramentas escolhidas mocha e chai.

A realização de estes de código foi muito importante pois com esta ferramenta foi possível encontrar erros de lógica de negócio bem como também erros de código tanto a nível da formulação de respostas como a nível de código.

2.4 Planificação do trabalho

De forma a obter uma visão geral do projeto e uma previsão de finalização foi realizada uma planificação expectável de tarefas.



Figura 2.1: Planeamento de sprints

3. Analise e especificação

Na proposta de sistema está contido a descrição do modelo de negócio, onde serão identificados os principais problemas do modelo atual e será indicada a necessidade de implementação do software. Nesta proposta estará também identificado os objetivos de negócio, conseguindo identificar o principal impacto deste software neste modelo de negócio. Nesta proposta estará também contida a engenharia de software com toda a especificação do projeto em mãos.

3.1 Modelo de negócio

Este projeto surgiu com a necessidade da Motorline melhorar a comunicação e a experiência das empresas clientes e técnicos. Atualmente sempre que um técnico possui uma questão este deverá contactar o suporte técnico o que pode levar a sobrecarga deste, ou então este deverá se juntar ao grupo do Facebook de técnicos e colocar neste esta questão. Sempre que é necessário um manual de produto ou sempre que o utilizador pretender ver o catálogo este deverá se deslocar ao site da Motorline e procurar o produto no catálogo não possuindo um acesso rápido a estes. Visto que este processo acaba por não ser prático para o utilizador, foi então decidido suprir este problema com o projeto Install & Go. Tendo estas questões em mente surgiu então a ideia de implementar um fórum de resolução de problemas.

3.2 Objetivos de negócio

Este software visa minimizar o problema acima descrito. Visto que acontece que muitas das questões dos técnicos são comuns, então surgiu a ideia de implementar um fórum onde o técnico poderá pesquisar por questões semelhantes à sua, encontrando solução sem necessidade de contactar o suporte técnico. O técnico poderá também expor a sua questão anexando imagens, facilitando o processo de resolução da sua questão. Para existir partilha de conhecimento o técnico poderá também responder a tópicos, pois caso encontre alguma questão que já tenha resolvido poderá ajudar outro técnico. O técnico também poderá alterar a visibilidade do seu tópico caso deseje que apenas técnicos visualizem a sua questão.

A empresa poderá realizar as mesmas ações que o técnico, mas esta poderá também criar contas para os seus técnicos e realizar a gestão destas sendo importante permitir apagar e impedir acesso à conta em caso de necessidade.

3.3 Estado da arte

Para ser possível entender todo o negócio onde este projeto se encaixa foi realizado um estudo englobando outras soluções do mesmo ramo. Estas soluções investigadas foram indicação do supervisor de estágio pois este já teria realizado um estudo da situação do negócio préviamente. As soluções investigadas foram, FixAndGo, My VDS e Ultimate Cardin. A solução FixAndGo foi indicada devido à capacidade de calcular a possibilidade de instalar em motor baseando-se em medidas indicadas pelo utilizador. Já a solução My VDS foi indicada devido a dispor documentação sobre os seus produtos como manuais e vídeos, assim como também uma forma de comunicação com a equipa de suporte. Por fim a solução Ultimate Cardin foi utilizada como exemplo de design a não seguir devido a ser confuso, esta foi também utilizada como exemplo de organização de design do ecrã de detalhes de produto.

3.4 Domínio de aplicação do sistema

Com o diagrama abaixo representado (Figura 3.1) é possível visualizar todos os atores do software e as suas ações, assim como também os sistemas envolvidos na aplicação e as suas funções. Destes é possível identificar que este software contém três atores principais, o Utilizador que é um utilizador sem sessão iniciada, o Técnico que é um utilizador com sessão iniciada, já a Empresa é uma empresa cliente da Motorline. Também é possível visualizar os diferentes sistemas integrados no projeto, como Servidor Motorline onde serão obtidas informações de catálogo de produtos, Servidor Install & Go onde estão todas as funções de suporte ao software, o Servidor de Imagens onde serão guardadas todas as imagens do fórum e por fim o Servidor de Email que enviará email com o código de validação de conta para os clientes assim que se registarem no software .

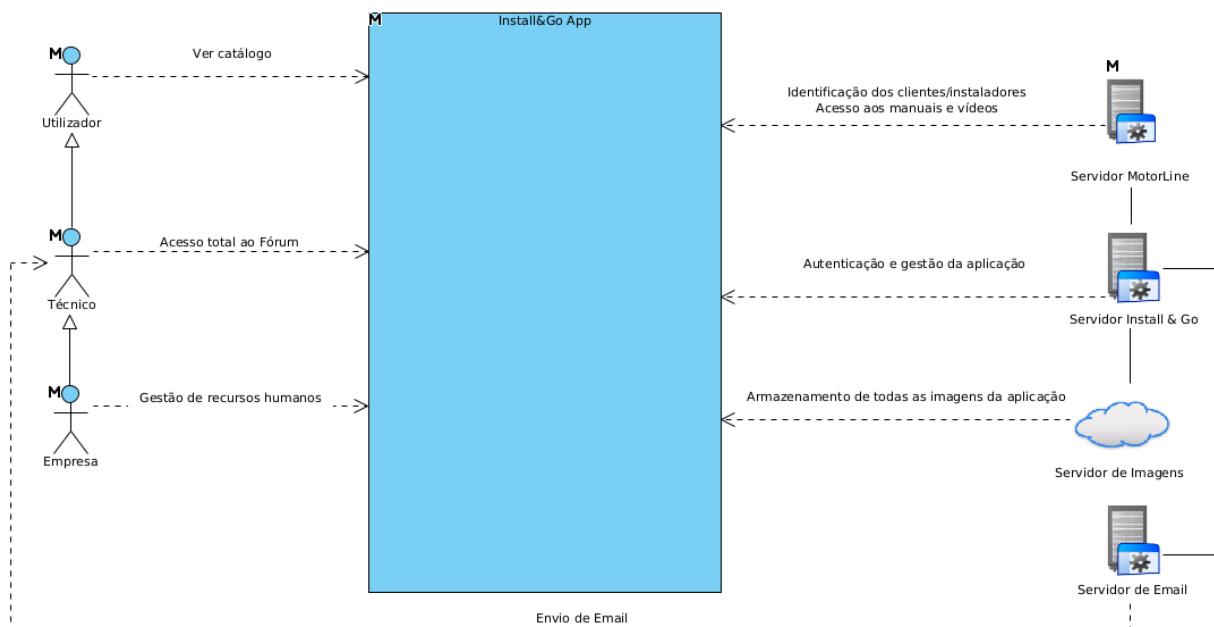


Figura 3.1: Diagrama de contexto da aplicação

3.5 Operações a realizar no sistema

A primeira tarefa a realizar no desenvolvimento desta etapa do projeto foi o levantamento de requisitos funcionais(Tabela 3.1) e não funcionais do projeto. Tendo sido estes posteriormente validados com cliente.

Tabela 3.1: Tabela de requisitos funcionais

#	Descrição	Fonte	Data
Autenticação			
RF01	O Utilizador deverá conseguir visualizar o catálogo	Helder Remelhe	2/13/2023
RF02	A Empresa deverá conseguir realizar o registo na aplicação	Helder Remelhe	2/13/2023
RF03	O Técnico deverá conseguir realizar o login na aplicação	Helder Remelhe	2/13/2023
RF04	O login deverá ser realizado utilizando número de contribuinte e password	Helder Remelhe	2/13/2023
RF05	Assim que o registo é realizado, a motor-line deverá validar a conta sendo posteriormente enviado um email para a empresa ativar e utilizar a conta	Rafael Viana	2/13/2023
RF06	O Técnico deverá conseguir pedir reenvio de código de verificação de conta	Rafael Viana	2/27/2023
RF07	Os Técnicos deverão ser identificados como técnicos certificados ou técnicos oficiais	Helder Remelhe	2/27/2023
Fórum			
RF08	O Técnico deverá conseguir aceder ao fórum e realizar operações	Helder Remelhe	2/13/2023
RF09	O Técnico deverá conseguir visualizar os tópicos mais recentes	Brainstorming	2/14/2023
RF10	O Técnico deverá conseguir visualizar os tópicos em destaque	Brainstorming	2/14/2023
RF11	O Técnico deverá conseguir visualizar os tópicos por responder	Brainstorming	2/14/2023
RF12	O Técnico deverá conseguir acessar aos tópicos privados do fórum	Helder Remelhe	2/13/2023
RF13	O Técnico deverá conseguir pesquisar por tópicos referentes a um assunto desejado	Brainstorming	2/14/2023
RF14	O Técnico deverá conseguir pesquisar por tópicos referentes a um produto desejado	Brainstorming	2/14/2023

Continued on next page

Tabela 3.1: Tabela de requisitos funcionais (Continued)

RF15	O Técnico deverá conseguir pesquisar por tópicos referentes a um produto por código QR	Brainstorming	2/14/2023
RF16	O Técnico deverá conseguir visualizar os seus tópicos	Brainstorming	2/14/2023
Criar Tópico			
RF17	O Técnico deverá conseguir criar tópicos para expor a sua questão	Helder Remelhe	2/13/2023
RF18	O Técnico deverá conseguir colocar o seu tópico público ou privado para assim apenas outras empresas conseguirem ver	Helder Remelhe	2/13/2023
RF19	O Técnico deverá conseguir indicar tipo de tópico para agilizar a identificação do mesmo	Helder Remelhe	2/14/2023
RF20	O Técnico deverá conseguir indicar o produto referente ao tópico para agilizar a identificação da sua questão	Brainstorming	2/14/2023
RF21	O Técnico deverá conseguir anexar imagens ou vídeos ao tópico em questão para agilizar a comunicação e identificação do seu problema	Helder Remelhe	2/13/2023
Gestão de tópico			
RF22	O Técnico deverá conseguir indicar qual a melhor resposta ao seu tópico para assim facilitar a pesquisa por resposta a outras empresas que possuam o mesmo problema	Helder Remelhe	2/14/2023
RF23	O Técnico deverá conseguir indicar que o tópico se encontra finalizado quando o seu problema se encontrar resolvido	Helder Remelhe	2/13/2023
RF24	O Técnico deverá conseguir remover o seu tópico	Brainstorming	2/14/2023
RF25	O Técnico deverá conseguir alterar a visibilidade do seu tópico	Helder Remelhe	2/13/2023
Tópicos			
RF26	O Técnico deverá conseguir ver todas as respostas ao tópico	Helder Remelhe	2/13/2023
RF27	O Técnico deverá conseguir gostar do tópico para dar destaque ao mesmo	Brainstorming	2/14/2023
RF28	O Técnico deverá conseguir remover uma resposta que colocou em um tópico	Brainstorming	2/14/2023

Continued on next page

Tabela 3.1: Tabela de requisitos funcionais (Continued)

	Respostas a Tópicos		
RF29	O Técnico deverá conseguir comentar tópicos	Helder Remelhe	2/13/2023
RF30	O Técnico deverá conseguir responder a comentários de tópicos	Helder Remelhe	2/13/2023
RF31	O Técnico deverá conseguir anexar imagens e videos à sua resposta	Helder Remelhe	2/13/2023
RF32	O Técnico deverá conseguir gostar de alguma resposta para dar destaque a esta resposta	Helder Remelhe	2/13/2023
	Perfil		
RF33	O Técnico deverá conseguir alterar o seu email	Helder Remelhe	2/27/2023
RF34	O Técnico deverá conseguir alterar imagem de perfil	Brainstorming	2/27/2023
RF35	O Técnico deverá conseguir alterar nome	Brainstorming	2/27/2023
RF36	O Técnico deverá ser identificado como Técnico oficial ou certificado	Helder Remelhe	2/28/2023
	Notificações		
RF37	O Técnico deverá conseguir receber notificações por email e/ou push	Rafael Viana	2/27/2023
RF38	O Técnico deverá conseguir alterar o tipo de notificação entre relatório diário e notificação direta para ambos os tipos	Rafael Viana	2/27/2023
RF39	O Técnico deverá conseguir ver as suas notificações	Helder Remelhe	2/27/2023
RF40	O Técnico deverá conseguir apagar as suas notificações	Helder Remelhe	2/27/2023
	Gestão de recursos humanos		
RF41	A Empresa deverá conseguir criar contas para os seus técnicos	Brainstorming	2/13/2023
RF42	Assim que a conta de técnico for criada, este deverá receber um email para registrar as restantes informações e ativar a conta	Helder Remelhe	2/27/2023
RF43	A Empresa deverá conseguir impedir acesso a uma conta de técnico	Helder Remelhe	2/27/2023
RF44	A Empresa deverá conseguir remover uma conta de técnico da empresa	Helder Remelhe	2/27/2023

3.6 Descrição dos intervenientes

O projeto envolve um conjunto de intervenientes, sendo estes, o utilizador, a empresa e o técnico. Estes desempenham um papel fundamental e podem realizar diferentes ações.

O utilizador sem sessão iniciada terá um fácil e rápido acesso ao catálogo de produtos, o que irá facilitar quando este deseja realizar a consulta do mesmo, terá também acesso apenas de visão de questões públicas do fórum, não conseguindo realizar nenhuma operação.

O técnico conseguirá realizar as mesmas ações que o utilizador, mas este ator conseguirá também ter acesso total ao fórum e a questões privadas. O fórum permite expor questões, com auxílio de imagens, permite a ligação da questão a uma categoria de questões e um produto em específico para facilitar a resolução da sua questão. As questões poderão ser públicas para assim qualquer um poder ver, ou então estas podem ser privadas para apenas técnicos conseguirem ver. Assim que o técnico estiver satisfeito com a sua questão este poderá indicar a melhor resposta que obteve para a destacar e então finalizar o tópico, para assim este ser indicado como finalizado.

O técnico pode também realizar pesquisas por questões em caso de ter algum problema. Com isto, este evita um telefonema ou o preenchimento de um formulário para contactar um técnico. Ao realizar pesquisas por questões este pode responder a outras questões, pode também pesquisar por questões em categoria e produto. As respostas podem conter imagens anexadas e podem também responder a outras respostas para manter uma comunicação continua. O técnico poderá também destacar tópicos e respostas de tópicos gostando destas.

A empresa pode realizar a gestão de contas de técnicos, conseguindo criar, impedir acesso e eliminar em caso de necessidade.

3.7 Partes Interessadas

Este projeto foi proposto pelo supervisor de estágio, sendo então este com a empresa Motorline representada por, Rafael Viana e André Viana, as partes interessadas.

3.8 Condições Específicas

Os requisitos não funcionais são características do software que não interferem diretamente com o funcionamento normal do software, estes podem referir-se a características como segurança e culta. Na Tabela 3.2 é possível visualizar os requisitos não funcionais levantados.

Tabela 3.2: Tabela de requisitos não funcionais

#	Tipo	Descrição
RNF01	Cultural	O software tem de suportar vários idiomas (prioritariamente, português e inglês)
RNF02	Configuração	A falha dos servidores, implica a inutilidade total da aplicação
RNF03	Conexão	Necessário o uso de dados móveis ou WIFI
RNF04	Segurança	O software tem de ser seguro para proteger os dados do consumidor
RNF05	Desenvolvimento	A aplicação deverá ser desenvolvida utilizando a tecnologia flutter

3.9 Esquematização do conteúdo das páginas

De forma a ser possível entender que dados serão necessários para alimentar o software, o que apresentar em cada página e também como navegar entre os ecrãs da aplicação foi então desenhado um esquema (Figura 3.2).

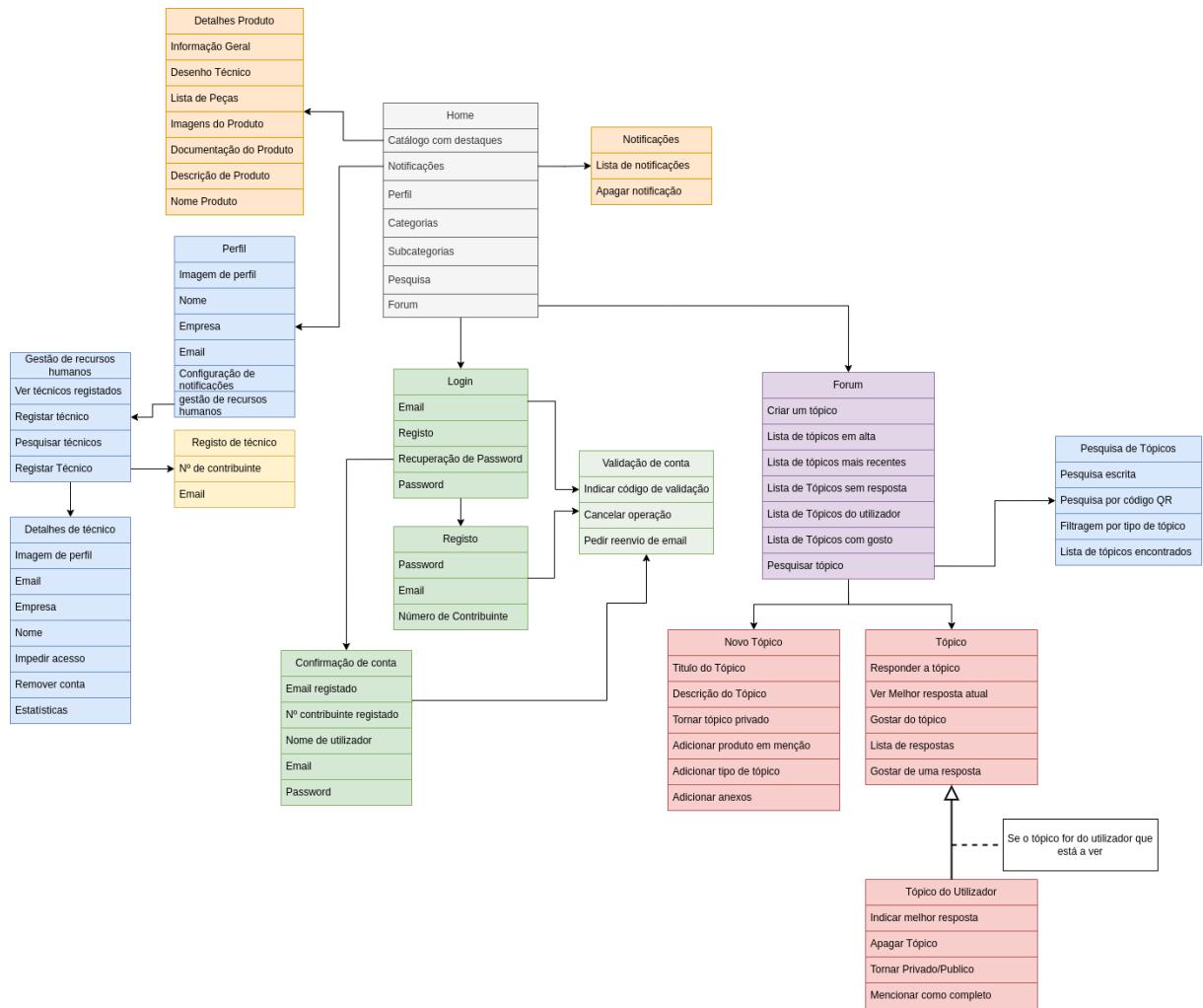


Figura 3.2: Esquema de organização de páginas do software

3.9.1 Autenticação e página Inicial

Através deste esquema é possível perceber que através do ecrã principal o utilizador tem acesso ao catálogo de produtos e ao fórum, neste o utilizador pode também realizar o login e o logout que redirecionam para os respetivos ecrãs.

O ecrã de login é então necessário o utilizador indicar o número de contribuinte e a password, neste ele pode também pedir recuperação de password e/ou redirecionar para o registo onde necessitará de número de contribuinte, password e email para o realizar.

Em caso de o utilizador não possuir a sua conta ativa, este será encaminhado para o ecrã de validar conta em que poderá indicar o código de validação, cancelar a operação voltando para o ecrã anterior e pedir o reenvio do código de validação.

Em caso de se tratar de um técnico que necessita de confirmar a sua conta, este conseguirá ver as informações registadas e introduzir o seu nome, alterar o seu email e password.

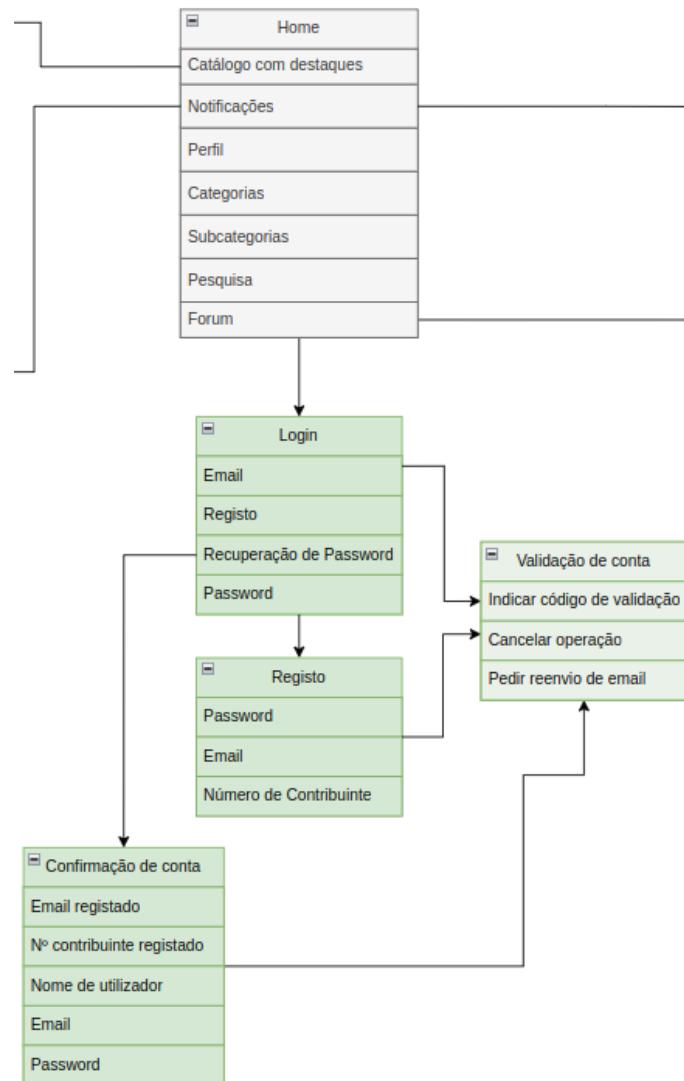


Figura 3.3: Esquema de organização das páginas de autenticação e página inicial

3.9.2 Fórum

Através do ecrã inicial o utilizador pode se dirigir para o ecrã de fórum, neste ecrã ele poderá pesquisar por tópicos, ou então aceder a tópicos em alta, tópicos mais recentes, tópicos sem resposta e pesquisar tópicos. O técnico consegue também ver os seus tópicos e criar tópicos.

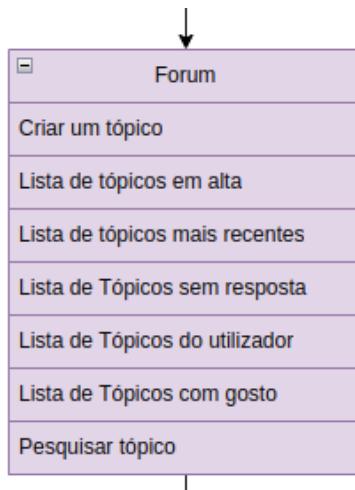


Figura 3.4: Esquema de organização da página de fórum

3.9.3 Criar novo tópico

Quando o técnico decide criar um tópico ele tem de indicar o título e a descrição do seu tópico, de seguida poderá indicar se o tópico é privado ou não, indicar o tipo de tópico, o produto referente e adicionar anexos.

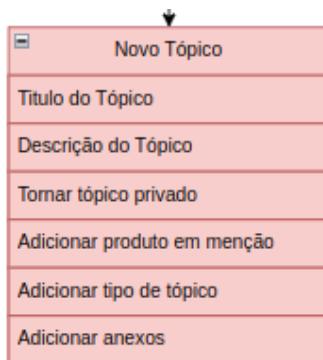


Figura 3.5: Esquema de organização da pagina de criação de tópico

3.9.4 Detalhes de tópico

O utilizador pode também ver os detalhes do tópico, já o técnico pode também responder a um tópico, gostar do tópico, gostar de uma resposta. Caso este tópico seja do mesmo, este pode indicar a melhor resposta, apagar o tópico, tornar o tópico público ou privado e mencionar como completo.

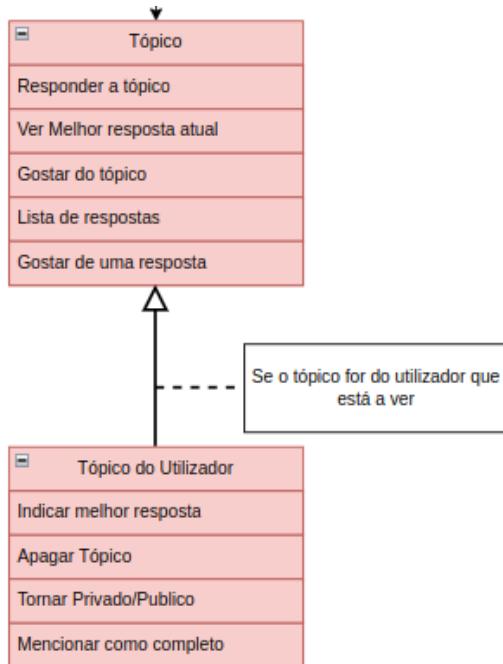


Figura 3.6: Esquema de organização da página de detalhes de tópico

3.9.5 Pesquisa de tópicos

A página de pesquisa permite ao utilizador pesquisar por tópicos específicos tanto pesquisando por nome como por código QR de um produto. Para além da pesquisa o utilizador pode também realizar a filtragem dos tópicos por tipo.

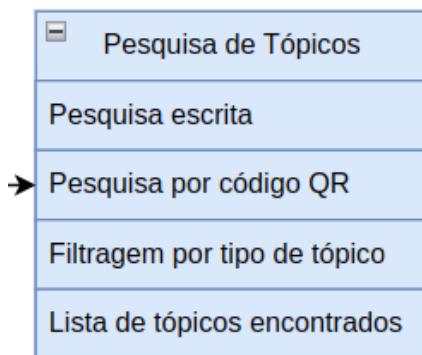


Figura 3.7: Esquema de organização da página de pesquisa de tópicos

3.9.6 Notificações

A página de notificações permite ao técnico visualizar as suas notificações, assim como também apagar estas.

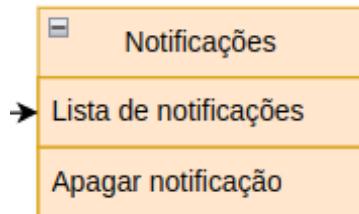


Figura 3.8: Esquema de organização da página de notificações

3.9.7 Perfil

A página de perfil de técnico permite a este visualizar as suas informações, assim como alterar o seu email e configurar as notificações do mesmo. Caso se trate de uma empresa a visualizar o seu perfil, esta poderá ter acesso à gestão de recursos humanos, para gerir os seus técnicos.

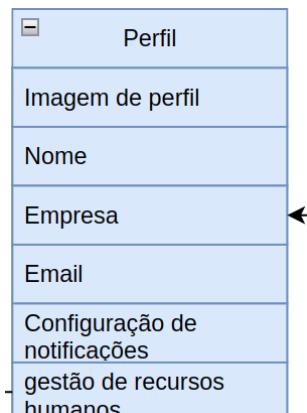


Figura 3.9: Esquema de organização da página de notificações

3.9.8 Gestão de recursos humanos

A página de perfil gestão de recursos humanos permite à empresa gerir todos os seus técnicos registados e criar contas de técnicos. Assim que a empresa seleciona um técnico, esta verá o seu perfil, com estatísticas do técnico

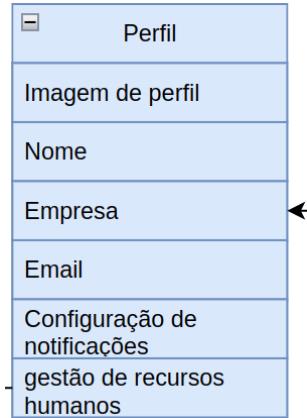


Figura 3.10: Esquema de organização da página de gestão de recursos humanos

3.10 Histórias de Utilizador

Antes de desenvolver os casos de uso foram criadas histórias de utilizador para ser possível descrever os objetivos do utilizador ao realizar uma ação.

Tabela 3.3: Tabela de histórias de utilizador

#	Ator	Descrição
Autenticação		
US01	Utilizador	Eu como Utilizador, quero conseguir utilizar a aplicação sem realizar o login
US02	Empresa	Eu como Empresa, quero conseguir realizar o registo na aplicação
US03	Técnico	Eu como Técnico, quero conseguir realizar o login na aplicação utilizando o número de contribuinte e password
US04	Técnico	Eu como Técnico, quero conseguir pedir reenvio de código de ativação de conta caso eu não receba o código
US05	Técnico	Eu como Técnico, quero conseguir ser identificado como tal na aplicação
Fórum		
US06	Técnico	Eu como Técnico, quero conseguir acessar ao fórum
US07	Técnico	Eu como Técnico, quero conseguir visualizar os tópicos mais recentes, de forma a conseguir ver os mais falados no dia atual
US08	Técnico	Eu como Técnico, quero conseguir visualizar os tópicos em destaque, de forma a ver quais são mais falados
US09	Técnico	Eu como Técnico, quero conseguir ver os meus tópicos de forma a conseguir aceder a estes facilmente
US10	Técnico	Eu como Técnico, quero conseguir visualizar os tópicos por responder, de forma a conseguir ajudar alguém com maior facilidade
US11	Técnico	Eu como Técnico, quero conseguir visualizar os tópicos privados
US12	Técnico	Eu como Técnico, quero conseguir realizar filtragem de tópicos por tipo
US13	Técnico	Eu como Técnico, quero conseguir pesquisar por um tópico relativo a um assunto de forma a obter a solução
US14	Técnico	Eu como Técnico, quero conseguir pesquisar por um tópico relativo a um produto de forma a encontrar questões comuns a este
US15	Técnico	Eu como Técnico, quero conseguir pesquisar por código QR de um produto de forma a encontrar tópicos referentes ao mesmo mais facilmente

Continued on next page

Tabela 3.3: Tabela de histórias de utilizador (Continued)

	Criar Tópico	
US16	Técnico	Eu como Técnico, quero conseguir criar tópicos de forma a conseguir expor questões
US17	Técnico	Eu como Técnico, quero conseguir indicar se o meu tópico é publico ou privado, de forma a conseguir respostas de qualquer cliente, ou apenas de técnicos
US18	Técnico	Eu como Técnico, quero conseguir indicar o tipo de tópico em que o este se enquadrar de forma a facilitar a sua identificação
US19	Técnico	Eu como Técnico, quero conseguir indicar o produto referente ao tópico para facilitar a identificação do mesmo
US20	Técnico	Eu como Técnico, quero conseguir anexar imagens ao tópico de forma a facilitar a comunicação e identificação do problema
	Gestão de Tópico	
US21	Técnico	Eu como Técnico quero conseguir indicar qual a melhor resposta ao meu tópico de forma a facilitar o encontro da solução do problema a outros clientes ou técnicos com o mesmo problema
US22	Técnico	Eu como Técnico quero conseguir indicar que o tópico se encontra finalizado quando o problema está resolvido
US23	Técnico	Eu como Técnico quero conseguir remover o meu tópico
US24	Técnico	Eu como Técnico quero conseguir alterar a visibilidade do meu tópico
	Tópicos	
US25	Técnico	Eu como Técnico, quero conseguir ver todas as respostas a um tópico
US26	Técnico	Eu como Técnico, quero conseguir gostar de um tópico caso o ache relevante
US27	Técnico	Eu como Técnico, quero conseguir apagar uma resposta minha
	Respostas a Tópicos	
US28	Técnico	Eu como Técnico, quero conseguir comentar um tópico de forma a dar a minha resposta
US29	Técnico	Eu como Técnico, quero conseguir responder a um comentário de forma a comunicar
US30	Técnico	Eu como Técnico, quero conseguir anexar imagens ao meu comentário
US31	Técnico	Eu como Técnico, quero conseguir gostar de um comentário caso ache este relevante

Continued on next page

Tabela 3.3: Tabela de histórias de utilizador (Continued)

		Perfil	
US32	Técnico	Eu como Técnico quero conseguir alterar o meu email	
US33	Técnico	Eu como Técnico quero conseguir alterar a minha imagem de perfil	
US34	Técnico	Eu como Técnico quero conseguir alterar o meu nome	
		Notificações	
US35	Técnico	Eu como Técnico quero conseguir receber notificações por email e/ou push de forma a manter-me atualizado das minhas questões	
US36	Técnico	Eu como Técnico quero conseguir alterar o tipo de notificação que recebo entre relatório diário e tempo real	
US37	Técnico	Eu como Técnico quero conseguir apagar as minhas notificações de forma a evitar aglomeração	
		Gestão de recursos humanos	
US38	Empresa	Eu como Empresa quero conseguir criar conta para os meus técnicos utilizarem o fórum	
US39	Empresa	Eu como Empresa quero conseguir impedir acesso a uma conta de técnico	
US40	Empresa	Eu como Empresa quero conseguir remover uma conta de técnico em caso de este já não pertencer à empresa	

3.11 Casos de uso

De forma a transformar as histórias de utilizador em ações e especificar todas as ações dos atores do software e todas as reações do sistema com o qual o ator interage foram desenvolvidos casos de uso.

Tabela 3.4: Tabela de casos de uso

#	User Story	Autor	Nome	Descrição
Criar tópico				
UC 1.0	US15	Técnico	Criar novo tópico	Criação de um novo tópico no fórum
Pesquisa de tópicos				
UC 1.1	US11	Técnico	Pesquisar tópicos específicos	Pesquisar por tópicos no fórum
UC 1.1.1	US12 e US13	Técnico	Pesquisa escrita	Pesquisar tópicos por assunto
UC 1.1.2	US14	Técnico	Pesquisa por código QR	Pesquisar tópicos referentes a um produto
Listagens de tópicos				
UC 1.2	US05	Técnico	Ver tópicos	Ver listagens de tópicos do fórum
Detalhes de tópico				
UC 1.3	US08	Técnico	Selecionar tópico	Ver detalhes de um tópico selecionado
UC 1.3.1	US21	Técnico	Finalizar tópico	Finalizar um tópico para indicar que está respondido
UC 1.3.2	US20	Técnico	Selecionar melhor resposta	Selecionar a melhor resposta do tópico
UC 1.3.3	US22	Técnico	Eliminar tópico	Eliminar um tópico do fórum
UC 1.3.4	US23	Técnico	Alterar visibilidade do tópico	Alterar a visibilidade de um tópico entre público e privado
UC 1.3.5	US28	Técnico	Comentar o tópico	Comentar um tópico
UC 1.3.6	US25	Técnico	Gostar de tópico	Gostar de um tópico
Comentários				
UC 1.3.7	US24	Técnico	Ver comentários	Ver comentários do tópico
UC 1.3.7.1	US27	Técnico	Apagar comentário	Apagar comentário de um tópico

Continued on next page

Tabela 3.4: Tabela de casos de uso (Continued)

UC 1.3.7.2	US29	Técnico	Responder a comentário	Responder a um comentário de um tópico
UC 1.3.7.3	US31	Técnico	Gostar de comentário	Gostar de um comentário
Ativação de conta				
UC 1.4	-	Técnico	Ativação de conta	Ativar conta de cliente
UC 1.4.1	US04	Técnico	Pedir reenvio de código de ativação	Pedir reenvio de email de código de ativação
Perfil				
UC 1.5	US31 - US32 - US33	Técnico	Ver Perfil	Ver perfil de utilizador
Notificações				
UC1.6	US34 e US36	Técnico	Ver notificações	Ver todas as notificações
UC1.7	US34 e US35	Técnico	Configuração de notificações	Configurar o modo e tipo de notificações a receber
Gestão de recursos humanos				
UC1.8	US37	Empresa	Registar Técnico	Registar conta de técnico da empresa
UC1.9	US38	Empresa	Impedir acesso a técnico	Registar conta de técnico da empresa
UC1.10	US39	Empresa	Remover conta de técnico	Registar conta de técnico da empresa

3.11.1 Especificação de casos de uso

De forma a demonstrar todas as interações entre os atores e o sistema, assim como também todas as ações destes e fluxos possíveis, foram realizadas especificações de casos de uso.

3.11.1.1 Especificação de caso de uso de listagem de tópicos

Através da listagem de tópicos é possível visualizar todas as listagens de tópicos que o utilizador poderá visualizar, sendo que o técnico consegue para além destas listagens, ver os seus tópicos e ver os tópicos privados.

Tabela 3.5: Tabela de especificação de caso de uso de listagem de tópicos do técnico

Caso de Uso	Ver listagem de tópicos	
Descrição	Ver a listagem de tópicos existentes no fórum por diversas categorias	
Autor	Técnico	
Pré-condição	-	
Pós-condição	-	
	Ator	Sistema
Fluxo Principal	1-Ver tópicos populares	
		2-Lista de tópicos populares
Fluxo Alternativo(A1)	1-Ver tópicos mais recentes	
		2-Lista de tópicos mais recentes
Fluxo Alternativo(A2)	1-Ver meus tópicos	
		2-Lista de tópicos do técnico
Fluxo Alternativo(A3)	1-Ver tópicos por responder	
		2-Lista de tópicos por responder

3.11.1.2 Especificação de caso de uso de criar novo tópico

Aquando a criação de um tópico um técnico poderá realizar diversas ações sendo que obrigatoriamente terá sempre de indicar o título, descrição do tópico e tipo de tópico, para além desta informação o técnico poderá também anexar imagens, referenciar um produto e indicar a visibilidade.

Tabela 3.6: Tabela de especificação de caso de uso login

Caso de Uso	Criar novo tópico	
Descrição	Criação de um novo tópico no fórum	
Autor	Técnico	
Pré-condição	Clicar em adicionar novo tópico	
Pós-condição	-	
	Ator	Sistema
Fluxo Principal	1-Indicar o título do tópico	
	2-Indicar a descrição do tópico	
	3-Indicar se o tópico é privado	
	4-Indicar o tipo do tópico	
	5-Indicar o produto referido no tópico	
	6-Adicionar imagens de anexo	
	7-Confirmar a criação do tópico	
		8-Verificar se título está inserido
		9-Verificar se descrição está inserida
		10-Inserir novo tópico no fórum
Fluxo Alternativo(A1)	1-Cancelar a criação do tópico	
Fluxo Alternativo(A2)	1-Indicar o título do tópico	
	2-Indicar se o tópico é privado	
	3-Confirmar a criação do tópico	
		4-Verificar se título está inserido
		5-Verificar se descrição está inserida
		6-Erro descrição em falta

3.11.1.3 Especificação de caso de uso de pesquisar tópicos por escrito

Assim que um utilizador deseje pesquisar por um assunto específico de tópico este poderá realizar uma pesquisa escrita onde conseguirá realizar filtragem por tipo de tópico também.

Tabela 3.7: Tabela de especificação de caso de uso de pesquisa por escrito

Caso de Uso	Pesquisa por escrita	
Descrição	Pesquisar por tópicos no fórum	
Autor	Utilizador	
Pré-condição	Selecionar pesquisa de fórum	
Pós-condição	-	
	Autor	Sistema
Fluxo Principal	1-Pesquisar assunto	
		2-Lista de tópicos do assunto
	3-Filtrar por tipo	3-Filtragem de tópicos do assunto por tipo
Fluxo Alternativo(A1)	1-Pesquisar assunto	
		2-Lista de tópicos do assunto

3.11.1.4 Especificação de caso de uso de ver finalizar tópico

Quando um técnico já se encontra satisfeito com a solução do problema este poderá indicar que o tópico se encontra finalizado sinalizando então para outros técnicos que o tópico tem solução.

Tabela 3.8: Tabela de especificação de caso de uso de finalizar tópico

Caso de Uso	Finalizar tópico	
Descrição	Finalizar um tópico para indicar que está respondido	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Alterar tópico para finalizado	
	Autor	Sistema
Fluxo Principal	1-Clicar em finalizar tópico	
		2-Alterar tópico para finalizado
Fluxo Alternativo(A1)	-	-

3.11.1.5 Especificação de caso de uso de selecionar melhor resposta

Sempre que o técnico encontrar uma resposta no seu tópico que se destaca na solução da sua questão, este poderá colocar esta resposta como melhor resposta do tópico, caso já exista uma melhor resposta no tópico, esta é removida de melhor resposta e a nova resposta é colocada como melhor resposta.

Tabela 3.9: Tabela de especificação de caso de uso de selecionar melhor resposta

Caso de Uso	Selecionar melhor resposta	
Descrição	Selecionar a melhor resposta do tópico	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Alterar a resposta para melhor resposta	
	Autor	Sistema
Fluxo Principal	1-Clicar em melhor resposta	
		2-Verificar se já existe uma melhor resposta - Não
		3- Colocar a resposta como melhor resposta do tópico
Fluxo Alternativo(A1)	1-Clicar em melhor resposta	
		2-Verificar se já existe uma melhor resposta - Sim
		3- Verificar se a resposta existente é a mesma selecionada - Não
		4-Alterar melhor resposta
Fluxo Alternativo(A2)	1-Clicar em melhor resposta	
		2-Verificar se já existe uma melhor resposta - Sim
		3- Verificar se a resposta existente é a mesma selecionada - Sim
		4-Remover melhor resposta

3.11.1.6 Especificação de caso de uso de eliminar tópico

O técnico sempre que desejar poderá eliminar o tópico, conseguindo assim remover este tópico do fórum, não sendo este novamente mostrado.

Tabela 3.10: Tabela de especificação do caso de uso de eliminar tópico

Caso de Uso	Eliminar tópico	
Descrição	Eliminar um tópico do fórum	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Remoção do tópico	
	Autor	Sistema
Fluxo Principal	1-Clicar em remover tópico	
		3-Remover o tópico
Fluxo Alternativo(A1)	-	-

3.11.1.7 Especificação de caso de uso de alterar visibilidade de um tópico

Quando um técnico publica um tópico este pode desejar alterar a sua visibilidade para apenas técnicos ou todos os utilizadores conseguirem ver.

Tabela 3.11: Tabela de especificação de caso de uso de alteração de visibilidade de um tópico

Caso de Uso	Alterar visibilidade do tópico	
Descrição	Alterar a visibilidade de um tópico entre público e privado	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Alterar visibilidade do tópico	
	Autor	Sistema
Fluxo Principal	1-Clicar em alterar visibilidade	
		2-Inverter visibilidade do tópico
Fluxo Alternativo(A1)	-	-

3.11.1.8 Especificação de caso de uso gostar de um tópico

O técnico sempre que encontra um tópico que identifica como útil, este poderá gostar o tópico dando assim destaque a este.

Tabela 3.12: Tabela de especificação de caso de uso de gostar de um tópico

Caso de Uso	Gostar do tópico	
Descrição	Gostar de um tópico	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Alterar gostos do tópico	
	Autor	Sistema
Fluxo Principal	1-Clicar em gosto	
		2-Verificar se o gosto já existe - Não
		3-Aumentar gosto ao tópico
Fluxo Alternativo(A1)	1-Clicar em gosto	
		2-Verificar se o gosto já existe - Sim
		3-Remover gosto do tópico

3.11.1.9 Especificação de caso de uso gostar de um comentário

Sempre que um técnico identificar um comentário como útil este poderá gostar dos comentários, dando assim destaque a este.

Tabela 3.13: Tabela de especificação de caso de uso de gostar de comentário

Caso de Uso	Gostar de comentário	
Descrição	Gostar de um comentário	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	-	
	Autor	Sistema
Fluxo Principal	1-Clicar em gosto	
		2-Verificar se o gosto já existe - Não
		3-Aumentar gosto ao comentário
Fluxo Alternativo(A1)	1-Clicar em gosto	
		2-Verificar se o gosto já existe - Sim
		3-Remover gosto do tópico

3.11.1.10 Especificação de caso de uso de comentar tópico

Sempre que um técnico encontra um tópico que sobre uma questão que poderá ajudar este consegue responder a um tópico, criando assim um comentário, onde este poderá anexar também imagens.

Tabela 3.14: Tabela de especificação de caso de uso de comentar um tópico

Caso de Uso	Comentar o tópico	
Descrição	Comentar um tópico	
Autor	Técnico	
Pré-condição	Clicar no tópico desejado	
Pós-condição	Inserir a resposta no tópico	
	Ator	Sistema
Fluxo Principal	1-Indicar a descrição da resposta	
	2-Anexar Imagem	
	3-Confirmar a resposta	
		4-Inserir novo comentário no tópico
Fluxo Alternativo(A1)	1-Indicar a descrição da resposta	
	2-Confirmar a resposta	
		3-Inserir novo comentário no tópico
Fluxo Alternativo(A2)	1-Cancelar a criação do tópico	

3.11.1.11 Especificação de caso de uso ativar conta

O técnico de forma a ativar a sua conta deverá introduzir o código de ativação de conta correto, caso contrário não será possível ativar a sua conta.

Tabela 3.15: Tabela de especificação de caso de uso ativação de conta

Caso de Uso	Ativar conta	
Descrição	Ativar conta de aplicação	
Autor	Técnico	
Pré-condição	-	
Pós-condição	-	
	Autor	Sistema
Fluxo Principal	1-Inserir código de validação	
	2-Validar conta	
		3-Verificar se o código está correto-Sim
		4-Validar conta
Fluxo Alternativo(A1)	1-Cancelar Ativação de conta	
Fluxo Alternativo(A2)	1-Inserir código de validação	
	2-Validar conta	
		3-Verificar se o código está correto-Não
		4-Código de validação incorreto

3.11.1.12 Especificação de caso de uso configurar notificações

As notificações da aplicação poderão ser personalizadas, sendo possível escolher entre email, push e ambos, para além destas configurações, é também possível personalizar o tipo de notificação para cada método, seja relatório diário de todas as notificações ou notificações em tempo real.

Tabela 3.16: Tabela de especificação de caso de uso de configuração de notificações

Caso de Uso	Configuração de notificações	
Descrição	Configuração de notificações do técnico	
Autor	Técnico	
Pré-condição	-	
Pós-condição	-	
	Ator	Sistema
Fluxo Principal	1-Indicar preferência de receção de notificações	
	2-Indicar tipo de receção de notificações	
Fluxo Alternativo(A1)	1-Ver notificações	

3.11.1.13 Especificação de caso de uso registar técnico

Sempre que uma empresa deseja realizar o registo de técnicos em seu nome, esta poderá indicar o nº contribuinte do mesmo e email, com isto este receberá um email para confirmar o registo de conta.

Tabela 3.17: Tabela de especificação de caso de uso de registar técnico

Caso de Uso	Registar técnico	
Descrição	Registar conta de técnico da empresa	
Autor	Empresa	
Pré-condição	-	
Pós-condição	-	
	Ator	Sistema
Fluxo Principal	1-Indicar o nº contribuinte	
	2-Indicar email	
		3-Registrar técnico
Fluxo Alternativo(A1)	-	-

3.11.2 Diagramas de casos de uso

De forma a ser possível visualizar graficamente todas as ações que os atores conseguem realizar e para melhorar a comunicação com as partes interessadas do projeto, foram desenvolvidos diagramas de casos de uso.

3.11.2.1 Casos de uso Fórum

Na imagem representada abaixo (Figura 3.11) é possível visualizar o diagrama de casos de uso para o fórum. Neste, o utilizador poderá ver as listagens de tópicos em destaque, tópicos mais recentes, caso seja um técnico poderá também ver os tópicos privados e os seus tópicos. O utilizador poderá também pesquisar por tópicos, dos quais ele poderá selecionar um tópico específico para visualizar. O técnico poderá, além disso, criar um novo tópico onde se dirigirá para a criação de tópicos, nesta o técnico será obrigado a inserir um título, descrição e tipo do tópico para o criar, mas também poderá inserir imagens e indicar produto referente. Para finalizar a criação o técnico poderá confirmar ou cancelar o processo.

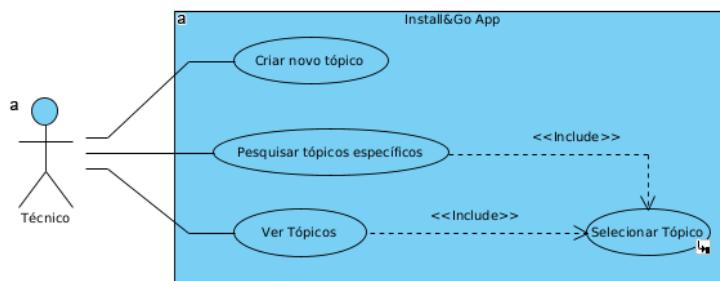


Figura 3.11: Diagrama de casos de uso de fórum

3.11.2.2 Casos de uso de pesquisar tópicos

O utilizador poderá realizar a pesquisa por tópicos específicos, esta pesquisa poderá ser realizada por escrito onde o utilizador indica o assunto a pesquisar, esta poderá ser filtrada. O utilizador poderá também pesquisar por tópicos específicos referentes a um produto, no qual o utilizador lê o código QR do produto, sendo este pesquisado no servidor Motorline, e de seguida poderá realizar uma pesquisa por escrito, sendo listado apenas tópicos com o produto em questão referenciado.

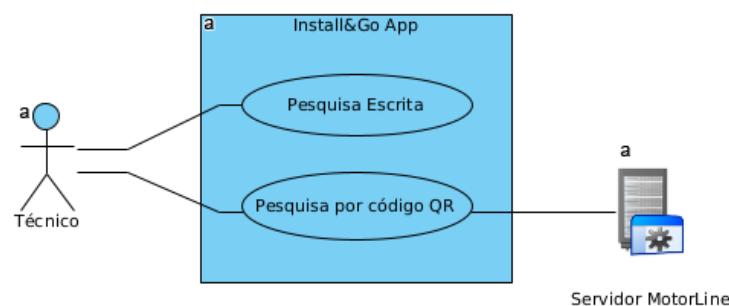


Figura 3.12: Diagrama de casos de uso de pesquisa de tópicos

3.11.2.3 Casos de uso ver detalhes de tópico

Assim que um utilizador seleciona um tópico é direcionado para os detalhes de tópico, onde apenas consegue visualizar o tópico e os comentários do mesmo.

Já o técnico consegue também responder ao tópico e, caso seja do mesmo, este consegue finalizar o tópico, selecionar a melhor resposta e remover, eliminar e alterar a visibilidade do tópico.

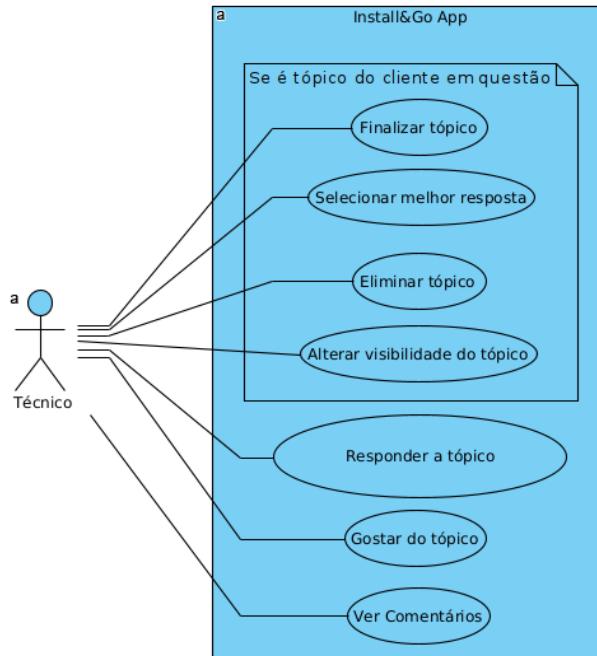


Figura 3.13: Diagrama de casos de uso de detalhes de tópico

3.11.2.4 Casos de uso ver comentários

O técnico quando decide visualizar os comentários consegue responder a um comentário e gostar de uma resposta ou comentário, caso este seja seu ainda o consegue apagar.

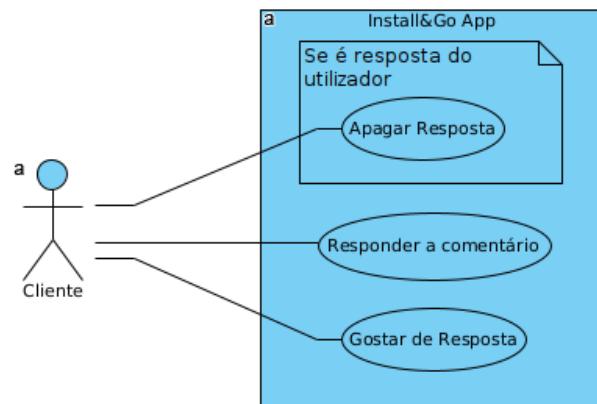


Figura 3.14: Diagrama de casos de uso de ver comentários

3.11.2.5 Casos de uso ativação de conta

Assim que uma conta é confirmada, um email de ativação é enviado para técnico e esta deverá ser ativada, para isto o código deverá ser indicado pelo técnico para se proceder à ativação da conta. Este poderá, em caso de necessidade, pedir o reenvio do código de ativação, o qual será gerado novamente e reenviado.

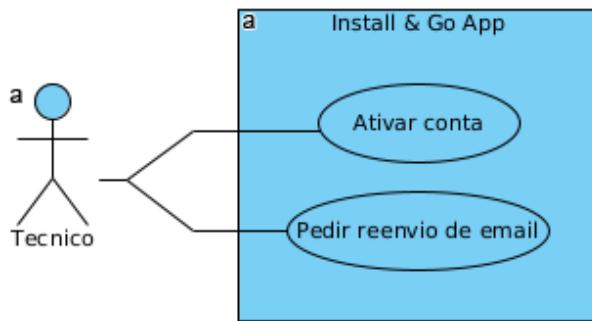


Figura 3.15: Diagrama de casos de uso de ativação de conta

3.11.2.6 Casos de uso perfil

Sempre que o técnico desejar alterar alguma informação sua, poderá alterar o seu nome, email e imagem de perfil.

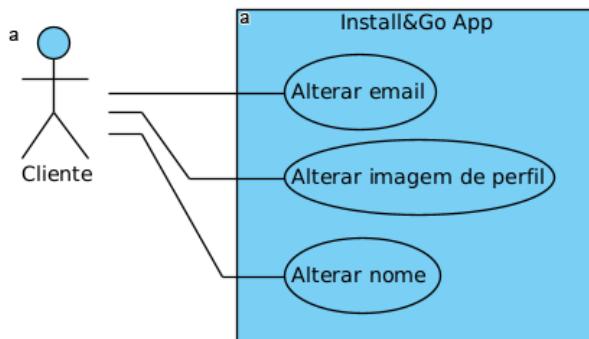


Figura 3.16: Diagrama de casos de uso de ativação de conta

3.11.2.7 Casos de uso notificações

Sempre que o técnico desejar ver as suas notificações, este poderá selecionar as suas notificações, este terá também a possibilidade de alterar a configuração das suas notificações, para apenas as receber por email ou push ou então ambas e poderá também personalizar cada método para receber relatório diário de notificações ou então notificações em tempo real.

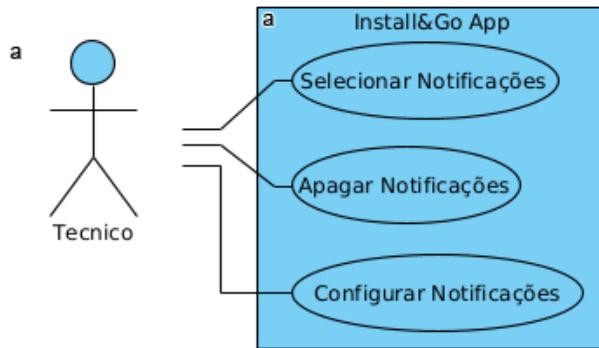


Figura 3.17: Diagrama de casos de uso de ativação de conta

3.11.2.8 Casos de uso gestão de recursos humanos

Uma empresa poderá registar contas para os seus técnicos em seu nome, indicando o seu email e nº contribuinte, esta poderá também impedir acesso a estas contas ou remover completamente a conta da aplicação.

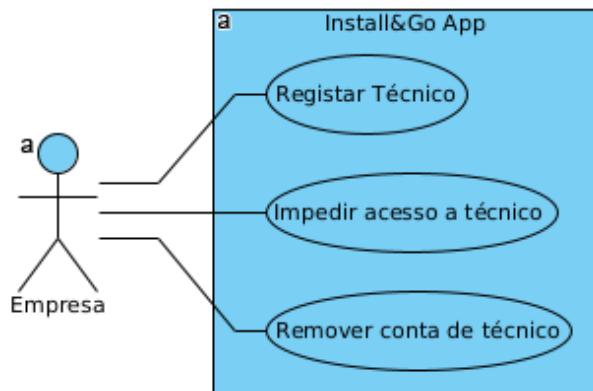


Figura 3.18: Diagrama de casos de uso de ativação de conta

3.12 Diagrama Entidade Relação

O software Install&Go é suportado por uma base de dados relacional, sendo esta esquematizada tendo por base as necessidades do projeto.

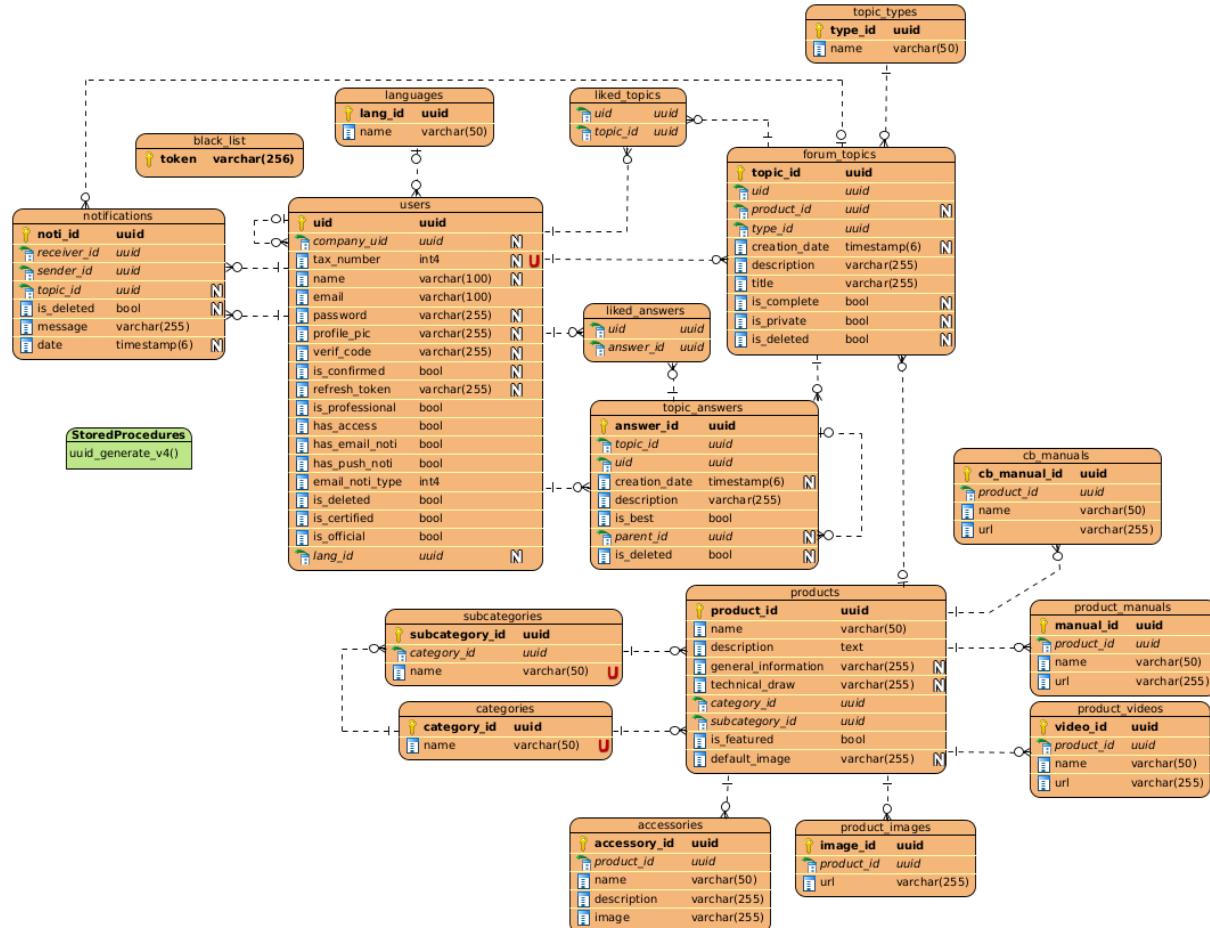


Figura 3.19: Diagrama Entidade Relação base de dados Install&Go

3.12.1 Escolha de diagrama de entidade relação

Durante o desenvolvimento do diagrama de entidade relação surgiu a opção de se separar as empresas dos seus técnicos em duas tabelas como exemplificado na Figura 3.20. O diagrama anteriormente mostrado foi escolhido devido à sua simplicidade, visto que no diagrama representado na Figura 3.20 sempre que se deseja, por exemplo, obter o utilizador que criou um tópico teria de se verificar se o uid contido é de uma empresa ou de um técnico e apenas de seguida se obter o utilizador que criou o tópico, sendo este um exemplo entre os demais do mesmo tipo. Tendo em conta este problema foi decidido optar pelo diagrama anteriormente demonstrado.

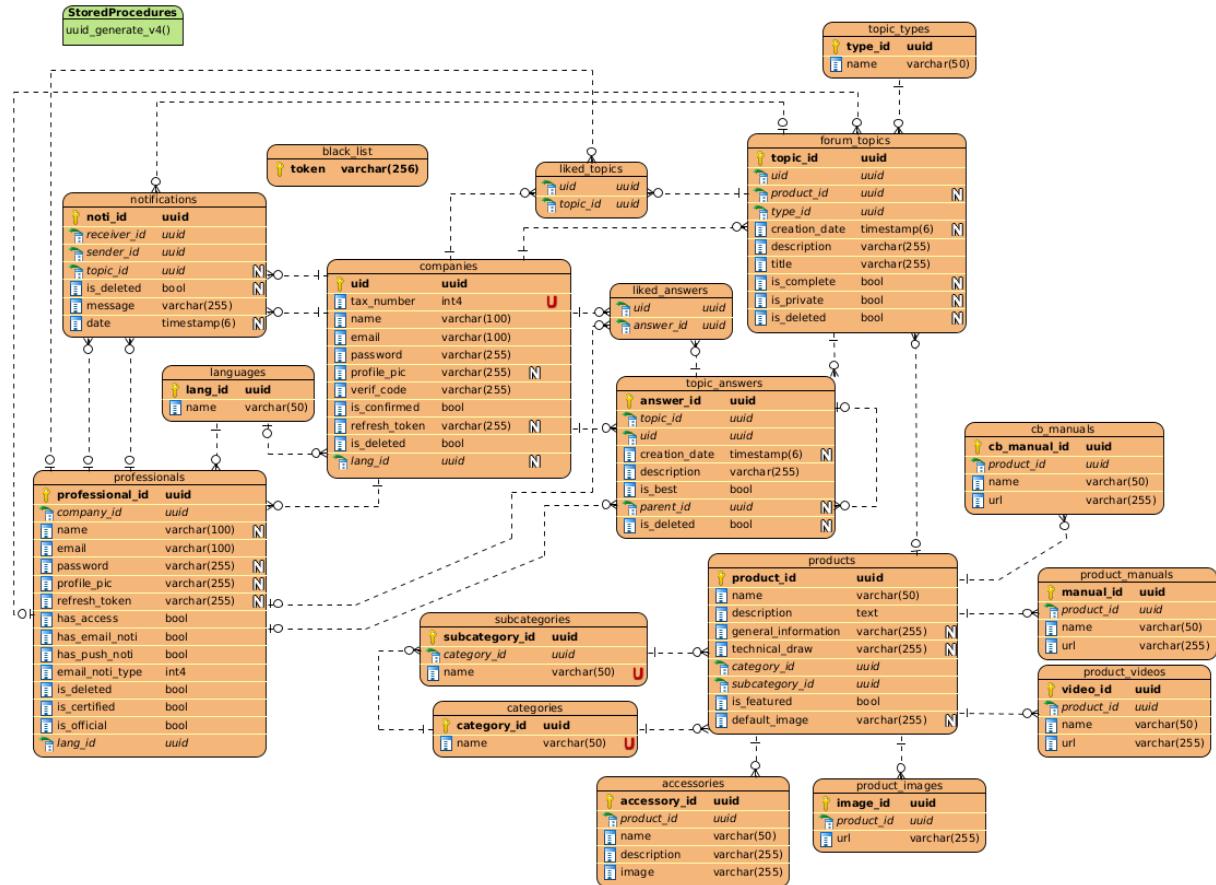


Figura 3.20: Diagrama Entidade Relação alternativo

3.12.2 Dicionário de termos

De forma a ser possível entender o propósito de cada tabela e atributo foi então criado um dicionário de termos para a base de dados(Tabela X).

Tabela 3.18: Dicionário de termos da base de dados

Tabela	Descrição	Atributos	Descrição
users	Tabela encarregue de guardar todos os dados referentes aos utilizadores da aplicação	uid	Id do utilizador
		company_id	Id da empresa referente ao técnico
		tax_number	Número de contribuinte do utilizador
		name	Nome do utilizador
		email	Email do utilizador
		password	Password do utilizador
		profile_pic	Imagen de perfil do utilizador
		verif_code	Código de verificação do utilizador
		is_confirmed	Verificação de se o código está confirmado
		refresh_token	Token de refresh
		is_professional	Verificação se é profissional
		has_access	Verificação se tem acesso à conta
		has_email_noti	Verificação se ativou notificações de email
		has_push_noti	Verificação se ativou notificações push
		email_noti_type	Tipo de notificação de email
		push_noti_type	Tipo de notificação push
		is_deleted	Verificação se a conta se encontra apagada
		is_certified	Verificação se é um técnico certificado
		is_official	Verificação se é um técnico oficial

Continued on next page

Tabela 3.18: Dicionário de termos da base de dados (Continued)

		lang_id	Id da linguagem preferencial do utilizador
black_list	Tabela que guarda os tokens a bloquear	token	Token a bloquear
liked_topics	Tabela encarregue de guardar todos os tópicos que foram gostados pelo utilizador	uid	Id do utilizador
		topic_id	Id do tópico
liked_answers	Tabela encarregue de guardar todas as respostas que receberam gosto do utilizador	uid	Id do utilizador
		answer_id	Id da resposta
topic_types	Tabela encarregue de guardar os tipos de tópico existentes	type_id	Id do tipo de tópico
		name	nome do tipo de tópico
notifications	Tabela encarregue de guardar todas as notificações do técnico	noti_id	Id da notificação
		receiver_id	Receptor da notificação
		sender_id	Emissor da notificação
		topic_id	Id do tópico em caso de estar referente a um tópico
		is_deleted	Verificação se a notificação está apagada
		message	Mensagem da notificação
		date	Data de emissão da notificação

Continued on next page

Tabela 3.18: Dicionário de termos da base de dados (Continued)

forum_topics	Tabela encarregue de guardar todos os tópicos existentes na aplicação	topic_id	Id do tópico
		uid	Id do dono do tópico
		product_id	Produto referente ao tópico
		ttype_id	Id do tipo referente ao tópico
		creation_date	Data de criação do tópico
		description	Descrição do tópico
		title	Titulo do tópico
		is_complete	Verificação se o tópico está finalizado
		is_private	Verificação se o tópico é privado
		is_deleted	Verificação se o tópico está apagado
topic_answers	Tabela encarregue de guardar todas as respostas a um tópico	answer_id	Id da resposta
		topic_id	Id do tópico
		uid	Id do dono da resposta
		creation_date	Data de criação da resposta
		description	Descrição da resposta
		is_best	Verificação se é a melhor resposta
		parent_id	Id da resposta pai
		is_deleted	Verificação se o topico se encontra apagado
categories	Tabela encarregue de guardar todas as categorias de produtos existentes	category_id	Id da categoria
		name	Nome da categoria
subcategories	Tabela encarregue de guardar as subcategorias de produtos existentes	subcategory_id	Id da subcategoria
		category_id	Id da categoria
		name	Nome da subcategoria

Continued on next page

Tabela 3.18: Dicionário de termos da base de dados (Continued)

cb_manuals	Tabela encarregue de guardar os manuais de utilização das placas de controlo	cb_manual_id	Id do manual
		product_id	Id do produto
		name	Nome da placa de controlo
		url	Url do manual
products	Tabela encarregue de guardar as informações dos produtos do catálogo da empresa	product_id	Id do produto
		name	Nome do produto
		description	Descrição do produto
		general_information	Url da informação geral do produto
		technical_draw	Url do desenho técnico do produto
		category_id	Id da categoria de produto
		subcategory_id	Id da subcategoria de produto
		is_featured	Verificação se o produto é um destaque
		default_image	Imagen do produto por omissão
product_manuals	Tabela encarregue de guardar os manuais de utilização dos produtos	cb_manual_id	Id do manual
		product_id	Id do produto
		name	Nome do manual
		url	Url do manual
product_videos	Tabela encarregue de guardar os videos de cada produto	cb_manual_id	Id do video
		product_id	Id do produto
		name	Nome do video
		url	Url do video
languages	Tabela encarregue de guardar as linguagens suportadas pela aplicação	lang_id	Id da linguagem
		name	Nome da linguagem

3.13 Diagrama de Classes

De forma a ser possível prever e organizar o software foi desenvolvido um diagrama de classes que permite visualizar cada classe que se espera conter no software, assim como também os seus atributos e métodos.

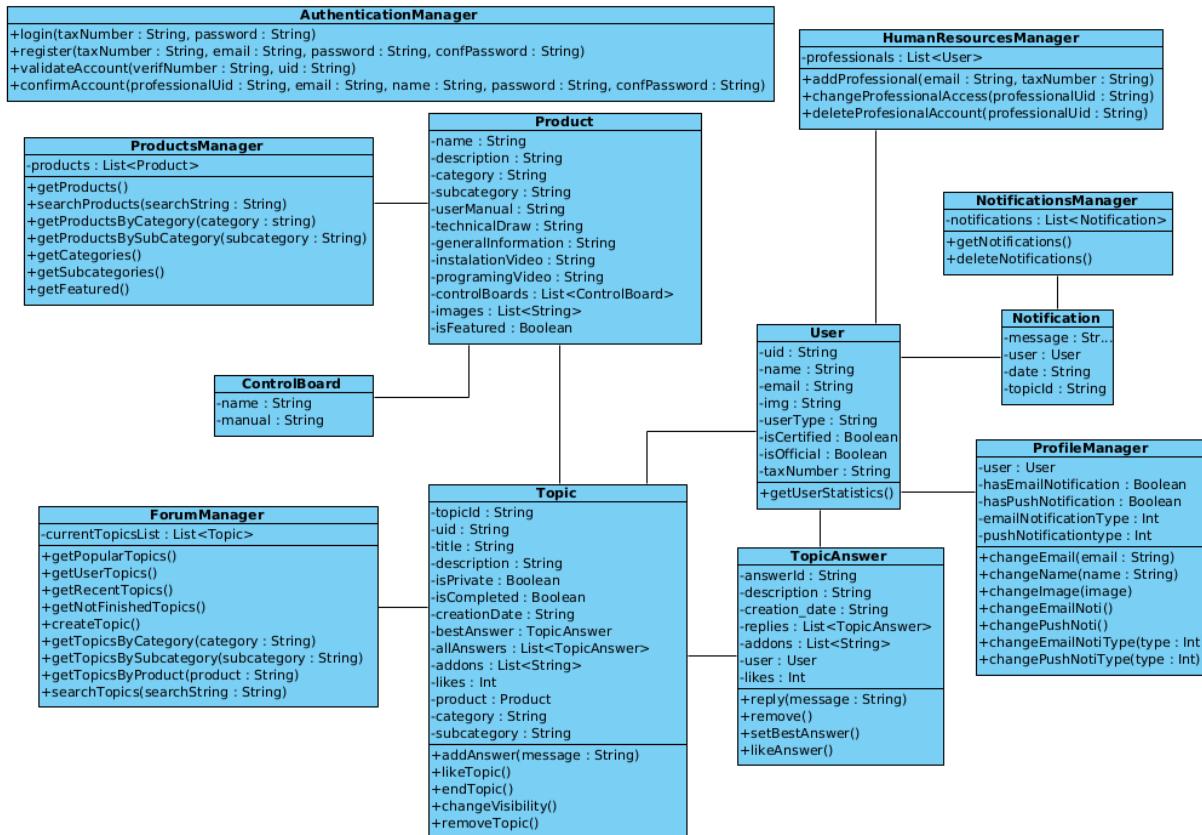


Figura 3.21: Diagrama de classes Install&Go

3.14 Mockups

A propósito de desenhar um design para seguir e apresentar às partes interessadas antes de iniciar a fase de desenvolvimento, então foram realizadas mockups do design da aplicação. Este design foi iterativamente revisto pelas partes interessadas e afinado até chegar ao seu estado final.

3.14.1 Página Inicial

A página inicial da aplicação, dá ao utilizador a possibilidade de navegar pelo produtos do catálogo, conseguindo também filtrar pelas categorias e subcategorias do catálogo, assim como também realizar uma pesquisa rápida pelos produtos, por fim este poderá navegar para o fórum. Caso um técnico esteja com sessão iniciada este poderá também visualizar o icon de notificações e a sua imagem de perfil.

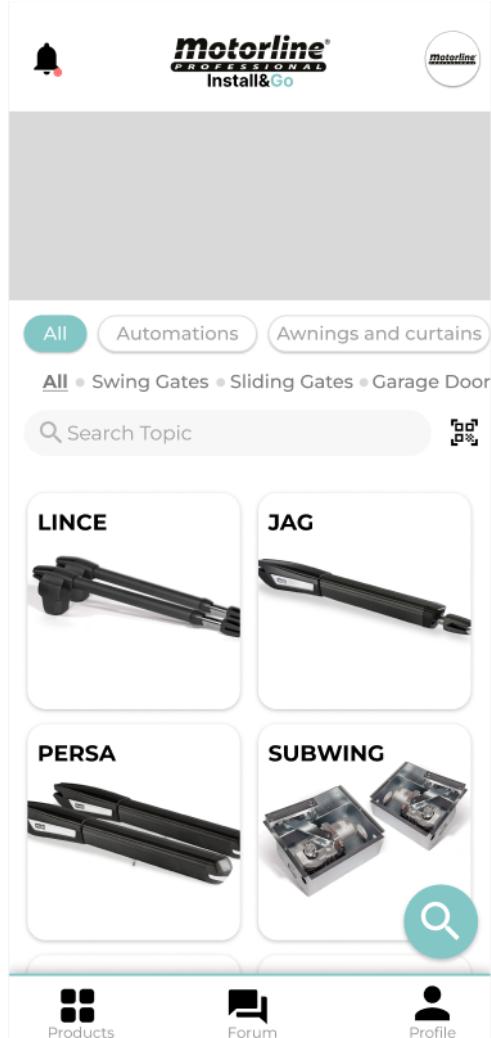


Figura 3.22: Página inicial do fórum

3.14.2 Autenticação - Login e Registo

Na autenticação primeiramente é possível realizar o login, onde o técnico poderá iniciar sessão no software e registo onde a empresa poderá realizar o registo no software.

The image displays two side-by-side screenshots of a software application interface titled "Install & Go".

(a) Página de login: This screenshot shows the login page. It features fields for "Tax number" (labeled "Tax Number") and "Password" (labeled "Password"). Below these fields are buttons for "Forgot Password?" and "Login". A "Continue without session" link is also present. At the bottom, there is a link for users who "Don't have an account yet? [Sign Up](#)".

(b) Página de registo: This screenshot shows the registration page. It features fields for "Tax number" (labeled "Tax Number"), "Company Name" (labeled "Company name"), and "Email" (labeled "Email"). Below these fields are buttons for "Password" and "Password Confirmation". A "Register" button is located at the bottom. At the top, there is a link for users who "Already have an account? [Login](#)".

Figura 3.23: Autenticação - Login e Registo

3.14.3 Autenticação - Ativação e Confirmação de conta

Na autenticação tem existe a página de confirmação de conta, onde um técnico que tem a sua conta recentemente adicionada poderá confirmar o registo da conta indicando as informações finais de conta, sendo por fim direcionado para a página de ativação de conta onde terá de colocar o código de ativação enviado para o seu email, esta página será também aberta caso um técnico realize login com uma conta que não foi ativada ou então sempre que um registo é finalizado.

(a) Página de confirmação de conta

Registered Details:
Email: joaopereira@motorline.com
Tax Number: 9999999999

Name
Name

Email
joaopereira@motorline.com

Password
Password

Password Confirmation
Password Confirmation

Register

Already have an account? [Login](#)

(b) Página de ativação de conta

Account Activation

Activation code
Activation code

Didn't receive code? [Resend code](#)

Cancel **Verify**

Figura 3.24: Autenticação - Ativação e Confirmação de conta

3.14.4 Página inicial fórum

O utilizador assim que se dirige ao fórum entrará na página inicial do mesmo, esta página permite navegar entre as diferentes listagens de tópicos acessíveis ao utilizador, aceder à página de pesquisas, filtrar a listagem atual por tipo de tópico e caso um técnico entre nesta página ele poderá também criar um novo tópico.

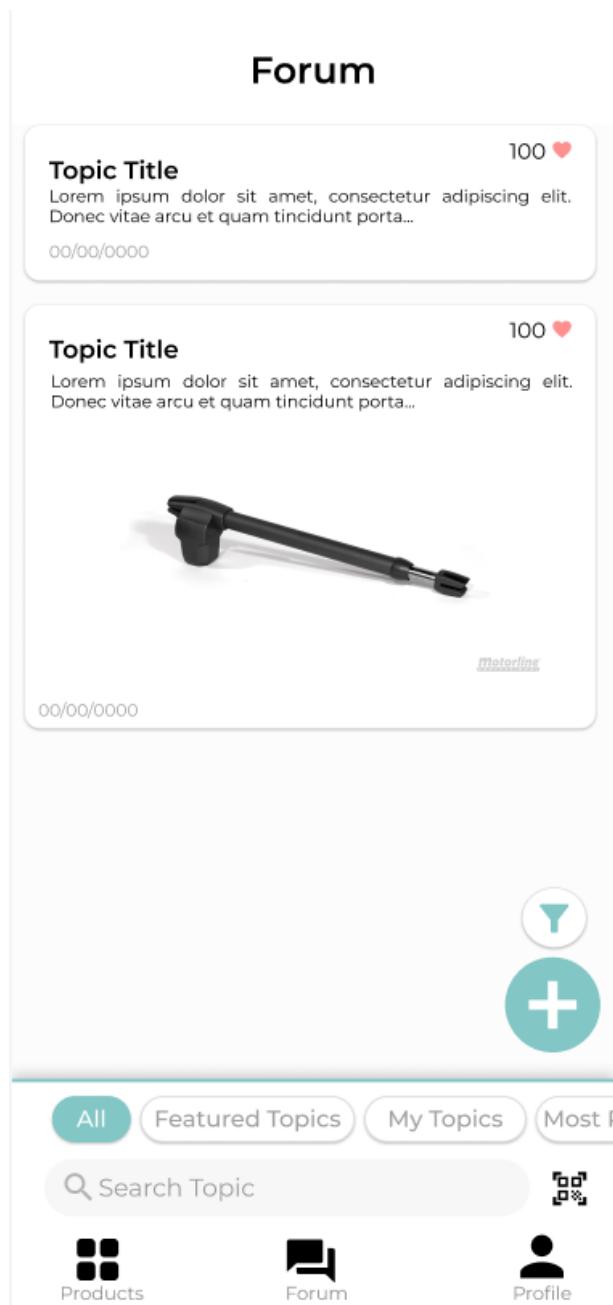


Figura 3.25: Página inicial do fórum

3.14.5 Página de detalhes de um tópico

Assim que o utilizador seleciona um tópico ele será encaminhado para a página de detalhes de um tópico, onde é indicado o nome do dono do tópico, a sua imagem de perfil, a hora de criação do mesmo, a quantidade de gostos, o título, descrição, imagens e comentários do tópico. Nesta página o utilizador poderá visualizar todas as respostas. Por fim o técnico consegue, além disso, gostar do tópico, gostar de respostas, responder ao tópico e a outras respostas.

Se o tópico for do técnico que está a visualizar o mesmo, este poderá também concluir, eliminar o tópico e alterar a sua visibilidade.

(a) Página de detalhes de um tópico

**(b) Página de detalhes de um tópico
do técnico**

Figura 3.26: Página de detalhes de tópico do software

3.14.6 Página de criação de um tópico

Quando um técnico inicia a criação de um tópico, este será direcionado para a página de criação de tópico, nesta página o técnico poderá inserir o título e descrição do tópico, assim como indicar a visibilidade do tópico, o tipo de tópico, o produto referente e por fim o técnico poderá anexar imagens. A qualquer momento o técnico poderá cancelar ou confirmar a criação do tópico.

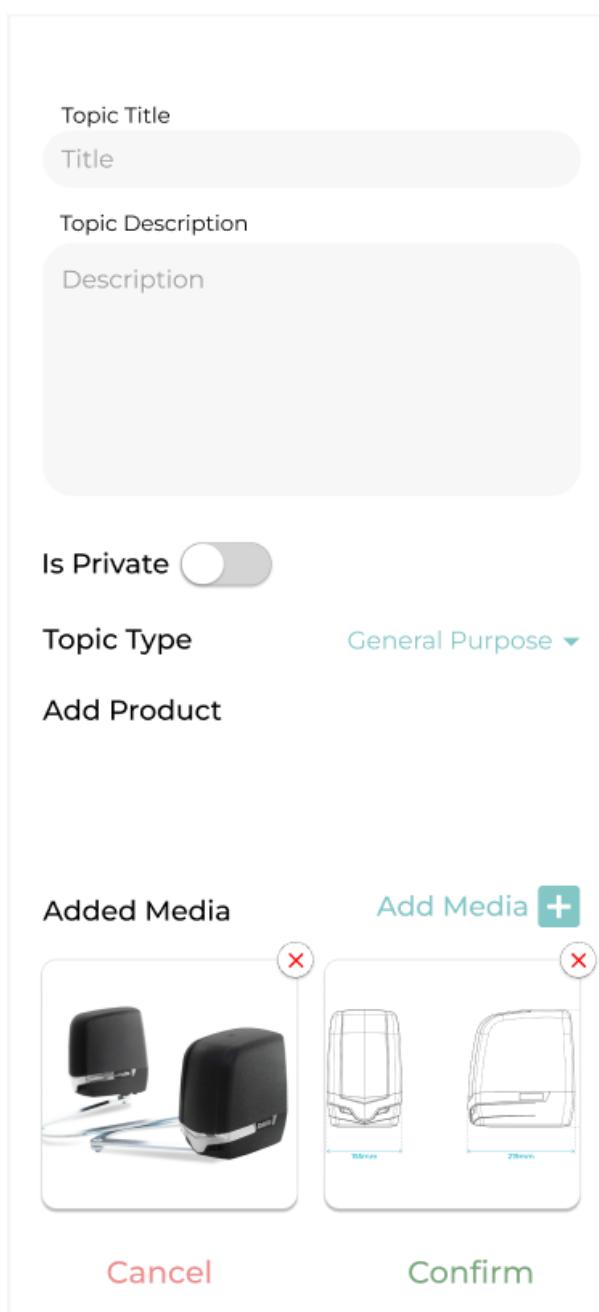


Figura 3.27: Página de criação de tópico

3.14.7 Página de notificações

Um técnico sempre que desejar poderá visualizar as suas notificações, para isso deverá se dirigir à página de notificações, neste é possível ver todas as notificações identificando quem enviou, qual a descrição da notificação e data de receção desta. O técnico poderá também apagar a notificação se assim desejar.



Figura 3.28: Página de notificações

3.14.8 Página de perfil de utilizador

O técnico sempre que desejar alterar as suas informações, deverá se deslocar ao seu perfil no qual é possível alterar o nome, email e imagem de perfil. Neste é possível também configurar as notificações indicando os métodos a receber notificação e o tipo de notificação a receber para cada método selecionado.

Caso uma empresa entre no perfil esta visualizará um botão para aceder à gestão de recursos humanos.

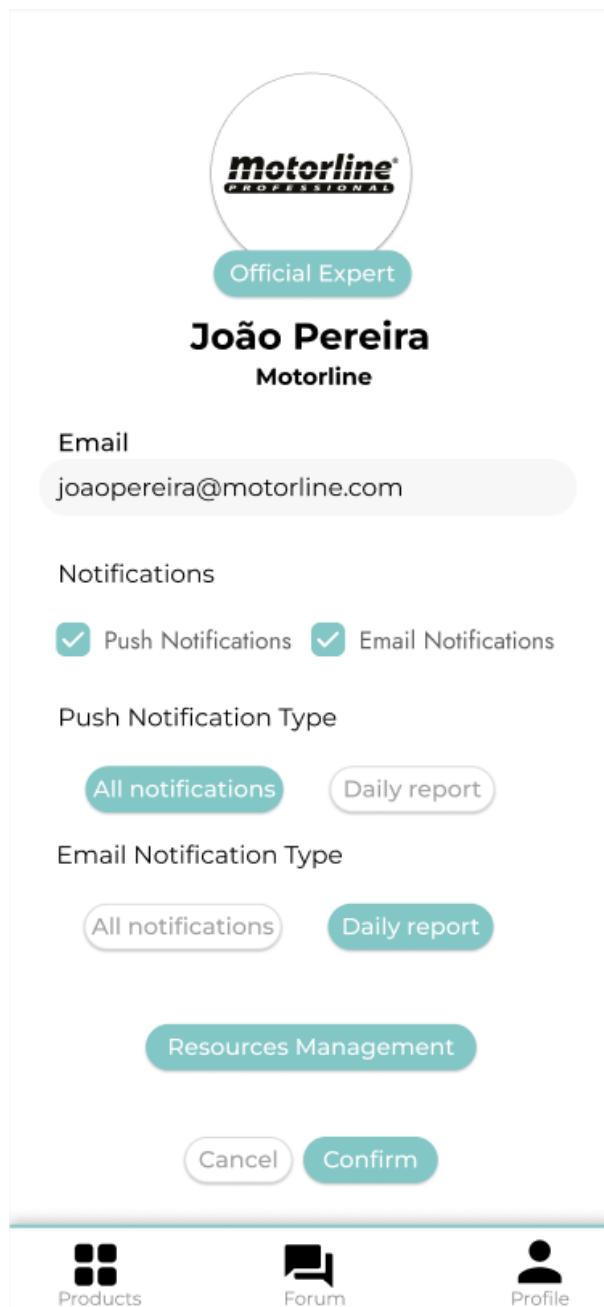


Figura 3.29: Página de perfil de utilizador

3.14.9 Página de gestão de recursos humanos

Sempre que é necessário registar novas contas de técnico a empresa deverá direcionar-se à página de gestão de recursos humanos, nesta página esta poderá registar novos técnicos ou gerir os técnicos já registados, conseguindo aceder aos seus perfis. Por fim a empresa consegue também pesquisar por técnico.

The mockup shows a mobile application interface for 'Resources Management'. At the top, there is a back arrow icon and the title 'Resources Management'. Below this is a table with two columns: 'Tax Number' and 'Name'. There are eight rows in the table, each containing a tax number (e.g., 9999999999) and a name (e.g., Nome Técnico). A large green circular button with a white plus sign is located at the bottom right of the table area. Below the table is a search bar with the placeholder 'Search Person' and a magnifying glass icon. At the very bottom, there are three navigation icons: 'Products' (represented by a grid of squares), 'Forum' (represented by a speech bubble), and 'Profile' (represented by a person icon).

Tax Number	Name
9999999999	Nome Técnico

Figura 3.30: Página de gestão de recursos humanos

3.14.10 Página de perfil de técnico registado

Para ver as estatísticas de um técnico ou para realizar alguma operação sobre este, a empresa deverá clicar no técnico desejado na página de recursos humanos, onde será direcionada para a página de perfil do técnico e visualizará as estatísticas deste, assim como também as suas informações e tem a possibilidade de impedir acesso à conta ou então remover a conta da plataforma.

The screenshot shows a profile page for João Pereira, an Official Expert at Motorline Professional. The page includes a circular logo, contact information, access control options, and performance metrics.

Profile Information:

- Logo: Motorline Professional
- Role: Official Expert
- Name: João Pereira
- Company: Motorline
- Email: joaopereira@motorline.com

Access Control:

- Impedir Acesso (Ban User) button with a crossed-out user icon
- Remover conta (Delete Account) button with a trash bin icon

Performance Metrics:

Category	Value
Questions	20
Answers	265
Topics Completed	41

Page Navigation:

- Products (grid icon)
- Forum (chat icon)
- Profile (person icon)

Figura 3.31: Página de perfil de técnico registado

3.14.11 Página de registo de novo técnico

Sempre que uma empresa deseja registar um novo técnico esta deverá clicar em registrar novo técnico onde será direcionada para a página de registo de técnicos, nesta página é pedido o número de contribuinte e email do técnico.

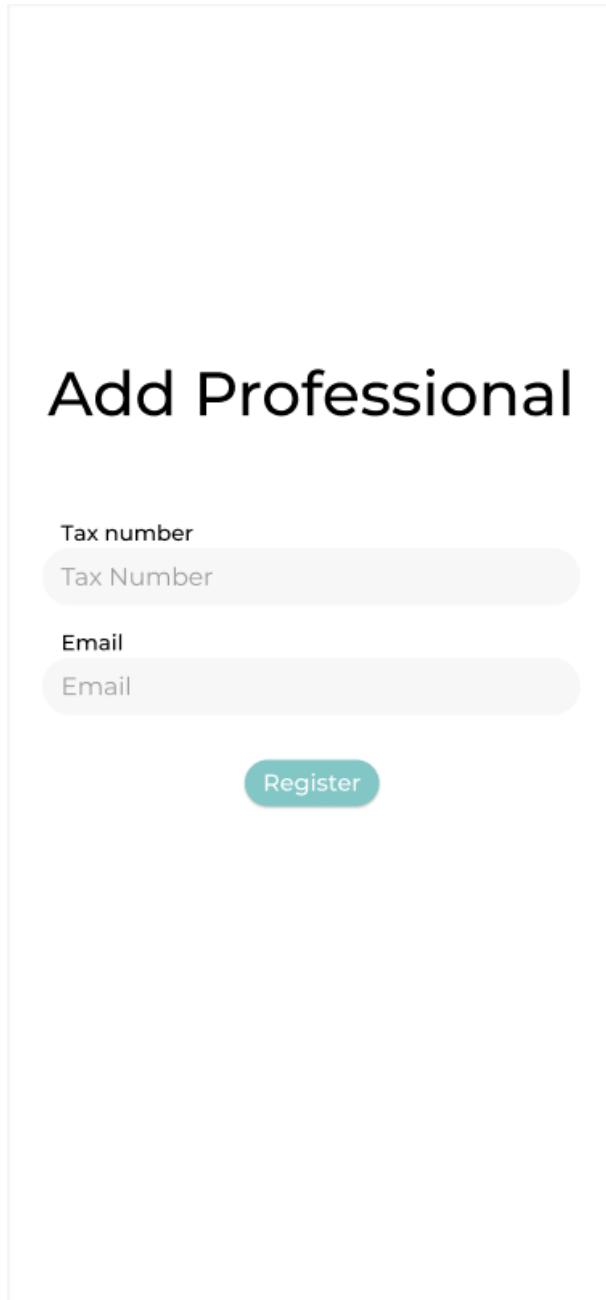


Figura 3.32: Página de registo de novo técnico

3.15 Diagramas de atividades

De forma a ser possível detalhar de forma simples as ações do ator nos diferentes ecrãs foram desenvolvidos diagramas de atividades, com isto

3.15.1 Diagrama de atividades página inicial

Na página inicial da aplicação é possível se deslocar para o fórum, ver as notificações, realizar operações de catálogo e deslocar-se para o perfil.

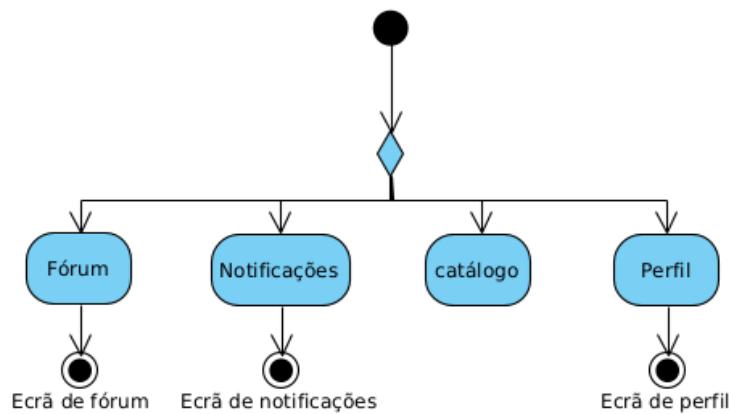


Figura 3.33: Diagrama de atividades de página inicial da aplicação

3.15.2 Diagrama de atividades página de perfil

Um técnico poderá necessitar de alterar alguma configuração ou informação sua, pelo que deverá se dirigir ao ecrã de perfil. Neste ecrã, este poderá alterar a sua imagem de perfil, o seu nome e seu email. Além disso poderá também selecionar os métodos de notificação que deseja receber e os tipos de notificação para estes métodos. Caso uma empresa veja o seu perfil este poderá além das operações acima mencionadas gerir os seus recursos humanos sendo encaminhada para o ecrã de gestão de recursos humanos.

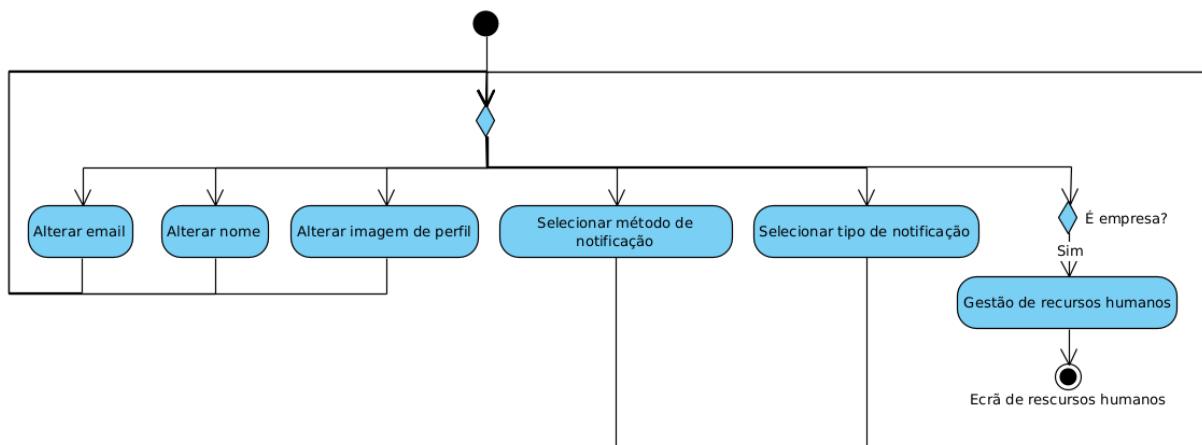


Figura 3.34: Diagrama de atividades de página de perfil

3.15.3 Diagrama de atividades página inicial do fórum

Na página inicial do fórum o utilizador poderá então selecionar um dos tipos de pesquisa, escrita ou código QR, filtrar por tipo de tópico ou selecionar tópico. Este conseguirá também ver as listagens de tópicos em destaque, tópicos mais recentes e por fim tópicos por responder, estas listas poderão ser também filtradas por tipo, conseguindo o técnico depois realizar todas as ações novamente, sobre estas listas o utilizador também poderá selecionar um tópico o que o redirecionará para o ecrã de detalhes de tópico. Para além das destas operações, o técnico consegue também ver os seus tópicos e criar um novo tópico.

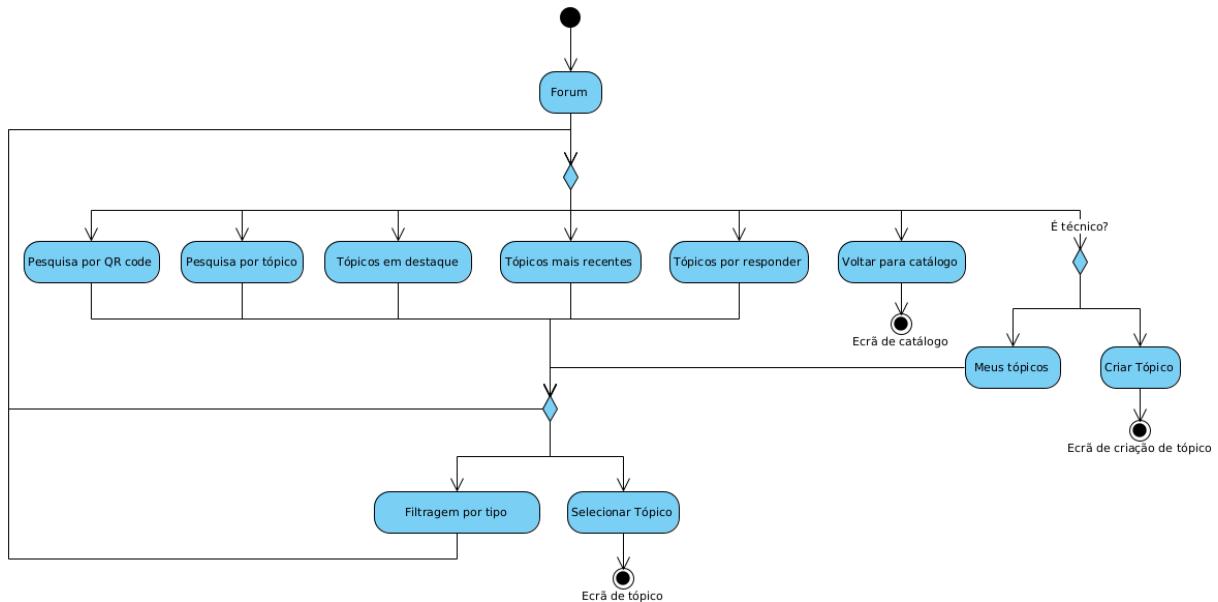


Figura 3.35: Diagrama de atividades de página inicial do fórum

3.15.4 Diagrama de atividades página de criação de tópico

Quando o técnico decide criar um tópico, este é encaminhado para o ecrã de criação de tópico onde este obrigatoriamente tem de indicar o título, descrição e tipo do tópico, por predefinição a visibilidade deste é pública, mas o técnico poderá alterar esta visibilidade também. Facultativamente o técnico poderá indicar o produto referente ao tópico, assim como anexar imagens, conseguindo também remover estas. A qualquer momento o técnico poderá confirmar a criação do tópico, quando esta ação inicia, é verificado se o título e descrição estão preenchidos, caso estes dados não estejam preenchidos é indicado que estes dados estão em falta.

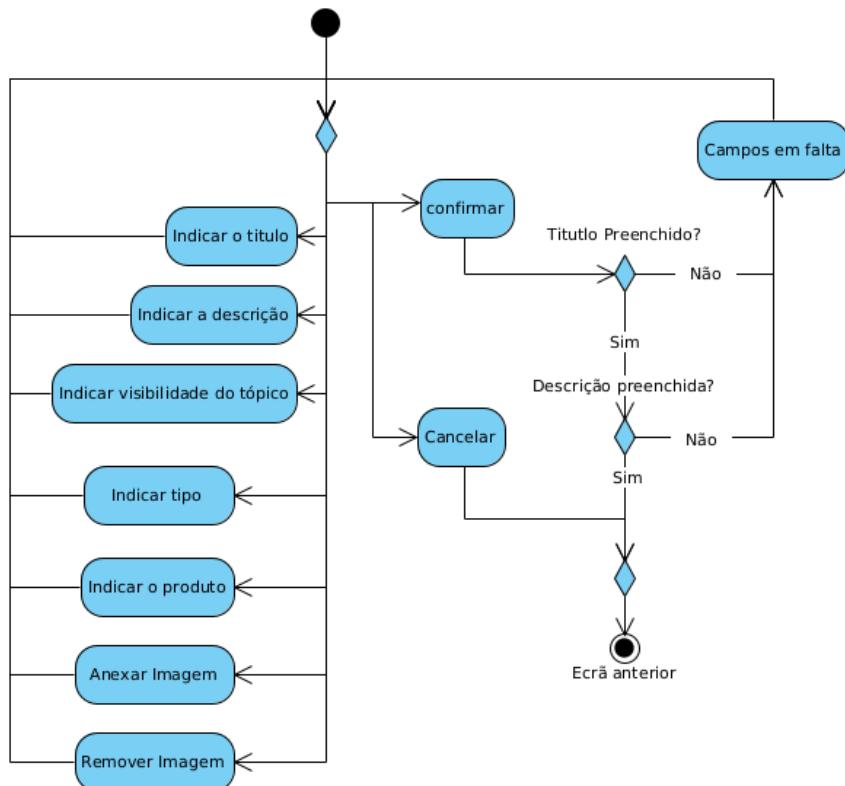


Figura 3.36: Diagrama de atividades de página de criação de tópico

3.15.5 Diagrama de atividades página de detalhes do tópico

Assim que o utilizador seleciona um tópico este é redirecionado para o ecrã de detalhes de tópico, no qual este poderá visualizar todas as respostas e ver as imagens anexadas em ponto grande. Já o técnico poderá, além disso, apagar um comentário caso seja seu, gostar do tópico e de uma resposta, comentar, responder a um comentário. Caso o tópico seja do técnico, este poderá também alterar a visibilidade do tópico, marcar como concluído ou remover este voltando para o ecrã anterior. A qualquer momento o técnico poderá também retroceder para o ecrã anterior.

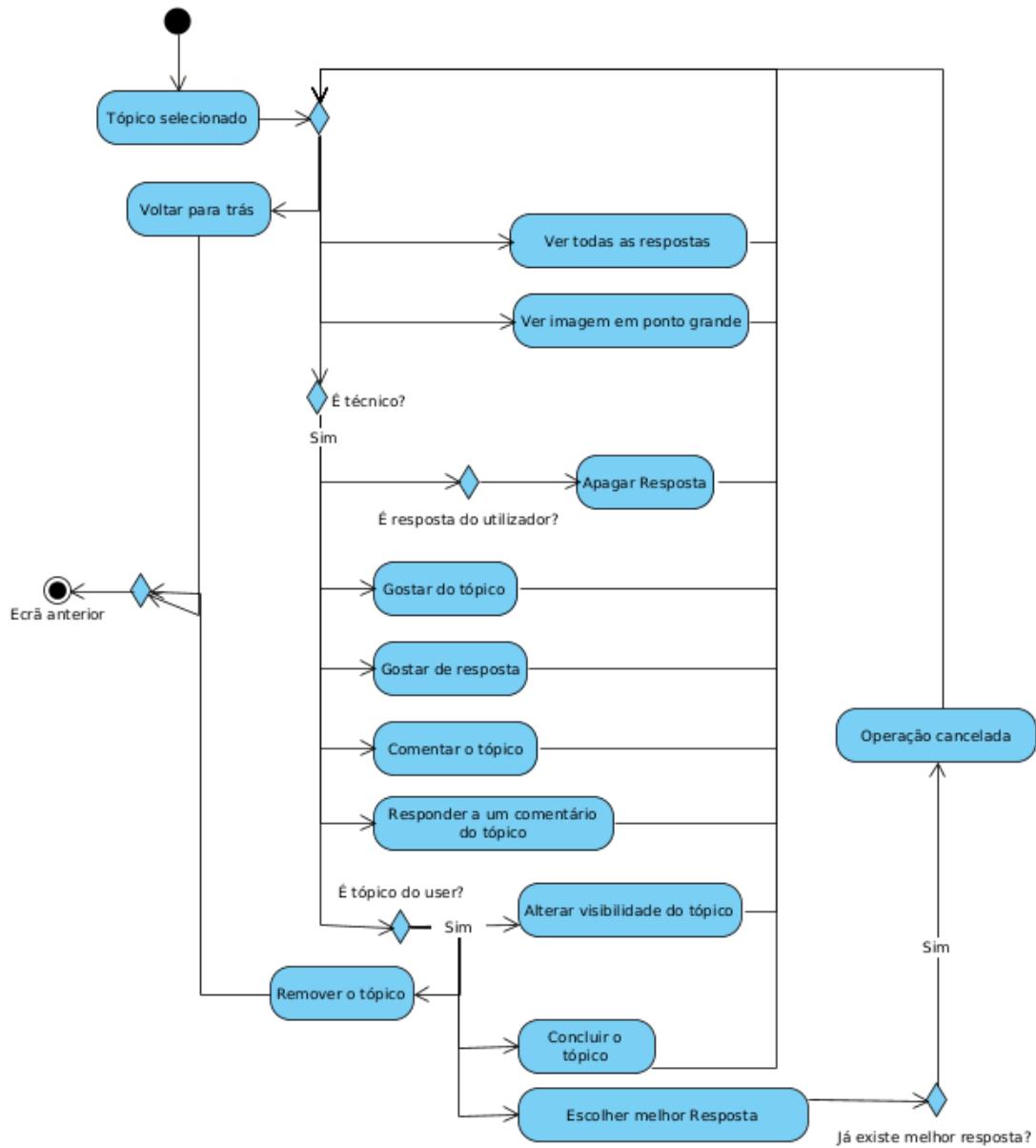


Figura 3.37: Diagrama de atividades de página de detalhes do tópico

3.15.6 Diagrama de atividades páginas de autenticação

Para realizar a ativação da conta de técnico, assim que este realiza o registo, confirmação de conta ou o login com uma conta que não se encontra ativada, este é encaminhado para o ecrã de ativação de conta, neste ecrã este poderá cancelar a ativação de conta, ou então indicar o código de ativação de conta, caso este código esteja errado, o técnico deverá inserir novamente o código, caso seja um código correto o técnico validará a sua conta e ficará autenticado. O técnico poderá também em caso de necessidade pedir o envio de um novo código de ativação de conta.

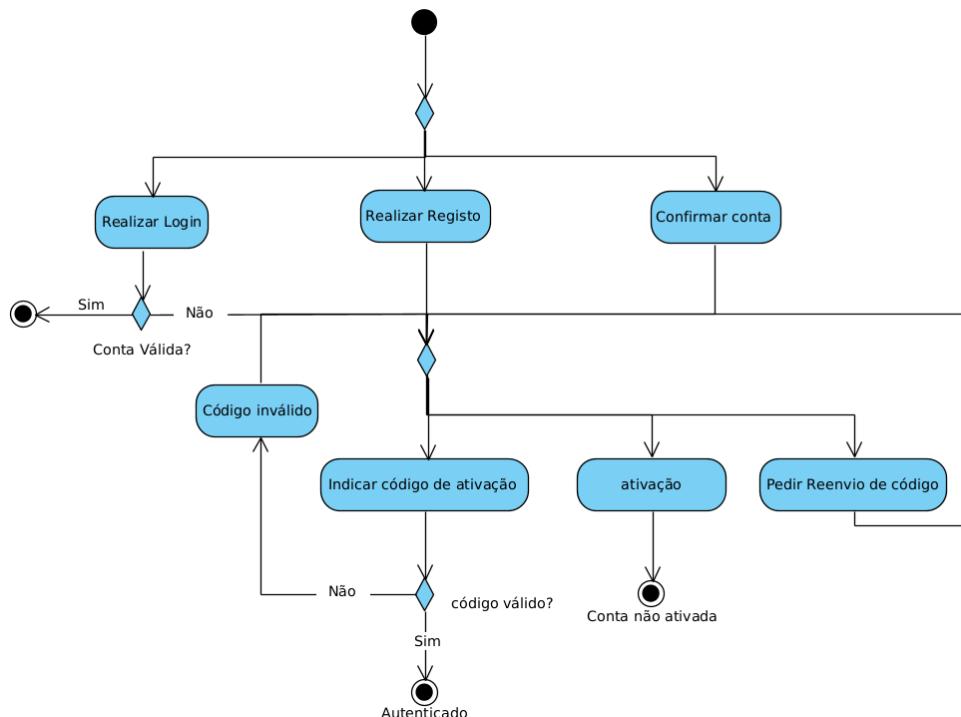


Figura 3.38: Diagrama de atividades de página de validação de conta

3.15.7 Diagrama de atividades registrar técnico

Assim que uma empresa inicia o registo de um técnico esta é redirecionada para a página de registo de técnico. Nesta página esta terá de indicar o número de contribuinte e email do técnico, por fim poderá confirmar o registo de conta sendo redirecionada para a página anterior.

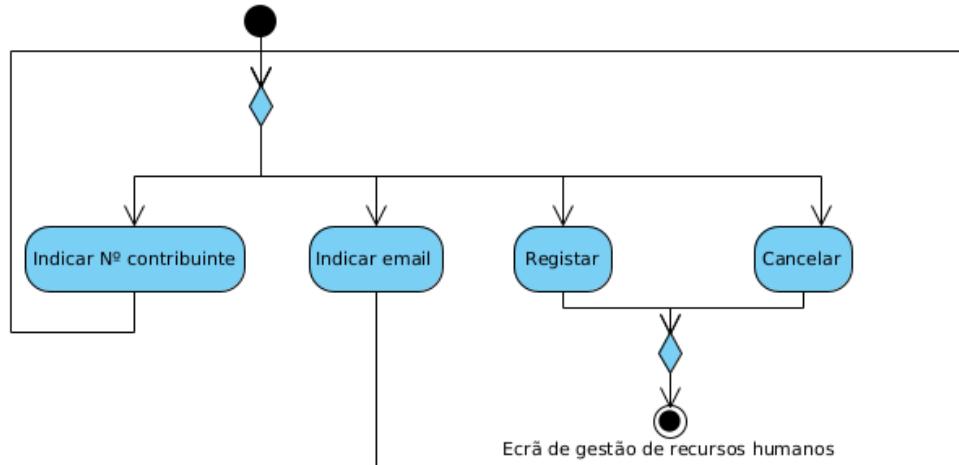


Figura 3.39: Diagrama de atividades de página de registrar técnico

3.15.8 Diagrama de atividades confirmar conta

Quando uma conta de técnico é registada um email de confirmação é enviado para o técnico, assim que este recebe o email deverá clicar em confirmar a conta sendo redirecionado para a página de confirmação de conta. Nesta página o técnico poderá alterar o seu email, indicar o seu nome, indicar a password e confirmar esta, finalizando quando decidir se registrar.

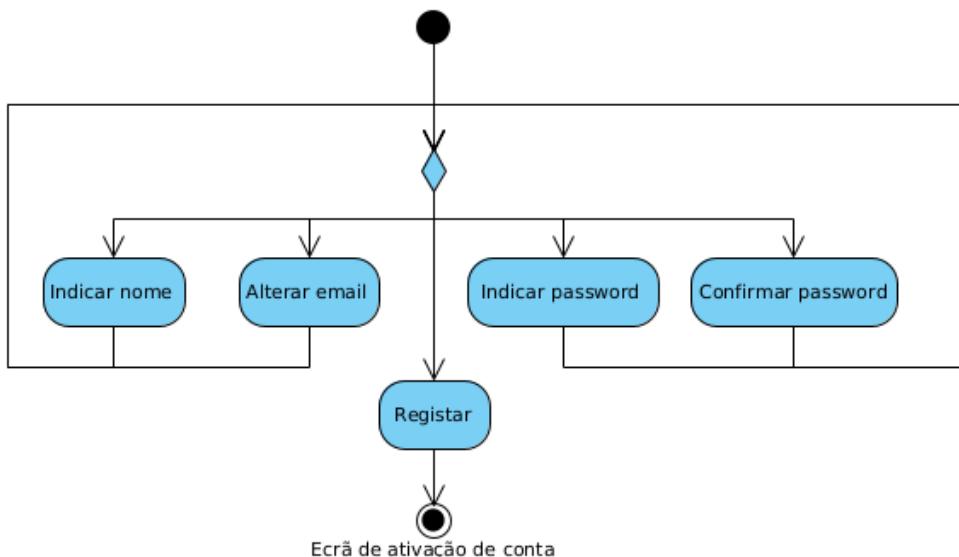


Figura 3.40: Diagrama de atividades de página de confirmar conta de técnico

3.16 Diagramas de estados

Para especificar os principais processos do projeto foram então desenvolvidos diagramas de estados, sendo então pretendido demonstrar o processo de criação de um tópico do fórum por parte de um técnico, o processo de aceder e responder a um tópico e o login com ativação de conta visto que estas interações são as de maior significância e regradas no software.

3.16.1 Diagrama de estados criação de tópico

Com o diagrama de estados de criação de tópico é pretendido demonstrar o processo de criação de um tópico por parte de um técnico. Assim sendo este primeiramente terá de estar autenticado, caso não esteja este será encaminhado para autenticação. De seguida criará tópico, após preencher os campos desejados este poderá confirmar o tópico, caso confirme é verificado se o tópico possui título, caso não possua, é invalido pelo que o técnico deverá preencher os dados em falta, caso o título esteja preenchido é verificado se possui descrição e tipo, caso não possua é seguido o mesmo fluxo que o caso anterior, caso contrário é criado um tópico. Se o técnico não desejar confirmar o tópico ele poderá cancelar, quando assim o faz este torna-se cancelado.

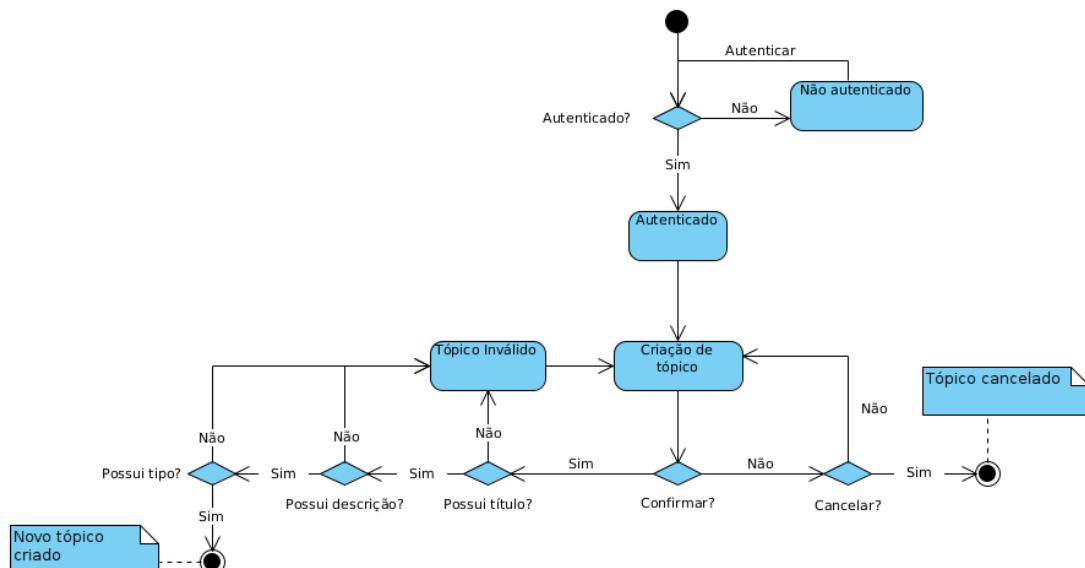


Figura 3.41: Diagrama de estados de criar tópico

3.16.2 Diagrama de estados responder a tópico

Com o diagrama de estados de responder a tópico é pretendido demonstrar o processo de seleção e responder a um tópico por parte de um técnico. Assim sendo que o técnico primeiramente deverá estar autenticado, caso não esteja este será encaminhado para a autenticação. Após a autenticação o técnico estará autenticado e irá por predefinição ver tópicos em destaque, nesta listagem este selecionará um tópico ficando assim o tópico selecionado. Assim que o tópico se encontra selecionado o técnico conseguirá responder a este criando um comentário. Após a criação do comentário este poderá confirmar o comentário, caso confirme o comentário ficará criado, caso contrário este comentário ficará cancelado.

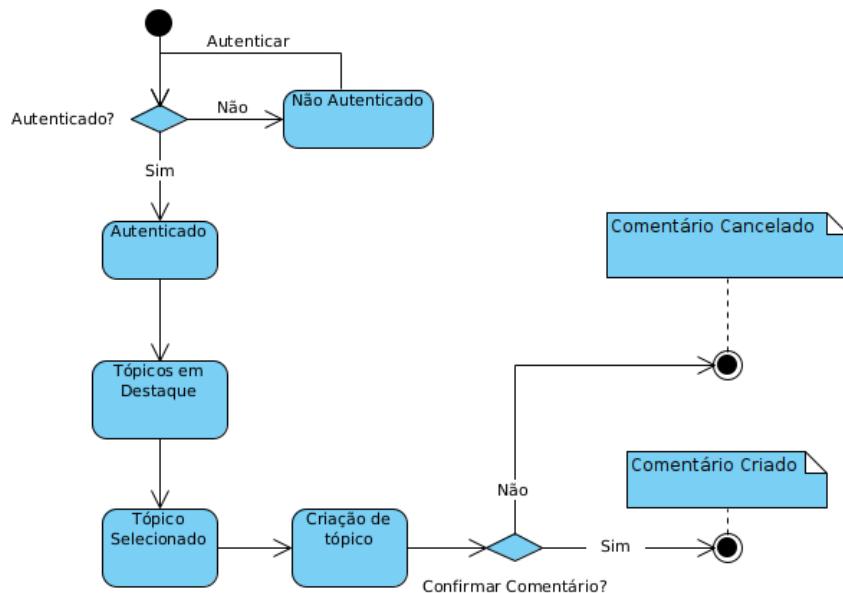


Figura 3.42: Diagrama de estados de criar tópico

3.16.3 Diagrama de estados autenticação e validação de conta

Assim que o técnico decide realizar o login na aplicação este indica as suas credenciais, caso estas credenciais não estejam corretas, a autenticação será incorreta e deverá alterar as credenciais. Caso as credenciais estejam corretas e a conta válida o técnico ficará autenticado, caso contrário o este terá uma conta inválida, pelo que deverá ser validada, para isso o este deverá inserir o código de validação, se o código estiver correto, a conta será validada e o técnico ficará autenticado, caso contrário o código será invalido e o deverá indicar o seu código de validação novamente.

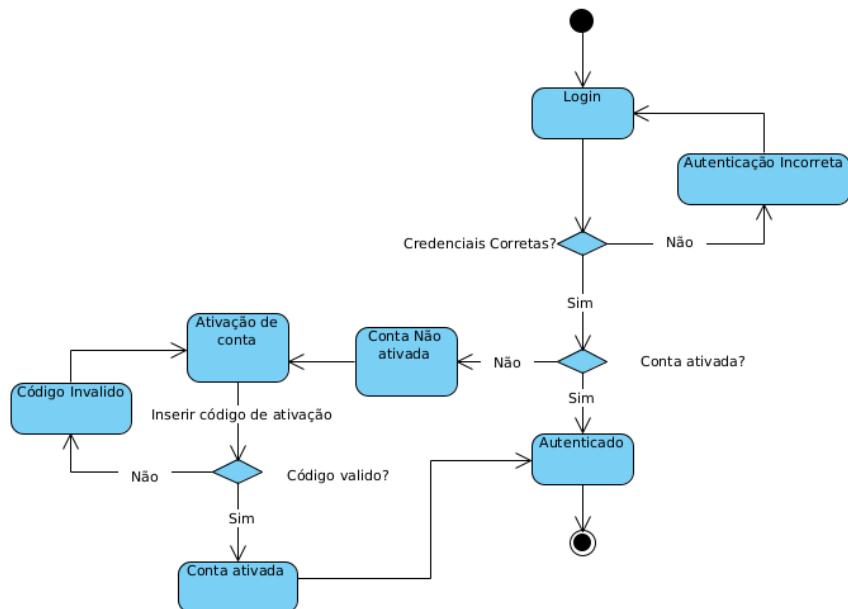


Figura 3.43: Diagrama de estados de autenticação e validação de conta

3.17 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.

3.17.1 Diagrama de sequência Login e ativação de conta

Através deste diagrama (Figura 3.44) é entendido 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.

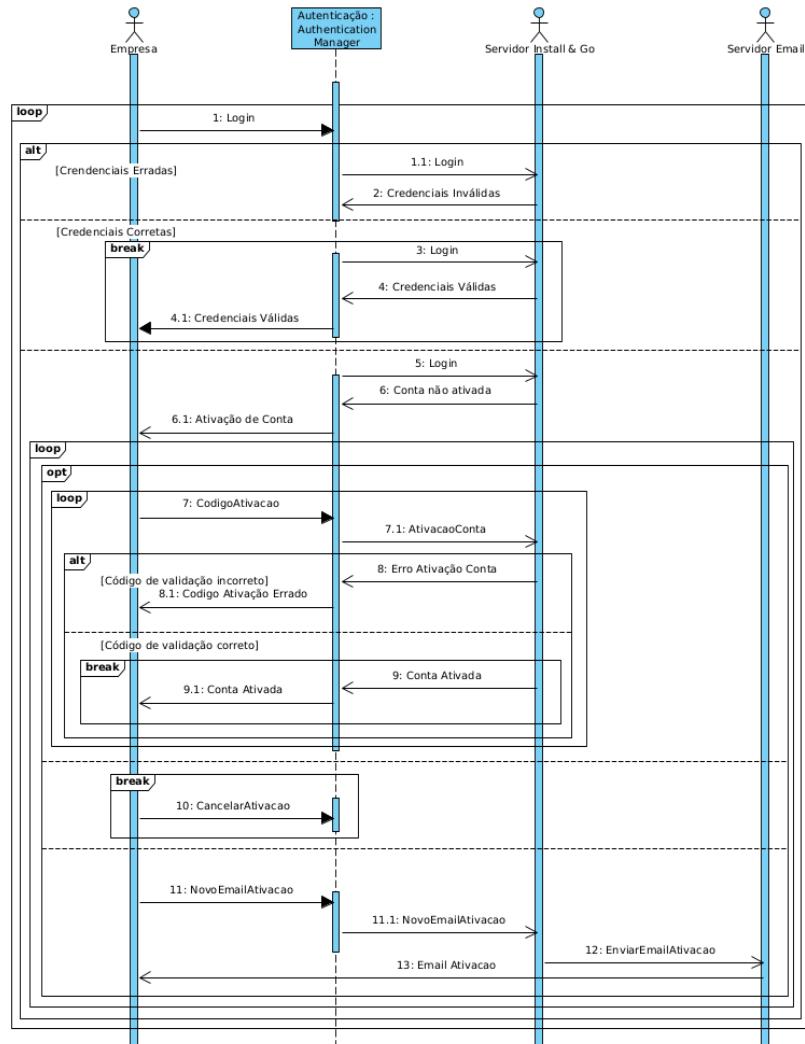


Figura 3.44: Diagrama de sequência de login e ativação de conta

3.17.2 Diagrama de sequência Registo e ativação de conta

Através do diagrama abaixo representado (Figura 3.45) é possível perceber que quando uma empresa realiza o registo este será enviado para o servidor, o qual registará 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.

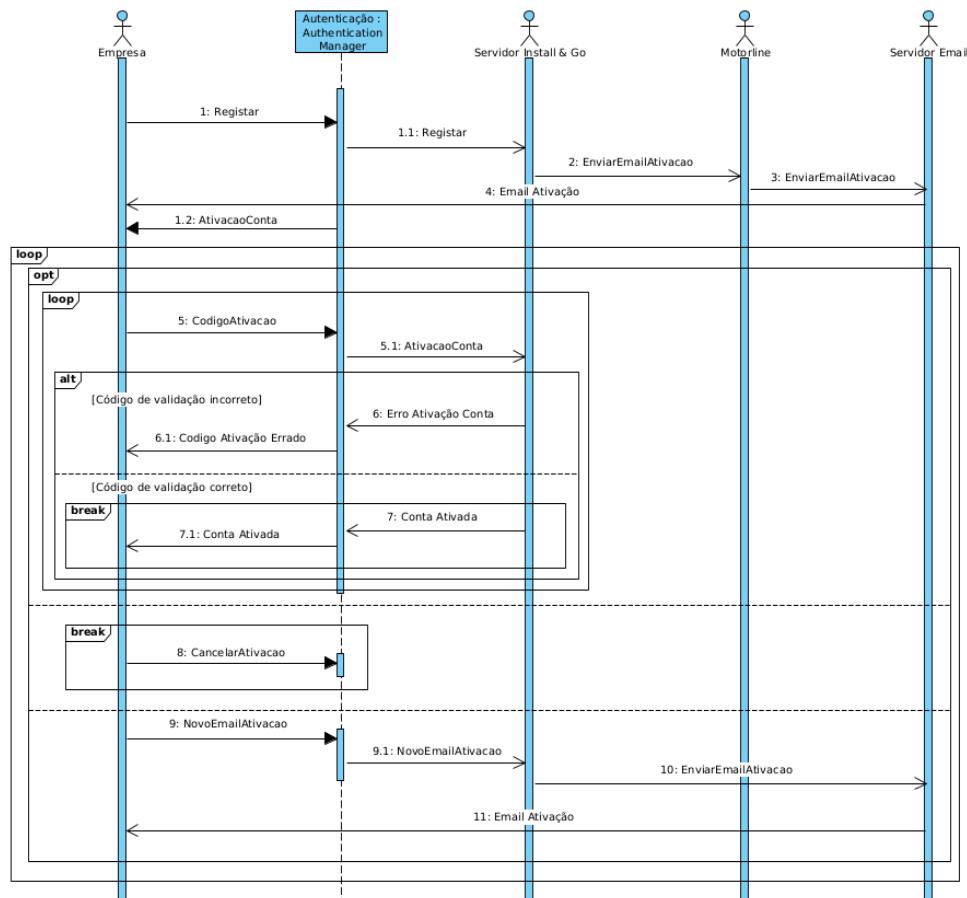


Figura 3.45: Diagrama de sequência de registo e validação de conta

3.17.3 Diagrama de sequência registo de técnicos

Através do diagrama abaixo representado (Figura 3.46) é possível perceber que quando uma empresa deseja registar 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.

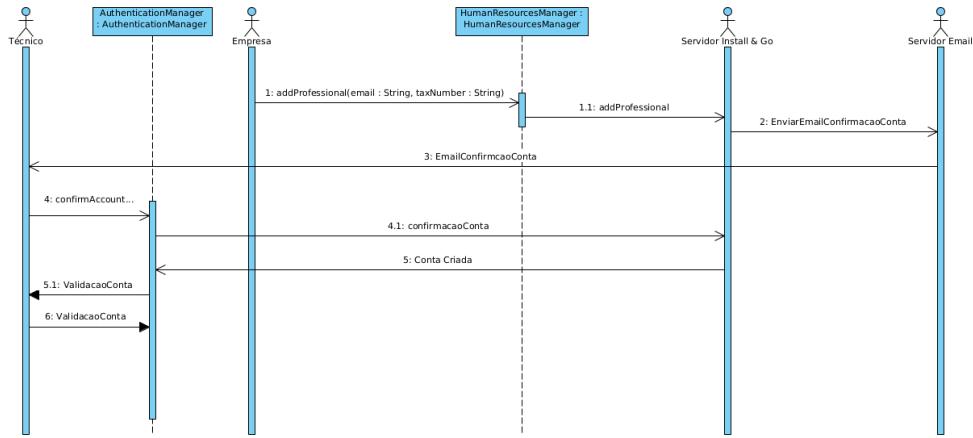


Figura 3.46: Diagrama de sequência de registo de técnicos

3.18 Arquitetura de sistema

Na Figura 3.47 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.

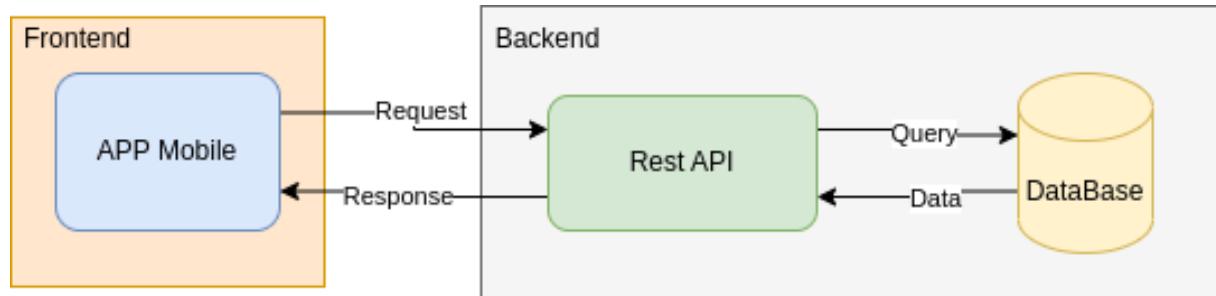


Figura 3.47: Arquitetura do sistema

3.18.1 Arquitetura de funcional

Para especificar a implementação da api rest foi então criada uma arquitetura de backend(Figura 3.48), 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.

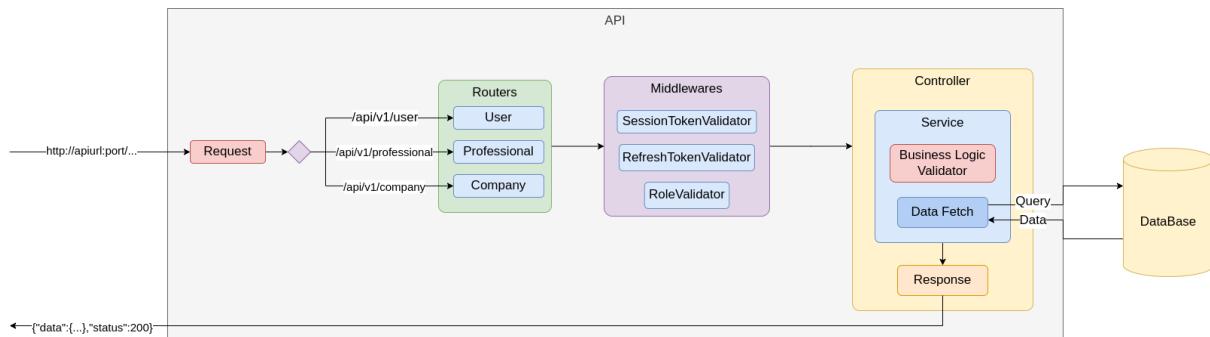


Figura 3.48: Arquitetura do funcional

3.18.2 Arquitetura de componentes

Após a percepção 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.

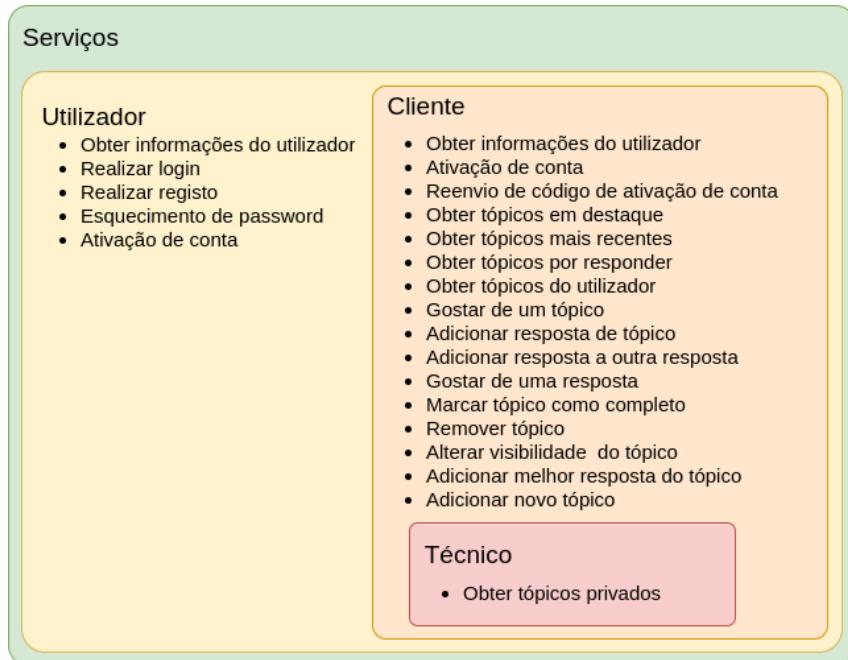


Figura 3.49: Arquitetura de componentes

3.18.3 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 3.19: Tabela de endpoints

Serviço	Autor	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
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

Continued on next page

Tabela 3.19: Tabela de endpoints (Continued)

Adicionar melhor resposta do tópico	Cliente	baseurl/client/topics/:topicId/answers/:answerId/best-answer	PUT
Adicionar novo tópico	Cliente	baseurl/client/topics/	POST

4. Trabalho desenvolvido

4.1 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 obter uma página web, ler a página e obter dados desta, geralmente utilizando bots. O grande problema com web scraping é que pode ser facilmente detetado. Tendo em conta este problema surgiram duas grandes formas principais de realizar web scraping, a mais comum sendo realizar um pedido para obter uma página web e ler então esta, sendo assim um processo rápido e simples. A segunda forma de realizar web scraping é através da simulação da ação humana conseguindo abrir o navegador pesquisar pela página desejada, descarregar a página e daí ler esta, tornando-se então em um processo lento e complexo. A grande diferença entre estas duas formas é a velocidade, visto que a segunda forma tem de esperar que o navegador inicie, de seguida terá de esperar que a página carregue e apenas após este processo poderá ser lida a página web.

Na reunião mencionada anteriormente foi decidido que o web scraper iria apenas correr 1 vez por mês de forma a evitar a sobrelocação do servidor, não existindo problema visto que o catálogo não é atualizado regularmente. Para agilizar a realização do web scraper foi disponibilizado pela empresa a estrutura do website a seguir para obter as informações da página web.

4.1.1 Implementação web scraper

De forma a implementar e testar o web scraper sem sobreloitar o servidor, foi então descarregado todo o website localmente, conseguindo assim simular o mesmo.

Para implementar o web scraper foi optado pela abordagem mais simples, realizar um pedido para obter a pagina web, ler a página para obter os dados e guardar os dados.

Para isto foi optado pela linguagem python devido à facilidade desta lidar com grandes quantidades de dados. De forma a facilitar a localização dos dados na página foi utilizada a biblioteca bs4, também conhecida como beautiful soup, esta biblioteca permite alimentar com uma página web e de seguida realizar pesquisas sobre esta página baseado em tags e atributos dos elementos.

Tendo esta base em conta foi então primeiramente estudado que dados seriam necessários, sendo estes então:

1. Categorias e subcategorias de produtos;
2. Produtos de cada categoria e subcategoria;
3. Documentação dos produtos;
4. Imagens e videos dos produtos;

Para guardar estes dados foi utilizado um dicionário que contém primeiramente como chaves as categorias de produtos, para cada categoria contém mais um dicionário com as subcategorias de produtos e para cada categoria existe uma lista de produtos, contendo o nome de produto, imagem de amostra e url do mesmo. Por fim a chave produtos contém a lista de todos os produtos, sendo cada produto representado também por um dicionário, que contém como chaves os atributos do mesmo. A utilização dicionários e listas para guardar estes dados deve-se a que o objetivo será guardar estes dados em json e a transformação é simplificada utilizando estas estruturas devido à sua proximidade com a estrutura json.

```
{
    "produtos": [
        {
            "nome": "nome do produto",
            "categoria": "categoria do produto",
            "subcategoria": "subcategoria do produto",
            "imagem-amostra": "imagem de amostra do produto",
            "imagens": [
                "links de imagens de produtos"
            ],
            "descricao": "descrição do produto",
            "documentacao": [
                {
                    "nome": "Nome da documentação",
                    "url": "link da documentação"
                }
            ],
            "placas-controlo": [
                {
                    "nome": "nome da placa de controlo",
                    "url": "documentação placa de controlo"
                }
            ],
            "informacao-geral": "imagem de informação geral",
            "desenho-tecnico": "imagem de desenho técnico",
            "videos": [
                {
                    "name": "nome do video",
                    "url": "link do video"
                }
            ]
        },
        "categorias": {
            "nome_categoria": {
                "nome_subcategoria": [
                    {
                        "nome": "nome produto",
                        "link": "link página do produto",
                        "imagem-amostra": "link imagem de amostra do produto"
                    }
                ]
            }
        }
    ]
}
```

Figura 4.1: Estrutura dos dados obtidos

Após uma análise da estrutura do website foi percebido que a página geral de produtos possui todas as categorias de produtos, assim como também as subcategorias de produtos com urls para as páginas que contém todos os produtos das subcategorias. Sendo assim foi primeiramente percebido que cada conjunto é uma secção, pelo que é obtido todas as secções de categorias e para cada uma destas secções é obtido o título da secção que equivale ao nome da categoria e também todos os correspondentes a clicáveis. Os clicáveis corresponde às subcategorias de cada categoria estes clicáveis contém também um url que redireciona para a página de produtos da subcategoria.



Figura 4.2: Página geral de produtos

Sendo assim já é possível identificar cada categoria e subcategoria, assim como também o url da página de produtos para cada subcategoria. Mas após alguma análise dos dados foi percebido que estas não contêm acentuação devido à sua formatação no website. Para resolver este problema foi pesquisado por ferramentas capazes de corrigir estes erros ortográficos. Pelo que foi descoberto que a biblioteca mais utilizada em python para resolver este problema é a biblioteca spellchecker, esta ferramenta é a mais utilizada devido à sua capacidade de corrigir erros ortográficos em diversas linguagens. Sendo assim sempre que uma categoria e subcategoria é obtida, antes de ser guardada, esta é corrigida.

Após isto cada url é aberto e são obtidos os urls de produtos e imagens de amostra dos produtos, para isto foi obtido todos os elementos clicáveis existentes na secção de produtos de cada página, sendo que cada um corresponde a um produto, para obter o nome do produto correspondente foi utilizado o nome contido no url da página de produto, sendo que todos os produtos seguem a mesma estrutura, sendo esta, /produtos/nome-produto. Sendo que em urls não é permitido utilizar acentuação e espaços, então todos os nomes foram corrigidos utilizando a mesma ferramenta mencionada anteriormente.

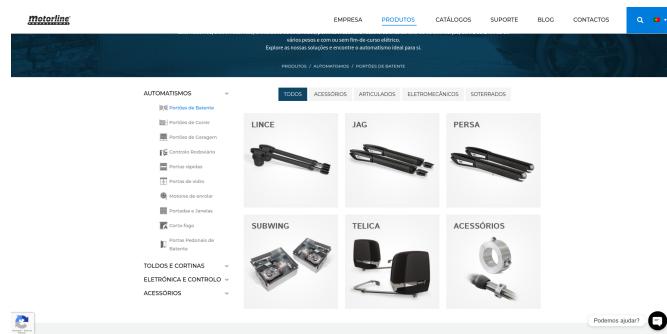


Figura 4.3: Página de produtos de uma subcategoria

Neste momento após correr o código foi percebido que existiam algumas páginas de produtos em que este não conseguia obter produtos, pelo que um erro era atirado, para perceber exatamente que páginas de produtos este erro acontecia, sempre que um erro era detetado este url seria adicionado a uma nova chave do dicionário mencionado anteriormente, esta chave tem o nome misses e contém todos os urls em que algum erro aconteceu. Foi então neste momento que foi percebido que nem todas as páginas de produtos são iguais e após uma reunião com o cliente este expôs que existem páginas de produtos e de detalhes de produtos que são muito diferentes das restantes.

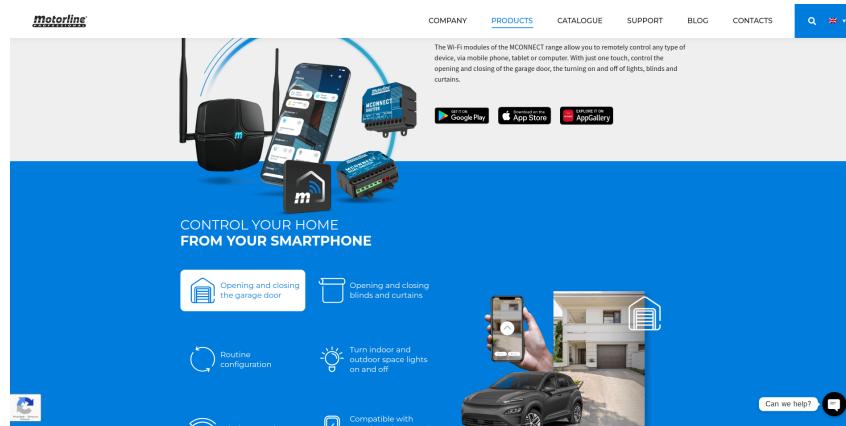


Figura 4.4: Página de produtos de uma subcategoria distinta

De forma ao restante do projeto não ser atrasado foi então decidido primeiramente obter todos os produtos que contêm páginas semelhantes, sendo assim para cada página de produto foi obtido o título que corresponde ao nome do produto, de seguida foi obtida a descrição do produto, o elemento que contém esta tem como id produto-descrição. As imagens dos produtos são disponibilizadas através de urls na secção da galeria do produto, sendo assim são obtidas todas as imagens desta galeria e de seguida todos os seus urls.

A documentação dos produtos pode ser disponibilizada através de urls para os manuais, ou com uma lista dropdown com todos os manuais disponíveis para download, sendo assim são obtidos todos os urls da secção de documentação, assim como os seus nomes e todas as opções de documentação do dropdown se este existir.

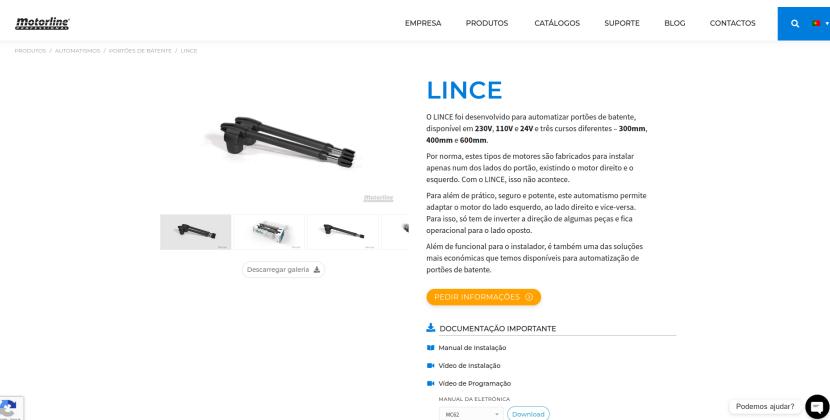


Figura 4.5: Página de detalhes de produto, secção inicial

As imagens de desenho técnico e informação geral estão disponibilizadas na secção correspondente ao nome de cada uma, sendo assim obtidas estas secções e caso estas existam são obtidas as imagens e os seus urls.

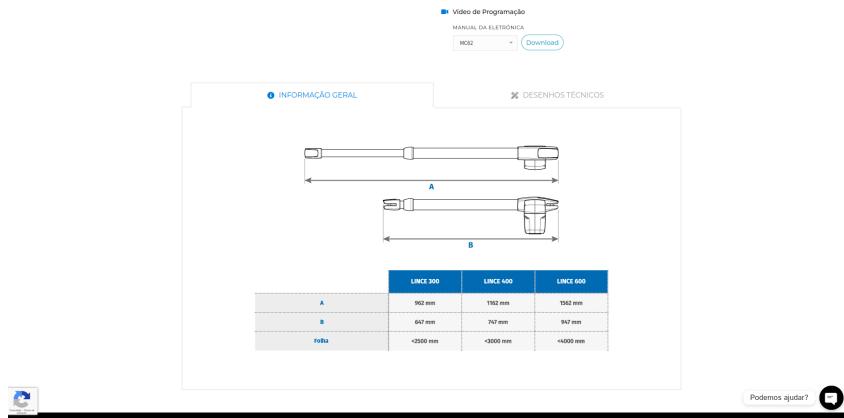


Figura 4.6: Página de detalhes de produto, secção de informações

Os videos de produtos estão disponíveis na secção de videos, sendo que cada secção de videos contém o nome do video e por sua vez o video. Estes videos são demonstrados utilizando um elemento iframe, este elemento contém um url para o video, mas após tentar visualizar este url, foi percebido que não é possível obter o vídeo a partir deste. Sendo assim foi investigada a plataforma vimeo, esta é a plataforma que contém todos os videos de produtos, pelo que para cada um é gerado um id unico e este poderá ser acedido através do url geral da plataforma seguido do id do video. Este id está também colocado no elemento iframe, pelo que este é obtido e acrescentado ao url da plataforma conseguindo assim guardar todos os videos de produtos.

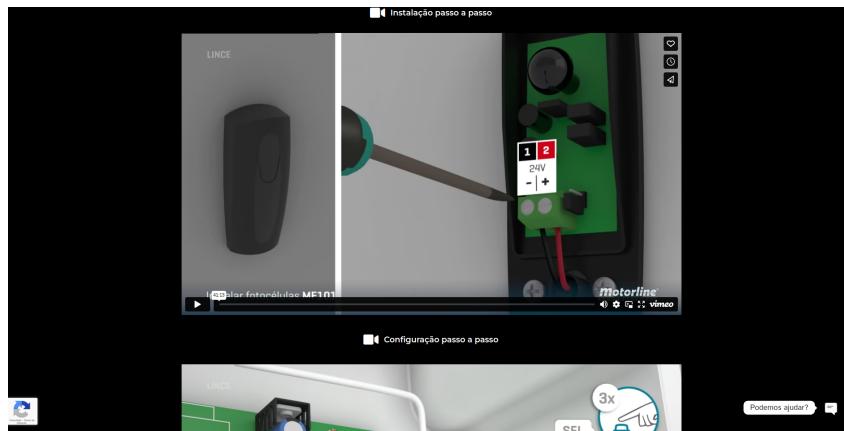


Figura 4.7: Página de detalhes de produto, secção de videos

4.1.1.1 Implementação no website

Após se verificar que eram obtidos pelo menos 80% dos produtos totais foi então decidido testar no website. Para isto foi utilizada a biblioteca requests, com a qual é realizado um pedido get a cada url necessário para se obter a página web. Assim que o código foi corrido e a resposta analisada foi percebido que o website bloqueia este tipo de solução recebendo a resposta demonstrada pela figura 4.8

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<p>Additionally, a 403 Forbidden
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
```

Figura 4.8: Resposta obtida quando o pedido ao url da página web

Através da resposta obtida foi então percebido que seria necessário alterar a abordagem visto que a abordagem anterior não seria possível utilizar. A abordagem opcional a seguir seria simular a ação humana abrindo um navegador e pesquisando pelo url desejado.

Após uma investigação foi descoberto que existem ferramentas que permitem controlar o dispositivo onde correm impedindo a utilização deste enquanto se encontram a correr, assim como ferramentas que apenas recebem o navegador a utilizar e abrem uma nova janela deste navegador para realizar a pesquisa. Visto que o processo de obter os produtos seria demorado, foi optado pela segunda opção visto que seria possível continuar com trabalho em paralelo com a obtenção de dados. Sendo assim a ferramenta mais recomendada para realizar esta operação é a biblioteca selenium, esta biblioteca permite realizar exatamente o processo referido anteriormente com a possibilidade de escalar com multi threading, permitindo abrir diversas janelas do navegador simultaneamente, diminuindo drasticamente o tempo de execução para obter os dados, esta funcionalidade não foi explorada devido a limitações de hardware, mas seria uma importante implementação futura.

Utilizando a biblioteca selenium foi primeiramente indicado qual o navegador a utilizar, neste caso foi escolhido o chrome devido a este já estar instalado no dispositivo. Após se indicar qual o navegador a utilizar, é necessário para cada página indicar qual elemento esperar que carregue, pois assim que a pesquisa é efetuada a página poderá demorar a carregar pelo quem se deverá indicar a espera pelo elemento que se deseja obter. Visto que a página carregada poderá não conter o elemento a obter foi então implementado um tempo de espera máximo de 5 segundos, assim que este tempo expira a operação é abortada e o url é adicionado à lista de urls com erros.

4.1.1.2 Melhoria de implementação

Após se completar o processo foi então decidido melhorar a implementação resolvendo os erros encontrados em urls específicos. De forma a perceber exatamente quais os urls que possuem erros foi então direcionado os dados obtidos para um ficheiro json. Sendo assim os urls com erros eram os indicados na figura 4.9.

```
"misses": [
    "https://motorline.pt/produto/acessorios-para-portoes-de-correr/",
    "https://motorline.pt/produto/acessorios-portoes-garagem/",
    "https://motorline.pt/produto/zuma/",
    "https://motorline.pt/produto/acessorios-barreira-eletromecanica/",
    "https://motorline.pt/produto/acessorios-para-portas-de-vidro/",
    "https://motorline.pt/produto/adaptador-tub/",
    "https://motorline.pt/produto/acessorios-motores-enrolar/",
    "https://motorline.pt/produto/cortina-corta-fogo-flama/",
    "https://motorline.pt/produto/acessorios-para-toldos/",
    "https://motorline.pt/produto(mb17)/",
    "https://motorline.pt/produto(mbn25)/",
    "https://motorline.pt/produto(mim1100u)/",
    "https://motorline.pt/produto/acessorios-controlo-de-acessos/",
    "https://motorline.pt/produto(stop)/",
    "https://motorline.pt/produto/acessorios-fotocelulas/"
],
```

Figura 4.9:Urls com erro primeira interação

Após uma primeira análise foi possível perceber que grande maioria dos erros provém de urls de acessórios de produtos, isto deve-se ao facto de os acessórios de produtos encontrarem-se na página de produtos de subcategoria e serem tratados como um url de detalhes de produto, sendo assim sempre que se trata de um url de acessórios seria necessário correr código para obter dados de destes ao invés de detalhes de produtos. Para desenvolver este código foi primeiramente analisada a página de acessórios de produtos(Figura 4.10), esta página contém para cada acessório um elemento do tipo artigo o qual contém uma imagem, titulo e descrição, esta descrição por vezes contém urls para os produtos aos quais este acessório se refere, pelo que sempre que estes urls são detetados, os nomes dos produtos são guardados para futuramente realizar a ligação entre os acessórios e os produtos, visto que não existem produtos com nomes iguais. Foi percebido que nos urls a palavra accessórios está sempre contida pelo que sempre que esta é detetada em um url é corrido o código referente à obtenção de acessórios.

Acessórios para Motores de Batente

Batente Lince

Batente em alumínio, para aplicação em toda a linha **LINCE**, para limitação na abertura, evitando assim a utilização de batente no portão.



RL180

Acessório que permite a abertura 180° de um motor **SUBWING**.



KIT BATENTE

Batente mecânico para um motor **SUBWING**.



Figura 4.10: Exemplo de página de acessórios

Após correr o novo código criado foi percebido que a quantidade de urls com erros diminuiu, mas existiam produtos com páginas de detalhes de produtos comuns pelo que estas foram analisadas e foi percebido que um erro ocorria devido a por vezes as páginas não conterem vídeos ou imagens de documentação, sendo que o código foi alterado para apenas obter estes dados se os elementos existirem na página. Após correr novamente o código foi percebido que a quantidade de falhas obtidas diminuiu drásticamente (Figura 4.11). Mas mesmo assim ainda existiam 3 falhas a ocorrer e após uma análise foi percebido que estas falhas estavam a ocorrer devido a:

1. Uma página de subcategoria de produtos conter um serviço;
2. Um produto conter uma página de detalhes de produto com subprodutos;
3. Existir uma página de adaptadores de produtos;
4. Um produto conter uma página de detalhes diferente das demais;

```
"misses": [
    "https://motorline.pt/produtos/elettronica-e-controlo/casa-inteligente/",
    "https://motorline.pt/produto/zuma/",
    "https://motorline.pt/produto/adaptador-tub/",
    "https://motorline.pt/produto/cortina-corta-fogo-flama/"
],
```

Figura 4.11: Exemplo de página de acessórios

Visto que a página de adaptadores de produtos segue uma estrutura similar à estrutura dos acessórios este foi o primeiro a ser abordado e resolvido, correndo o código de obter detalhes de acessórios sempre a palavra acessórios ou adaptadores se encontra no url. De seguida foi percebido que para resolver o problema de existirem serviços e subprodutos o diagrama de base de entidade relação teria de ser alterado pelo que primeiramente foi resolvido o problema do produto que contém uma página de detalhes diferente das demais.

Este produto para além da dificuldade de ser uma página completamente diferente as informações encontram-se espalhadas pela página (Figura 4.12), pelo que estas deveriam

ser combinadas para construir os detalhes do produto. Após a obtenção destes dados foi percebido que as imagens do produto têm dados escritos por cima que não se encontram na imagem original, sendo assim foi decidido em conversa com o cliente que estas seriam obtidas com screenshots e guardadas num servidor de imagens.



Figura 4.12: Exemplo de página de produto incomum

De forma a resolver o problema de existirem produtos com subprodutos, foi primeiramente alterada a base de dados adicionando uma nova ligação na tabela produtos para ela própria sendo assim um subproduto possui a chave estrangeira de produto principal preenchida. Após a alteração na base de dados, foi então seguido para a obtenção dos dados do catálogo. Para os dados foi primeiramente obtido todos os dados do produto principal e de seguida os dados específicos a cada subproduto, sendo que a organização do produto principal é igual aos restantes mas com um dado extra com os subprodutos.

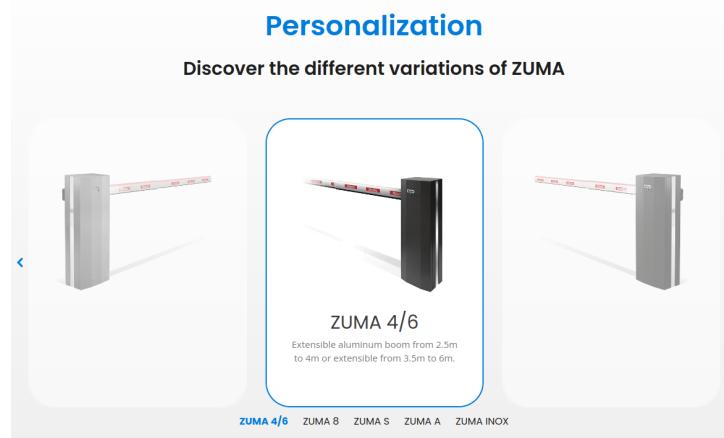


Figura 4.13: Exemplo de página de produto com subprodutos

O último problema a solucionar é a existência de serviços, iniciou-se então pela análise do serviço existente, este possui então descrição e imagens como os produtos, mas possui também vídeos direcionados a plataformas diferentes, registo de atualizações do serviço planos de pagamento de serviços, sendo que cada plano contém diversas ofertas e por fim produtos do serviço. Possuindo então a estrutura de um serviço foi adicionado à base de dados as tabelas de serviço, planos do serviço e suas ofertas, vídeos do serviço, informações de atualizações dos serviços, imagens do serviço e a ligação à tabela produtos identificando os produtos do serviço, sendo também aqui identificado a necessidade de manter guardado os links para todas as imagens e vídeos na própria base de dados, sendo estas tabelas acrescentadas também aos produtos. Após isto foi percebido que existem dados que não são necessários na descrição pelo que o cliente indicou que seria mais indicado guardar estes dados como screenshots.

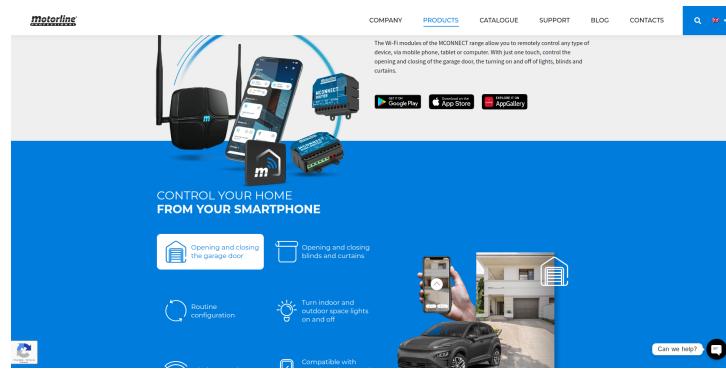


Figura 4.14: Exemplo de página de serviço

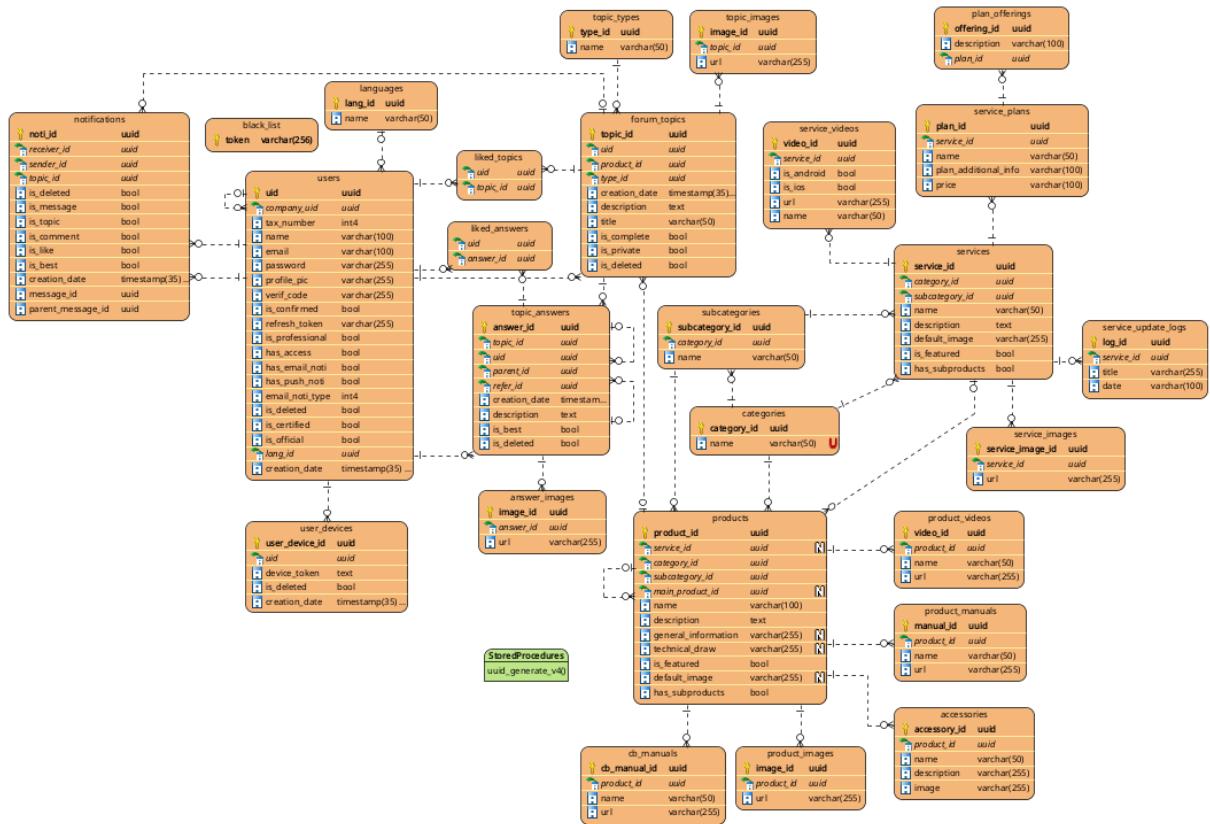


Figura 4.15: atualização da base de dados

4.1.1.3 Armazenamento de dados

Após se obter os dados dos produtos, é necessário guardar estes na base de dados para disponibilizar para a sua utilização no backend. Para realizar esta operação existem duas opções, criar um serviço para inserir produtos e realizar um pedido a este serviço, ou então conectar diretamente o web scraper à base de dados. Visto que não seria de grande interesse conectar diretamente à base de dados, foi decidido criar um serviço que recebe um produto e o insere na base de dados. O grande problema que surgiu com esta solução é que os pedidos ocorrem de forma sequencial, mas com pouco tempo de espera entre estes, o que levava a que o limite máximo de conexões com a base de dados fosse extrapolado. Isto acontece porque para cada serviço chamado é criado uma nova conexão à base de dados, todas as operações são realizadas e por fim a conexão é terminada, mas enquanto estas operações estão a decorrer, o servidor poderá receber mais pedidos, o que leva a que mais conexões sejam criadas, atingindo assim rapidamente o limite de conexões da base de dados. Como solução para este problema foi acrescentado uma espera de 0.5 segundos a cada pedido. A inserção de serviços decorreu com o mesmo processo.

A inserção de categorias decorre enviando as categorias a inserir em um array, sendo inseridas todas num pedido, já as subcategorias, visto que não se sabe o id da categoria referente, foi utilizado o nome da categoria pois este é único, sendo assim é enviado o nome da categoria e as subcategorias referentes, sendo assim todas inseridas com a referência para a sua categoria.

4.2 Backend

Após o desenvolvimento do webscraper, foi então procedido ao desenvolvimento do suporte backend do projeto.

4.2.1 Organização do projeto

Antes de iniciar a implementação, foi definido qual a estrutura de projeto a seguir, pelo que a escolhida foi MVC devido a ser a mais comum e estabelecida. Sendo assim a organização do projeto segue a seguinte estrutura:

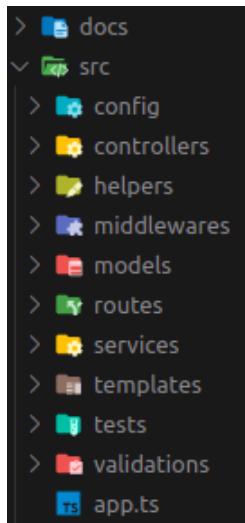


Figura 4.16: Exemplo de página de produto incomum

- **docs** - Documentação gerada;
- **src** - Base de todo o projeto;
- **config** - Ficheiros de configuração do projeto;
- **controllers** - Controladores para cada pedido;
- **helpers** - Ficheiros com funções gerais utilizadas regularmente;
- **middlewares** - Ficheiros com os middlewares da api;
- **models** - Classes criadas para representação de base de dados e para as entidades de resposta;
- **routes** - Rotas existentes;
- **services** - Serviços para cada pedido;
- **templates** - Templates de email a serem enviados;
- **tests** - Testes de código realizados;

- **validations** - Validações a realizar a nível de modelo de negócio e validação de dados;
- **app** - Ficheiro de início do projeto;

4.2.2 Definição de rotas base

Após a definição da estrutura do projeto foi então definido as rotas base a existir, estas são rotas que se referem a cada tipo de utilizador. Para melhor organização destas rotas e aplicação de regras foram definidos 3 routers, user para utilizadores se sessão, professional para técnicos e company para empresas. De forma a definir para o projeto qual o router a utilizar em cada pedido foi então definido que:

- **http://baseurl:port/professional** - Encaminhar para router de técnicos;
- **http://baseurl:port/company** - Encaminhar para router de empresas;
- **Restantes** - Encaminhar para router de user;

4.2.3 Middlewares

Um middleware comporta-se como uma ligação entre porções de código, sendo possível este também executar código.

4.2.3.1 Linguagem

O bem essencial em uma boa comunicação entre duas partes é a utilização da mesma linguagem, sendo assim foi necessário perceber qual a linguagem a utilizar quando se responde a um pedido. Para este fim foi então desenvolvido um middleware, o objetivo deste é verificar se existe a chave language no cabeçalho do pedido, caso esta exista é então obtida a linguagem e guardada nas variáveis locais do pedido. Em caso de esta tag não existir, foi então decidido que a aplicação responderá em português por omissão, este valor poderá ser futuramente alterado de forma simples.

4.2.3.2 Autenticação

De forma a assegurar a autenticação dos utilizadores que necessitam desta foi então decidido implementar JsonWebToken, este tipo de autenticação baseia-se em a utilização de tokens com tempo de expiração, sendo que enquanto o token estiver válido, o utilizador poderá realizar pedidos e assim que este token expirar este terá de se autenticar novamente para obter um novo token. A utilização de tokens permite também assegurar que os pedidos são realizados com tokens gerados pela api através de utilização de uma chave de assinatura de token, impedindo assim a utilização de tokens gerados por utilizadores.

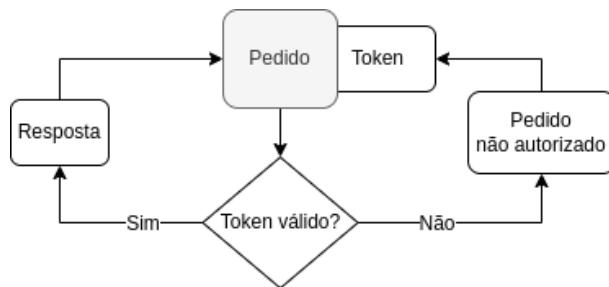


Figura 4.17: Exemplo de página de produto incomum

A grande valia da utilização a técnica de autenticação mencionada anteriormente é a segurança desta, mas este nível de segurança leva a que as aplicações que não necessitam de um nível de segurança muito alto se tornem impráticas. Isto acontece porque estes tokens têm geralmente uma duração muito curta como por exemplo 15 minutos, e sempre que um token de sessão expira o utilizador teria de realizar novamente o login.

A solução deste problema sem a perda de segurança significativa veio pelo meio da utilização de tokens de duração maior em conjunto com os tokens de duração curta, sendo que enquanto o token de grande duração estiver válido, novos tokens de curta duração são gerados para o utilizador nunca perdendo assim a sua sessão. Estes tokens de grande duração tem por nome tokens de refresh e os tokens de curta duração têm por nome tokens de sessão. Sempre que o utilizador termina a sua sessão o token de refresh deverá ser apagado.

Sempre que um utilizador realiza um pedido o seu token de sessão deverá ser validado, caso este seja válido, o seu token de refresh deverá também ser validado e apenas após isto o utilizador estará autenticado. Caso o token de sessão ou de refresh esteja expirado, este continuará a estar sem autorização para realizar o pedido, mas poderá pedir um novo token de sessão enquanto o seu token de refresh estiver válido, isto acontece sem realizar novamente o login e sem o utilizador perceber.

Além das funcionalidades atrás mencionadas é possível também associar dados em formato json a um token jwt, esta funcionalidade foi utilizada para enviar o id do utilizador a qual pertence este token e também o cargo do mesmo.

4.2.3.3 Validação de Papel

Com finalidade de garantir que apenas empresas podem realizar os pedidos de empresas e apenas técnicos e empresas podem realizar os pedidos de técnicos foi então criado um middleware que valida se o utilizador que realizou o pedido tem permissões para o mesmo. Este middleware interliga-se com o middleware anterior pois como mencionado o cargo do utilizador em questão é enviado no token, sendo assim é obtido este cargo e realizada uma comparação, com o cargo desejado. Para isto foram criados 2 middlewares diferentes, um valida o cargo de empresas e o outro o cargo de técnicos. Visto que as empresas podem realizar operações de técnicos então no middleware de técnicos é verificado se o token corresponde a um utilizador empresa ou a um utilizador técnico, já no middleware de validação de empresa é verificado se o utilizador tem cargo de empresa.

4.2.4 Controllers

Assim que um pedido consegue passar por todos os middlewares sem ser impedido, este é então redirecionado para um controller. Um controller é

4.2.4.1 Estruturação dos controllers

De forma a evitar que o código destes controllers varie em termos de estrutura, foi então decidido desenhar uma estrutura de controller e aplicar esta perante o demais código. A estrutura deste segue as seguintes etapas:

1. Obter dados do pedido
2. Validar se os dados obrigatórios são obtidos
3. Validar o pedido perante o modelo de negócio
4. Executar a lógica do pedido
5. Formular a resposta e enviar
6. Em caso de erro este deverá ser capturado e processado de forma a enviar um erro para o utilizador

Para garantir que esta estrutura sempre será aplicada foram utilizados snippets de código que permitem criar um modelo de estrutura de código sendo apenas necessário escrever a palavra chave e toda a estrutura é escrita, necessitando de seguida de efetuar as alterações necessárias perante o contexto.

4.2.4.2 Execução da lógica de negócio

A execução da lógica de negócio passa por direcionar os dados para a ação correta, sendo que esta ação geralmente resulta em uma operação de base de dados. Inicialmente foi desenvolvida toda a validação de código e todas as operações de base de dados diretamente na execução da lógica de negócio, após uma revisão desta organização de código com o professor orientador, foi decidido separar esta funcionalidades, surgindo assim a componente de validação de dados, a componente de operações de base de dados e por fim a componente de lógica de negócio que implementa a componente de operações de base de dados. Sendo assim de forma a evitar que estas operações sobre a base de dados estejam em conjunto com o direcionamento dos dados, foram então criados modelos para cada tabela. Cada modelo contém um conjunto de operações sobre a sua tabela correspondente, estas operações estão contidas sobre métodos que podem receber dados para executar na operação e devolver a resposta da mesma.

4.2.4.3 Validação dos dados

A validação dos dados é necessária de forma a evitar erros a nível de servidor com dados em falta e também para aplicar as regras de negócio, garantindo assim que estas são cumpridas. Para realizar estas validações é primeiramente verificado que todos os dados são recebidos, de seguida estes são enviados para um validador. O validador executa todas as verificações necessárias a nível de regras de negócio e em caso de alguma regra não ser cumprida, é então atirado um erro.

4.2.4.4 Formulação da resposta

Assim como mencionado anteriormente o bem mais importante numa boa comunicação é a utilização da mesma linguagem, sendo assim a resposta do servidor deverá utilizar a linguagem indicada pelo utilizador. De forma a realizar esta tradução foi utilizado o mesmo conceito que é utilizado para a tradução de aplicações android onde nestas é criado um ficheiro que contém um conjunto de chaves e a cada chave corresponde um texto, para cada tradução estas chaves têm de existir de forma a ser possível obter o texto correto para cada chave. Sendo assim foi utilizado um ficheiro json contendo as chaves das linguagens suportadas, a cada linguagem corresponde um conjunto de outras chaves que contém todas as traduções necessárias, utilizando neste caso numeração, em vez de palavras. Esta abordagem permite que de forma fácil futuramente seja possível adicionar outras linguagens ao servidor.

Para dar suporte a este ficheiro foi criada uma operação que recebe a chave desejada e a linguagem desejada, devolvendo o texto correspondente, sendo assim na Formulação da resposta esta operação é executada indicando a chave da resposta a enviar e a linguagem desejada obtendo o texto traduzido, sendo então este devolvido para o utilizador.

4.2.4.5 Processamento de erros

Visto que não é de interesse enviar para o utilizador erros do próprio servidor, foi então decidido controlar estes, para isso foi criado um erro customizado, tendo este por base o erro da própria linguagem. Este erro recebe por parâmetro o código da tradução da mensagem de erro. Esta abordagem permite também evitar que sempre que um erro é lançado o sistema pare. Mesmo com esta abordagem acontece que sempre que um erro é lançado por base de dados, erro de código ou de biblioteca, o erro original é chamado, pelo que foi decidido que sempre que é detetado um erro que não é do tipo do erro customizado, então será devolvido um erro com mensagem de erro de servidor evitando que dados sensíveis e desnecessários para o utilizador sejam devolvidos.

4.2.4.6 Envio de emails

Como mencionado na apresentação da ferramenta tabular, esta não permite a utilização de acentuação na escrita de emails, sendo assim estes erros necessitam de ser resolvidos. Para permitir que os dados dos emails sejam personalizáveis, como por exemplo dados de utilizador e link para clicar, estes emails são colocados dentro de métodos que recebem por parâmetro os dados personalizáveis, sendo estes então colocados dentro do html do email, este método por fim devolve em string o html do email a enviar.

Para o envio de emails é necessário um servidor, sendo assim para vias de teste foi utilizado um servidor gratuito de email sendo este hospedado por DIZER QUEM HOSPEDAVA, após a fase de testes foi então alterado para o servidor de email da empresa permitindo assim que este email seja identificado como empresa Motorline.

A configuração do servidor de emails do nodemailer é realizada através de uma chamada ao objeto de servidor indicando as configurações do mesmo que estão no ficheiro .env, esta chamada ao servidor por sua vez devolve um objeto que fornece métodos para enviar emails, sendo este então criado e utilizado sempre que se deseja enviar emails no servidor.

De forma a evitar que sempre que se deseja enviar um email seja necessário indicar todos os dados, foi então criado um método que cria e devolve o objeto de servidor sempre que necessário. Sendo assim sempre que se deseja enviar um email é então primeiramente obtido o objeto do servidor, de seguida é invocado o método para obter o conteúdo html do email e por fim é utilizado o objeto do servidor para chamar o método de envio de emails no qual se terá de indicar o email do destinatário, o assunto e o conteúdo do email.

4.2.4.7 Logging de erros

A identificação dos erros no servidor é difícil, pois uma vez que os erros são tratados, os dados referentes aos mesmos não são guardados ou utilizados, para resolver este problema foi então decidido que sempre que um erro que não é customizado é detetado, é registado um log, este contém informações sobre o pedido como data e hora, dados recebidos e assim como também a descrição original do erro e serviço referente.

Para implementar esta solução foi então primeiramente criado um middleware que sempre que deteta erros dentro do serviço executa um método. Este método por sua vez obtém os dados referentes ao pedido realizado, sendo estes a data e hora, os dados recebidos e o serviço pedido. Após se obter os dados do pedido é obtida a mensagem do erro, sendo então estes dados acrescentados em uma nova linha no ficheiro de erros do servidor.

4.2.4.8 Logging de pedidos com morgan

Para realizar o logging de pedidos a serviços foi então utilizado o morgan, esta ferramenta apenas precisa que seja indicado o ficheiro onde escrever os logs, sendo utilizado um ficheiro chamado log, este também precisa de saber qual o tipo de log a realizar, sendo estes tipos indicados pela ferramenta, o tipo utilizado foi o combinado, este é o tipo de log mais completo da ferramenta, visto que é o que obtém mais dados, sendo estes, a data e hora do pedido, o tipo de pedido, o serviço pedido, os dados recebidos, a resposta devolvida e também a descrição do sistema utilizado.

4.2.4.9 Agendamento de tarefas

Utilizando a ferramenta node-cron foi inicialmente programado para este enviar o relatório de notificações todos os dias às 22 horas, para realizar esta configuração é necessário indicar primeiramente a programação de horário de envio, para isso é utilizada a estrutura segundo, minuto, hora, dia do mês, mês, dia da semana. Visto que o objetivo é às 22 horas os segundos e minutos foram indicados como 0 e as horas foram indicadas como 22, já o restante foi indicado com o símbolo "*" que indica que o processo deverá ocorrer em todas as instâncias dos restantes valores, significando assim todos os dias. A configuração final obtida foi então "0 0 22 * * *".

Quando o método do agendamento é invocado, é então obtido todos os utilizadores com a configuração de relatório de notificações ativa, sendo de seguida para cada um destes obtidas todas as notificações do dia. De seguida é criado o objeto de servidor e invocado o método para obter o email de notificações enviando todas as notificações do utilizador para o mesmo. Por fim é enviado o email para o utilizador com todas as suas notificações e um link para aceder rapidamente ao ecrã de notificações.

4.2.4.10 Cifragem de passwords

Aquando o registo de clientes é indicada a password, pelo que esta deverá ser guardada cifrada. Para isto é utilizada a biblioteca bcrypt, sendo que no processo de registo, antes do envio dos dados do utilizador para o servidor, é invocado o método de cifragem da password, indicando como salt o valor 8, após a execução é então obtida a password cifrada, sendo esta enviada para a base de dados com conjunto com os restantes dados.

4.2.4.11 Cifragem de configurações do servidor

Para a cifragem das configurações do servidor, foi então executado o comando de cifragem da biblioteca secure-env, sendo pedido a password da mesma, após esta indicação, foi criado o ficheiro cifrado, pelo que o ficheiro original deverá ser eliminado ou em caso de necessidade guardado em um local seguro de forma a evitar que durante algum ataque seja possível obter os dados do mesmo.

Após a cifragem do ficheiro o servidor foi configurado para aquando a sua inicialização, este deverá pedir para o utilizador indicar a password do mesmo, sendo esta a indicada na cifragem do ficheiro de configurações do servidor, para esta configuração foi utilizada a biblioteca readline-sync no modo de password. Quando o utilizador indica a password em caso de esta estar errada o servidor irá falhar, pois a biblioteca de cifragem irá tentar decifrar o ficheiro de configurações e irá falhar não concluindo o processo de inicialização, caso contrário, este continuará o processo de inicialização, sendo o primeiro passo a utilização da password para decifrar o ficheiro de configurações e carregar as variáveis de ambiente.

4.2.4.12 Documentação Typedoc

Durante a implementação desta ferramenta foi detetado que a categorização de documentação não se encontrava funcional devido a um problema encontrado pelos desenvolvedores da ferramenta, foi decidido então reduzir a versão da mesma para uma com a funcionalidade ativada, mas esta não se encontrava compatível com a versão mais atualizada de typescript, pelo que não foi possível explorar esta funcionalidade. Para contornar o problema foi então decidido explorar outra funcionalidade menos utilizada da ferramenta, esta funcionalidade permite converter qualquer documentação em módulos, estes módulos podem então ser categorizados, o problema destes módulos é que cria um modelo genérico do código não sendo facilmente identificado as tipagens de scripts. Estes módulos permitem também a categorização dos mesmos permitindo assim um nível de organização da documentação gerada.

4.2.4.13 Documentação Swagger

Apesar da ferramenta disponibilizar a geração automática de documentação a partir de comentários de código, foram encontrados alguns problemas com esta funcionalidade, acabando por não gerar a documentação, pelo que foi optado por manter a documentação manualmente com o ficheiro json. Esta ferramenta oferece diversas funcionalidades como autenticação, definição de estruturas de dados para os serviços e exemplos de respostas para os mesmos, ambas estas funcionalidades foram exploradas conseguindo assim um bom suporte de documentação para qualquer utilizador.

Figura 4.18: Documentação swagger

Figura 4.19: Exemplo de documentação de serviço

4.3 Frontend

Após o desenvolvimento do suporte backend, foi então prosseguido para o desenvolvimento frontend do projeto.

4.3.1 Organização do projeto

Assim como o backend o modelo a seguir para o frontend foi o MVC.

Como recomendado de boas práticas de código limpo da framework as cores do tema da aplicação foram todas colocadas em um ficheiro separado de forma a garantir que a troca de tema da aplicação é facilitada. Outras aplicações de boas práticas de código limpo foram, sempre que possível partitionar o código das páginas em vários widgets de forma a ser de fácil navegação e também a criação de widgets reutilizaveis de forma a evitar a repetição de código e também para agilizar o desenvolvimento de código. Sendo assim a estrutura do projeto foi organizada da seguinte forma:

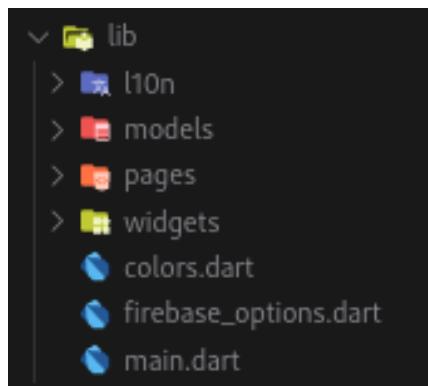


Figura 4.20: Organização do projeto

- **l10n** - Traduções da aplicação;
- **models** - Modelos de classes como handlers, helpers, providers, entre outros;
- **pages** - Páginas da aplicação;
- **widgets** - Widgets referentes às páginas;

4.3.2 Extensions

A linguagem de programação dart, assim como outras linguagens de programação orientadas as objetos permite alterar e adicionar métodos aos objetos base da linguagem, para realizar isso são criadas extensões do objeto, neste caso foi criada uma extensão para o objeto string de forma a facilmente capitalizar um texto.

4.3.3 Handlers

Os handlers são porções de código que permitem a execução de código perante um evento como por exemplo realizar uma ação perante um clique no ecrã, neste caso os handlers foram utilizados para detetar o estado da aplicação e realizar ações perante estes estados. Os estados da aplicação referem-se a se a aplicação se encontra em primeiro plano, segundo plano, a resumir ou então desligada. Com estes handlers é possível alterar o funcionamento da aplicação perante estes estados, como por exemplo tratar de notificações da aplicação.

4.3.4 Providers

Os providers são classes criadas para ajudar com comunicações externas, neste caso chamdas à API, estes providers foram criados de acordo com os diversos modelos de dados que se recebem, como por exemplo, uma publicação do fórum. Um provider oferece perante um modelo um conjunto de métodos para as diferentes chamadas necessárias à API, utilizando o exemplo anterior, para uma publicação existem métodos para eliminar, adicionar, editar e obter dados, sendo que acada método terá os seus requisitos do serviço que invocam.

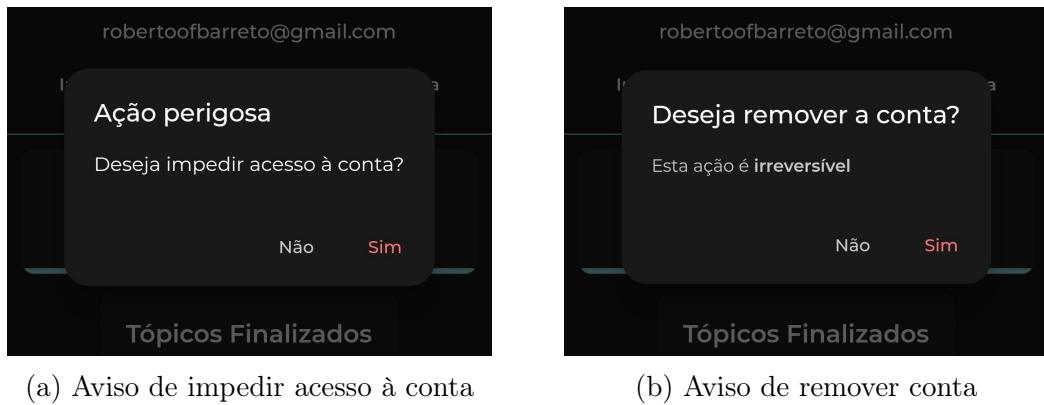
Estes providers automaticamente detetam a linguagem da aplicação e realizam o pedido ao serviço utilizando essa linguagem.

4.3.5 Helpers

Os helpers como o próprio nome indica são classes que ajudam com a realização de uma tarefa, neste caso os helpers, foram utilizados no auxílio de mensagens de notificações, estas mensagens deveriam conter o nome do utilizador e também a ação do mesmo traduzida na linguagem da aplicação, sendo assim foi criada a classe de helper de notificação que contém um método estático para obter a mensagem de uma notificação de acordo com a ação da mesma.

4.3.6 Gestão de utilizadores

Um requisito da aplicação e que apenas as empresas têm a possibilidade de se registarem, pelo que apenas estas poderão registar os seus técnicos. Para isso foi criada a página de gestão de utilizadores apenas acessível às empresas, nestas páginas estas poderão pesquisar pelos seus técnicos, ou registar novos técnicos, sendo que tem-se ser obrigatoriamente indicado o nome, email e tipo de utilizador. Outras ações que este utilizador poderá realizar é a visualizar um técnico onde poderá bloquear o acesso ou até apagar a conta do mesmo, sendo que esta ação é irreversível.



Sempre que um utilizador sem acesso ou com a conta apagada efetua o login, este receberá uma mensagem de erro mencionando que a sua este não possui acesso à conta impedindo assim o acesso.

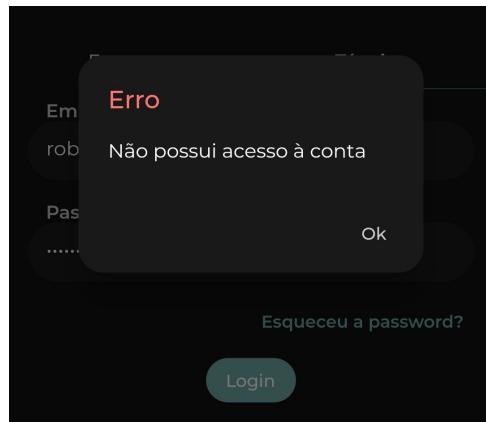


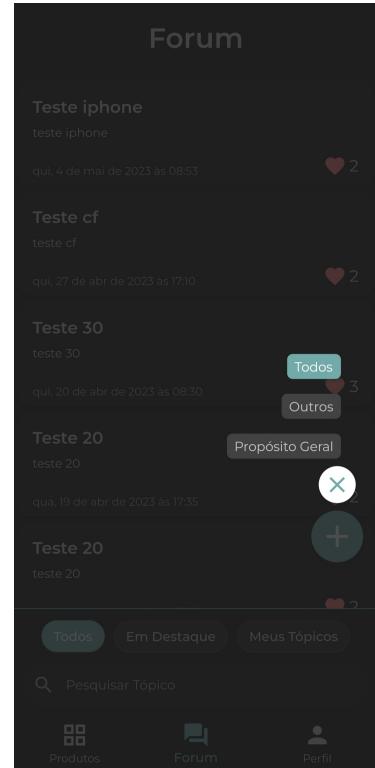
Figura 4.22: Aviso de login a conta sem acesso

4.3.7 Fórum

O desenvolvimento da página de fórum trouxe diversas dificuldades entre elas a gestão de filtros e pesquisas e também a mostragem de publicações e atualização das mesmas.



(a) Página de forum



(b) Filtragem de tipo

4.3.7.1 Filtragem de tópicos

O grande problema com a filtragem dos tópicos é que existem 3 tipos de filtros, o filtro de categoria de tópico, o filtro de tipo de tópico e o filtro de pesquisa.

Uma vez que algum filtro seja alterado, os filtros seguintes deverão ser novamente executados de forma a garantir que todos estes se encontram aplicados, inicialmente este tipo de filtragem não era executado, o que levava a problemas como por exemplo sempre que se efetua uma pesquisa, esta não era efetuada sobre as publicações filtradas o que levava a que a pesquisa fosse efetuada por todas as publicações.

Outro problema encontrado era na troca de categoria, por vezes acontecia que os filtros de categoria adicionavam-se o que levava a que estes não mostrassem publicações.

Sendo assim foram criados métodos para auxilio na filtragem, assim como também uma prioridade, sendo que a cada método invoca o método seguinte em forma de encadeação. Primeiramente aplica-se o filtro de categoria, de seguida este envia o resultado da filtragem para o método de filtragem por tipo e por fim se existir algum tipo de pesquisa os tópicos serão filtrados pela mesma.

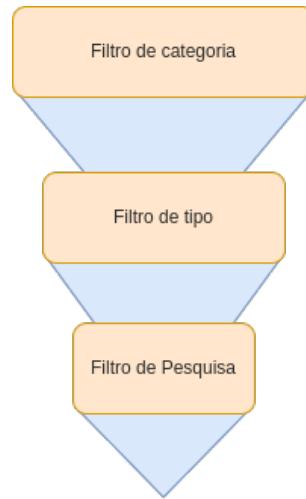


Figura 4.24: Filtragem do forum

4.3.7.2 Carregamento de tópicos

Inicialmente o carregamento de tópicos fazia-se por inteiro, desde carregamento de todos os tópicos, até ao carregamento de todos os dados dos mesmos, pois visto que não existiam muitos tópicos este não seria um problema para a API, mas conforme os testes foram sendo realizados a quantidade de tópicos existentes foi sendo incrementada, sendo possível visualizar o tempo de demora de resposta do servidor a aumentar, assim como também a performance da aplicação no fórum a piorar.

De forma a resolver este problema primeiramente pensou-se em uma técnica de sliding window no total, o problema é que esta solução iria levar a que se o utilizador desejar voltar para trás nos tópicos a partir de uma quantidade escolhida de tópicos estes teriam de ser carregados novamente, o que não levaria a uma boa experiência de utilização.

Para resolver este problema foi utilizada a ideia de sliding window, mas os tópicos iriam se manter carregados, sendo que a própria framework consegue através da lista retirar de renderização os tópicos que o utilizador não consegue ver.

Esta solução foi implementada utilizando 3 valores, quantidade de tópicos a obter, índice inicial e data do primeiro tópico. O valor de quantidade de tópicos a obter, inicialmente 10, permite limitar a quantidade de tópicos que a API irá processar reduzindo o tempo de resposta, o índice inicial, permite indicar qual o índice do primeiro elemento que se deseja obter da lista e a data do primeiro tópico permite manter uma referência temporal para obter tópicos, garantindo assim que a lista que se está a visualizar é sempre a mesma.

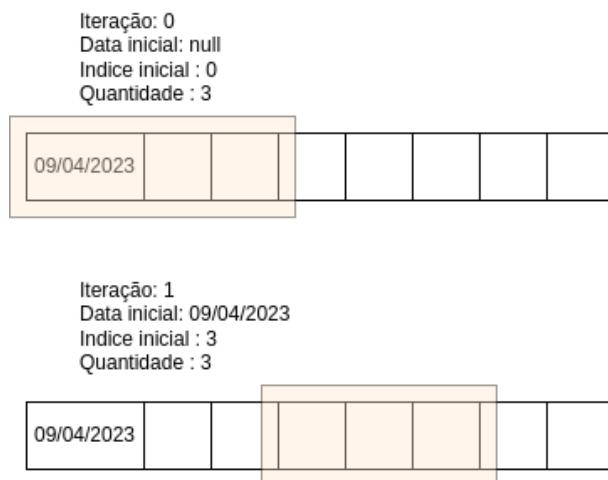


Figura 4.25: Carregamento de tópicos

Foi também reduzido a quantidade de dados carregados por cada tópico, sendo assim apenas os comentários diretos à publicação são carregados não carregando as respostas a estes, o que também contribuiu para a melhoria de performance.

Por fim sempre que o utilizador alcança o fim da lista de tópicos este poderá deslizar para carregar mais tópicos.

4.3.7.3 Detalhes de tópico

A página de detalhes de tópico sofreu os mesmos problemas que a página anterior sendo necessário aplicar a mesma solução sobre os comentários de tópico e também sobre as respostas ao mesmo. Sendo assim são carregados os primeiros 10 comentários e por fim mostrado ao utilizador quantos mais comentários existem que poderá carregar sendo o limite 10 comentários.

Estes comentários podem também conter respostas sendo que estas podem ser carregadas também 10 de cada vez conseguindo o utilizador esconder ou mostrar estas.

Um problema que surgiu no desenvolvimento da página de detalhes de tópico foi também o destaque de uma mensagem para por exemplo destacar alguma resposta de um notificação. Este foi um grande problema pois com a nova implementação as mensagens não se encontram carregadas no momento de destacar a mensagem, pelo que é necessário procurar a mesma dentro das mensagens carregadas, expandindo assim as respostas do comentário que contém a mensagem a destacar.

O destacamento de mensagens também continha um erro no qual sempre que algo no ecrã é atualizado, este recarregava a animação pelo que este código teve de ser movido de forma a apenas ser executado no momento de inicialização do ecrã após todos os elementos se encontrarem devidamente carregados.



Figura 4.26: Destaque de mensagens

4.3.8 Firestore

Visto que a desenvolvedora do Flutter e do Firebase é a mesma, esta disponibilizou recursos que facilitam a utilização desta ferramenta pelo Flutter, sendo assim todas as imagens e vídeos de utilizadores, publicações e comentários são guardadas diretamente da aplicação para o firestore, assim como o acesso às mesmas é realizado diretamente.

Para permitir este tipo de acesso o Firebase disponibiliza de um configurador de projeto que permite através do terminal configurar a ligação entre o projeto e o servidor do firebase, sendo no final apenas necessário importar a biblioteca do serviço do firebase que se deseja invocar a classe do mesmo sempre que se deseja realizar alguma ação.

Os ficheiros ficam então organizados de acordo com o seu contexto, no caso de utilizadores existe a pasta utilizadores, no caso de tópicos existe a pasta tópicos e no caso de comentários existe a pasta comentários.

Dentro dos utilizadores como cada utilizador apenas contém uma imagem então estas são guardadas com o nome do id do utilizador substituindo a imagem anterior se existir. No caso de tópicos e comentários como podem conter várias imagens e vídeos, então estes são guardados em pastas com ids dos mesmos contendo dentro destas os ficheiros referentes.

The screenshot shows the 'Storage' interface in the Firebase console. At the top, there are tabs for 'Files', 'Rules', 'Usage', and 'Extensões NOVO'. Below the tabs, there's a header with a link to the storage bucket ('gs://install-and-go.appspot.com'), a blue 'Fazer upload do arquivo' (Upload file) button, and a '+' icon. A table lists three folders: 'Products/' (Tamanho: -, Tipo: Pasta, Última modificação: -), 'topics/' (Tamanho: -, Tipo: Pasta, Última modificação: -), and 'users/' (Tamanho: -, Tipo: Pasta, Última modificação: -). Each folder has a checkbox and a three-dot menu icon.

Name	Tamanho	Tipo	Última modificação
Products/	-	Pasta	-
topics/	-	Pasta	-
users/	-	Pasta	-

(a) Raiz do firestore

This screenshot shows the 'Storage' interface again, but now with the path 'gs://install-and-go.appspot.com > topics' selected in the header. The table below lists three sub-folders under 'topics': '1ddd308-9fd5-41ac-8d17-b2d0410486fd/' (Tamanho: -, Tipo: Pasta, Última modificação: -), '256c6ac4-7ce4-495c-a42a-97cd2270f3e3/' (Tamanho: -, Tipo: Pasta, Última modificação: -), and '2c77661f-ded5-488b-943c-d35e667b9194/' (Tamanho: -, Tipo: Pasta, Última modificação: -).

Name	Tamanho	Tipo	Última modificação
1ddd308-9fd5-41ac-8d17-b2d0410486fd/	-	Pasta	-
256c6ac4-7ce4-495c-a42a-97cd2270f3e3/	-	Pasta	-
2c77661f-ded5-488b-943c-d35e667b9194/	-	Pasta	-

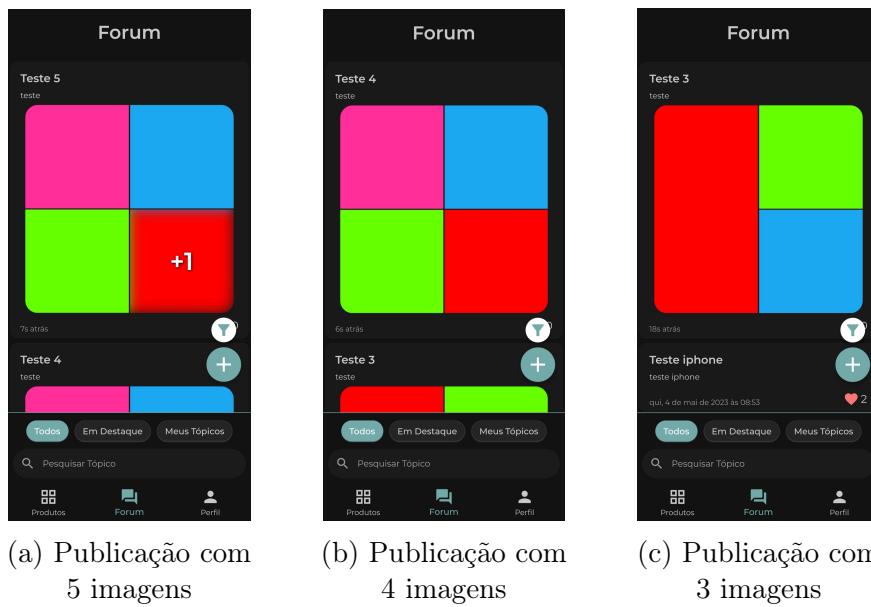
(b) Pasta topics do firestore

4.3.9 Apresentação de Imagens

A apresentação de imagens é definida em 2 níveis, a nível de pré-visualização, por exemplo ver a miniatura da imagem de uma publicação no forum e a nível de detalhe, onde é possível ver a imagem em ponto grande e realizar zoom na mesma.

Para a apresentação de miniatura da imagem foi decidido mostrar até 4 imagens, sendo que acima de 4 imagens seria mostrado apenas 3 imagens significando assim que a quarta imagem indicaria quantas mais imagens existem para mostrar.

Para a implementação da apresentação das miniaturas de imagens, foi utilizada a biblioteca staggered_grid_view, esta biblioteca permite organizar imagens em grelha. Neste contexto desejava-se organizar estas imagens em diferentes aspetos, e disposições, pelo que esta biblioteca permite indicar quantas colunas e linhas existem na grelha ao criar o agrupamento de imagens, sendo assim foi decidido que quando são duas imagens estas dividem a grelha, quando são três imagens a primeira divide metade da grelha e as outras duas dividem a outra metade, quando são 4 ou mais então as 4 imagens dividem a grelha por igual. Quando se encontram existentes mais do que 4 imagens foi então decidido colocar um filtro de desfoque sobre a ultima imagem e colocar por cima desta quantas mais imagens existem para mostrar.



4.3.10 Apresentação de Imagens em carrossel

A apresentação de imagens deverá permitir que o utilizador as visualize em ponto grande conseguindo também realizar diversas ações sobre estas, para isso foram experimentadas diversas bibliotecas, mas nunca se conseguia o comportamento desejado, sendo assim foi decidido criar o próprio carrossel de imagens, sendo que o próprio flutter já disponibiliza um widget para tal.

O ponto de maior dificuldade para este processo foi a implementação de zoom, visto que o flutter não dispõe de widgets para tal, sendo assim foi necessário primeiramente detetar gestos com o detetor de gestos da ferramenta e aplicado um zoom sobre o centro do gesto, os gestos aceites foram o de pinça e o gesto de duplo clique.

O grande problema com esta solução é que visto que é permitido um scroll horizontal de imagens os gestos por vezes poderão não funcionar corretamente, principalmente o gesto de pinça que se efetuado na horizontal poderá resultar em scroll horizontal. Para resolver tal problema foi decidido que quando dois dedos são detetados no ecrã a navegação horizontal fica bloqueada e assim que estes são levantados, a navegação horizontal é ativada novamente.

4.3.11 Carregamento de Imagens

O carregamento de imagens do dispositivo poderá ser realizado por meio de seleção de imagens da galeria, para realizar este tipo de seleção primeiramente foi testada a biblioteca `image_picker`, mas esta biblioteca utiliza o seletor de ficheiro do dispositivo, o problema é que este tipo de seleção permite também a seleção de ficheiros também o que faz com que seja necessário um conjunto de outras verificações para garantir que apenas as imagens são selecionadas o que levaria a uma possível perda de performance e perda de qualidade na experiência de utilização do utilizador.

Sendo assim foi de seguida testada a biblioteca `advance_image_picker`, mas o problema desta biblioteca é que não permite selecionar vídeos, pelo que foi decidido experimentar a biblioteca `wechat_asset_picker`, esta biblioteca cria uma página de seleção de imagens própria para seleção de imagens e vídeos diretamente da galeria, esta permite indicar o limite máximo de seleção e os tipos de ficheiros que o utilizador poderá selecionar de forma a que apenas os tipos aceites são mostrados, acrescentando também que esta biblioteca permite visualizar a tipagem de cada ficheiro no próprio objeto, o único ponto desvantajoso é que não é possível traduzir o botão de confirmação de seleção.

Após a carregar os ficheiros para memória, estes são então enviados para o `firestorage` como mencionado anteriormente.

4.3.12 Videos

A grande dificuldade encontrada com os vídeos foi o facto de ser necessário um comportamento diferente nestes quando se encontram no ecrã de visualização de imagens e vídeos, pois ao contrário de imagens, os vídeos necessitam de um player, sendo sempre necessário verificar qual o tipo de ficheiro antes de carregar o widget do mesmo.

Para resolver o problema de utilizar um player foi primeiramente testada uma biblioteca que permite a utilização do player nativo do dispositivo, ou seja, android utilizaria o player do android e ios utilizaria o player de ios. O grande problema com esta solução é que o player de android tem os botões completamente brancos, sem nenhum tipo de fundo para os destacar significando isto que se um video branco fosse visualizado, o utilizador não conseguiria visualizar os botões do player.

Após o problema anterior foi decidido desenvolver o player próprio, após a implementação de funções como, pausar, resumir video, avançar 5 segundos e voltar 5 segundos, assim como também esconder e mostrar os botões, existiam 2 grandes problemas, sendo estes mostrar o video em ponto grande e retornar para o mesmo tempo de video em ponto pequeno e também o comportamento do player não era completamente fluido.

Após alguma pesquisa foi percebido que o player de ios resolvia os problemas do player de android através da colocação de um fundo nos botões do player. Através da biblioteca appinio, é possível utilizar o player nativo de ios em android, sendo também possível configurar este. Sendo assim foi decidido utilizar o player de ios em ambos os sistemas operativos resolvendo assim todo o problema. Este player também permitiu a resolução de um problema menor que era a visualização de vídeos em ponto grande sendo que dependendo a orientação do vídeo o player alterava a orientação da aplicação voltando à orientação vertical uma vez que se termina a visualização do vídeo em ponto grande.

4.3.13 Links

Uma funcionalidade da aplicação necessária é também a utilização de links, para isto programação mobile oferece duas soluções, deep links e dynamic links. Como mencionado na secção de tecnologias foi decidido implementar a solução de dynamic links da firebase.

Para implementar esta solução primeiramente a nível de backend foi necessário gerar os links, para isto, existem 2 soluções, implementação do firebase no próprio backend ou então uma chamada ao firebase utilizando uma chave de pedido. Primeiramente foi testado a implementação do firebase no próprio backend, mas esta implementação surge com alguns problemas pois existem diversas configurações específicas necessárias, sendo então recomendado pelos colegas de trabalho a chamada ao firebase devido à sua simplicidade.

Sendo assim para a realização de chamadas ao firebase foi utilizado o axios, realizando assim um método post para o serviço de dynamic links do firebase, indicando o prefixo do projeto. De forma a permitir a reutilização deste código, este foi então colocado em um método onde este é chamado indicando o link desejado e os dados a enviar. O link é utilizado para por exemplo como em uma página web, indicar qual página se deseja direcionar o utilizador, já os parâmetros, assim como em um url web, são enviados através do próprio link, sendo assim estes dados são colocados na string do link permitindo a configuração do mesmo perante diversas situações. Por fim a firebase retorna o link criado sendo este colocado no email desejado.

Para a implementação de frontend foi necessário primeiramente importar a biblioteca de dynamic links do firebase, sendo de seguida no código de inicialização da aplicação colocado um método para em caso de a aplicação ser aberta a partir de um link, este o ler. Quando este método lê o link primeiramente este extrai a página indicada pelo link e de seguida extrai a lista de parametros recebidos, sendo então o utilizador redirecionado para a página do link indicando como parametros os dados recebidos no link.

Aquando a testagem da implementação diversas tentativas de abertura de links foram realizadas mas sem sucesso, a grande dificuldade desta implementação é que os links dinamicos não permitem realizar debug, sendo que ou funcionará na totalidade ou então não funcionará levando que seja complicado identificar bugs, pelo que a documentação do serviço foi de grande auxilio pois estava em falta a indicação do nome do pacote da aplicação para android e ios, após a colocação destes dados, tudo seguiu em completo funcionamento.

4.3.14 Notificações

A implementação de notificações revelou-se ser a implementação de maior dificuldade devido a diversos imprevistos na implementação da mesma. Para implementação de notificações foi utilizado o serviço de notificações do firebase, esta implementação surge assim como os links em 2 secções, primeiramente backend e de seguida frontend.

A nível de backend foi necessário utilizar axios para realizar um pedido ao serviço de notificações do firebase, mas assim como na criação dos links o serviço não indica qualquer informações sobre erro, apenas o dispositivo poderá ou não receber a notificação.

Primeiramente foi utilizado o conteúdo a enviar indicado pela documentação do serviço, mas o serviço apenas retornava uma mensagem de erro, se seguida foi pesquisado outras implementações de outros utilizadores e testado, mas novamente surgia um erro, pelo que foi decidido utilizar o conteúdo indicado pelo professor de programação de dispositivos móveis, tendo este conteúdo funcionado sem qualquer indicação de erro.

No conteúdo da notificação é enviado em formato json a mensagem a mostrar na notificação e como as notificações serão sempre referentes a tópicos ou comentários, então é indicado o id do tópico, do comentário e em caso de necessidade o id do comentário pai.

Este processo de notificação foi também aproveitado para direcionar os mesmos dados para notificação de email em caso do utilizador possuir ativo as notificações por email, sendo gerado um link dinâmico com os dados na notificação e colocado no email.

A nível de frontend foi então importada a biblioteca do serviço de notificações do firebase, sendo então esta implementada de acordo com a documentação do mesmo. Sendo que é necessário detetar notificações no iniciar da aplicação, durante a utilização e quando esta se encontra em segundo plano. Sendo assim aproveitado estas deteções para implementar o direcionamento do utilizador para as páginas referentes às notificações, utilizando os dados enviados na notificação.

O grande problema encontrado com as notificações é que o icon de notificação não era mostrado corretamente sendo que ou era mostrado um quadrado escuro ou nenhum icon. A biblioteca de notificações do firebase não permite a customização do icon da mesma, pelo que foi decidido utilizar a biblioteca flutter_local_notifications, esta permite a total customização das notificações sendo então enviado como icon o icon desenhado para a

aplicação, mas mesmo assim as notificações continuavam com o mesmo erro, pelo que foi decidido realizar uma pesquisa sobre o mesmo e foi percebido que android possui um novo sistema para os icons das notificações pelo que é necessário transformar estes icons em preto ou branco com fundo transparente, sendo então de seguida tratados pelo próprio android.

Sendo assim foi realizada a transformação e novamente alterados os icons das notificações, após uma nova testagem foi percebido que mesmo assim apenas o icon da notificação expandida teria sido alterado, após uma nova pesquisa foi percebido que android necessita de 2 icons de notificação para aplicar a ambas as situações de notificação, tendo sido assim resolvido o problema em questão.

Nesta implementação apenas um problema continuou sem resolução, o problema em questão é a abertura de notificações quando a aplicação se encontra terminada, foram testadas várias soluções, mas após a leitura de documentação e de soluções de outros utilizadores, o flutter não permite a reconfiguração do comportamento de notificações quando a aplicação se encontra terminada, pelo que este problema não contém solução de momento. O flutter possui como futura implementação a permissão de reconfiguração do comportamento do click neste tipo de notificação, mas de momento não dispõe de solução.

4.3.15 Permissões

De forma a garantir o acesso à galeria, às notificações e à internet é necessário pedir permissão ao utilizador.

Para pedir permissão à galeria, foi alterado o android manifest para pedir permissão ao utilizador assim que for necessário aceder à galeria. Sendo assim sempre que é executado algum código de acesso à galeria pela primeira vez, o utilizador receberá um alerta a pedir permissão para acesso à galeria, sendo que em caso de recusa não será possível aceder à galeria.

Para acesso à internet, foi realizado o passo anterior, mas esta permissão é pedida no ato de inicialização da aplicação em conjunto com as permissões de notificações, sendo que as permissões de notificações são encarregues da biblioteca de notificações do flutter.

4.3.16 Ios

Após o desenvolvimento completo da aplicação para dispositivos android, foi proposto pela Motorline testar como esta se comportava em ambiente ios, para isso esta disponibilizou acesso a um dispositivo móvel apple, um computador, uma conta de desenvolvedor e um colega de trabalho que já tinha a experiência de desenvolvimento ios com flutter.

Para a compilação primeiramente foi necessário configurar a ferramenta XCode, esta configuração foi realizada em conjunto com o colega de trabalho.

Após o processo de configuração o projeto foi compilado para ios, nesta primeira compilação todas a aplicação funcionava corretamente, sendo apontado pelo colega de trabalho algumas configurações de design comuns em ios como por exemplo os botões de cancelar e confirmar se encontrarem no topo do ecrã e também a explicação de como lidar com navegação uma vez que ios não dispõem de função de voltar para trás.

Para resolver o problema de navegação foram acrescentados botões de navegação para trás em todas as páginas que provêm de páginas principais. Já para implementar a sugestão dos botões de cancelar e confirmar foi então declarado que se o dispositivo em que a aplicação está a correr for um dispositivo apple então estes botões estarão no topo da página, caso contrário ficarão no fundo.

Após esta implementação foi decidido testar as notificações do sistema, pelo que foi percebido que estas não funcionavam. Para isto foi pesquisado qual seria a possível fonte do problema pelo que foi detetado que as permissões poderiam não estar configuradas, para realizar a configuração das mesmas foi necessário atribuir as mesmas permissões de android para ios, mas sendo estas agora implementadas através de um ficheiro com o nome de info.plist.

Após esta adição a aplicação foi compilada novamente, mas mesmo assim as notificações não funcionavam pelo que foi indicado pelo colega de trabalho que no envio do pedido de notificações para o firebase é necessário indicar as configurações de ios também, e foi então procurado na documentação como se enviava as configurações de ios e utilizando estas foi então alterado o pedido de notificações e também o pedido de links dinâmicos, uma vez que é um pedido do mesmo estilo, evitando assim problemas futuros.

Após esta resolução foi testado novamente, mas mesmo assim o problema persistia e após alguma pesquisa foi percebido que as notificações e links têm também de ser configurados no XCode, após esta configuração foi novamente testado e todas as notificações, links de emails e permissões funcionaram como planeado.

5. Análise de resultados

O objetivo do projeto era o desenvolvimento de uma aplicação mobile com foco na área de forum da mesma, devendo este permitir o registo de empresas, gestão de utilizadores pelas mesmas, notificações, comunicações por email, gestão de conta e notificações do utilizador.

5.1 Tarefas

A nível de tarefas o esperado, como mencionado no capítulo 2.4, seria terminar o projeto no final de março, o que não se tornou realidade devido a diversos imprevistos que ocorreram no caminho, sendo o principal imprevisto o atraso no desenvolvimento do webscraper, devido à constante atualização do catálogo o que levava a alterações no conteúdo do website sendo assim necessário alterar o código de leitura do mesmo a cada alteração. Outro grande problema encontrado foi a alteração de alguns requisitos durante o desenvolvimento, o que levava a mais tarefas e por si mais tempo de desenvolvimento.

Estes imprevistos e dificuldades no desenvolvimento levaram a um atraso no desenvolvimento da aplicação em 1 mês e 1 semana, sendo assim estava previsto o término em março tendo a aplicação sido terminada no início de maio.

Apesar de todas as dificuldades todas as tarefas previstas foram concluídas não existindo assim funcionalidades por implementar.

January 2023	February 2023	March 2023	April 2023	May 2023
9	16	23	30	6
> Sprint 1 18 Feb 13 - Feb 19	Mon, Feb 13 - Sun, Feb 19			
> Sprint 2 19 Feb 20 - Mar 05	Tue, Feb 21 - Sun, Mar 5			
> Sprint 3 20 Mar 06 - Mar 19	Mon, Mar 6 - Thu, Mar 16			
> Sprint 4 19 Mar 20 - Apr 02		Mon, Mar 20 - Thu, Mar 30		
> Sprint 5 20 Apr 03 - Apr 16			Mon, Apr 3 - Thu, Apr 13	
> Sprint 6 20 Apr 17 - Apr 30				Mon, Apr 17 - Thu, Apr 27
> Sprint 7 19 May 01 - May 14				Tue, May 2 - Thu, May 11

Figura 5.1: Organização de tarefas final

5.2 Base de dados

Um dos pontos que sofreu mais alterações durante o desenvolvimento do projeto foi a base de dados, pois com alteração de requisitos e de catálogo de produtos foram necessárias realizar várias alterações até alcançar a versão final.

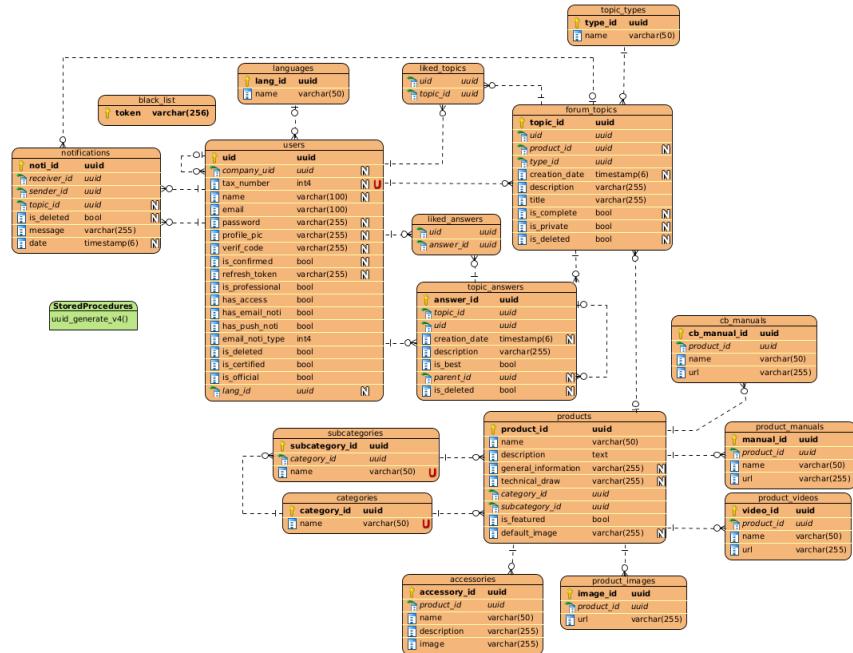


Figura 5.2: Base de dados inicial

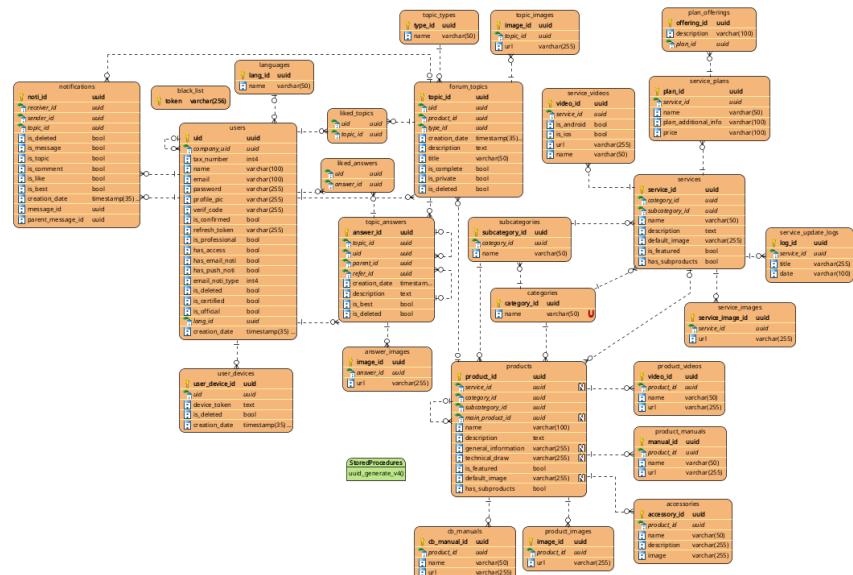


Figura 5.3: Base de dados final

5.3 Opinião do cliente

Após o desenvolvimento do projeto este foi apresentado ao cliente de forma a verificar se este se encontrava de acordo com os requisitos e expectativas do mesmo.

Após a apresentação, o cliente, dispôs de uma opinião positiva, indicando que a aplicação encontra-se de acordo com as expectativas e requisitos, apenas faltando a testagem da mesma em relação a erros e performance, sendo que esta só poderia ser realizada uma vez que a aplicação seja lançada, sendo que seria necessário os utilizadores indicarem os erros.

Bibliografia

- Bracha, Gilad. 2015. The dart programming language.
- Chuvakin, Dr. Anton A., Kevin J. Schmidt & Christopher Philips. 2012. Logging and log management.
- Developers, Android. 2023. Handling android app links. <https://developer.android.com/training/app-links#android-app-links>. [Abril-2023].
- Firebase. 2023. Firebase dynamic links | firebase documentation. <https://firebase.google.com/docs/dynamic-links>. [Abril-2023].
- Halili, Festim & Erenis Ramadani. 2018. Web services: A comparison of soap and rest services. *Modern Applied Science* 12. 175. doi:10.5539/mas.v12n3p175.
- Jin, Brenda, Saurabh Sahni & Amir Shevat. 2018. *Designing web apis*. O'Reilly Media, Inc.
- Mainkar, Prajyot & Salvatore Giordano. 2019. *Google flutter mobile development quick start guide : Get up and running with ios and android mobile app development*. Packt Publishing Ltd.
- Masse, Mark. 2011. *Rest api design rulebook*. O'Reilly Media, Inc.
- Moroney, Laurence. 2017. *The definitive guide to firebase*. Apress. doi:10.1007/978-1-4842-2943-9.
- Snell, James., Doug. Tidwell & Pavel. Kulchenko. 2002. *Programming web services with soap*. O'Reilly & Associates.
- Vanderkam, Dan. 2019. Effective typescript 62 specific ways to improve your typescript.