



Universidade do Minho  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2017/2018

### *Home Massage*

**A77617 André Salgueiro**

**A78565 Hugo Oliveira**

**A77672 Paula Pereira**

**A78085 Rui Calheno**

Junho 2018

Data de Receção	
Responsável	
Avaliação	
Observações	

## ***Home Massage***

**A77617 André Salgueiro**

**A78565 Hugo Oliveira**

**A77672 Paula Pereira**

**A78085 Rui Calheno**

Junho 2018

## Resumo

Este documento diz respeito ao projeto prático elaborado no âmbito da unidade curricular de Laboratórios de Informática IV, lecionada no 3ºano do curso de Engenharia Informática, na Universidade do Minho. O projeto surgiu da necessidade da criação de uma aplicação para uma empresa cujo objetivo é oferecer aos seus clientes um serviço de massagem ao domicílio, disponível a qualquer momento. Neste documento poderá encontrar informação sobre cada uma das etapas de trabalho tais como fundamentação do sistema a desenvolver, especificação do sistema e ainda a sua implementação.

**Área de Aplicação:** Especificação e desenvolvimento de aplicações do mundo real, utilizando metodologias e processos da Engenharia de Software.

**Palavras-Chave:** Engenharia de *Software*, desenvolvimento de *software*, *Microsoft*, *Office*, aplicação *web*, aplicação móvel, *Home Massage*, UML, *use case*, diagrama, base de dados, modelação conceptual, modelação lógica, Modelo MVC, *SQL Server*, *ASP.NET*, *C#*, *Javascript*, HTML.

# Índice

Resumo	i
Índice	ii
Índice de Figuras	vi
Índice de Tabelas	viii
1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objetivos	2
1.4. Estrutura do Relatório	3
2. Fundamentação do Sistema	5
2.1. Justificação, utilidade e viabilidade do sistema	5
2.2. Estabelecimento da identidade do projeto	6
3. Planeamento do projeto	7
3.1. Identificação dos recursos necessários	7
3.2. Maqueta do sistema	8
3.3. Definição de medidas de sucesso	9
3.4. Plano de desenvolvimento	10
4. Análise de Requisitos e Funcionalidades do Sistema	12
4.1. Registrar utilizadores na aplicação	12
4.2. Autenticar utilizadores na aplicação	12
4.3. Terminar sessão	12
4.4. Editar perfil	13
4.5. Requisitar serviços	13
4.6. Consultar localizações	13
4.7. Consultar catálogo de mensagens	14
4.8. Consultar funcionamento	14
4.9. Consultar serviços requisitados	14
4.10. Cancelar serviços	15
4.11. Classificar serviços prestados	15
4.12. Consultar serviços agendados	15

4.13. Obter direções para localizações	16
4.14. Finalizar serviços	16
4.15. Registrar funcionários	17
4.16. Inativar funcionários	17
5. Modelo de domínio	18
5.1. Dicionário de termos	18
6. Diagrama de Use case	20
6.1. Atores do sistema	21
6.2. Especificação dos Use cases	21
6.2.1 Registrar-se na aplicação	22
6.2.2 Autenticar-se na aplicação	23
6.2.3 Terminar sessão	23
6.2.4 Editar perfil	24
6.2.5 Requisitar serviço	25
6.2.6 Cancelar serviço	26
6.2.7 Consultar serviço	26
6.2.8 Classificar serviço	27
6.2.9 Consultar localização	28
6.2.10 Consultar catálogo	29
6.2.11 Consultar funcionamento	29
6.2.12 Obter direções	30
6.2.13 Consultar serviço agendado	31
6.2.14 Finalizar serviço	32
6.2.15 Registrar funcionário	33
6.2.16 Inativar funcionário	34
7. Diagramas de Atividade	35
7.1. Autenticar-se na aplicação	35
7.2. Editar perfil	36
7.3. Requisitar Serviço	37
7.4. Classificar serviço prestado	38
7.5. Finalizar serviço	39
8. Diagramas de Sequência de Subsistemas	40
8.1. Registrar-se na aplicação	40
8.2. Autenticar-se na aplicação	41
8.3. Terminar sessão	41
8.4. Editar perfil	42
8.5. Requisitar serviço	42
8.6. Cancelar serviço	43

8.7. Consultar serviço	43
8.8. Classificar serviço	44
8.9. Consultar localização	44
8.10. Consultar catálogo	45
8.11. Consultar funcionamento	45
8.12. Obter direções	46
8.13. Consultar serviço agendado	46
8.14. Finalizar serviço	47
8.15. Registar funcionário	48
8.16. Inativar funcionário	48
9. Diagrama de classes	49
10. Base de dados	50
10.1. Levantamento e análise de requisitos	50
10.1.1 Requisitos de descrição	50
10.1.2 Requisitos de exploração	51
10.1.3 Requisitos de controlo	51
10.2. Modelação conceptual	52
10.2.1 Identificação e caracterização das entidades	52
10.2.2 Identificação e caracterização dos relacionamentos	52
10.2.3 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	54
10.2.4 Detalhe ou generalização de entidades	55
10.2.5 Diagrama ER	56
10.3. Modelação lógica	56
10.3.1 Construção e validação de modelo de dados lógicos	56
10.3.2 Desenho do modelo lógico	58
10.3.3 Validação do modelo através da normalização	58
11. Idealização da interface	60
11.1. Menu inicial	60
11.2. Menu de registo na aplicação	61
11.3. Menu de autenticação na aplicação	62
11.4. Menu de requisição de serviço	63
11.5. Menu de consulta de serviço agendado	64
12. Implementação	65
12.1. Modelo MVC	65
12.2. Base de dados	66
12.3. Ferramentas utilizadas	66
12.3.1 Sistema de Representação de Mapas	66

12.3.2 Sistema de envio de emails	67
12.3.3 Sistema de envio de SMS	67
13. Produto final	68
13.1. Página Inicial	68
13.2. Registrar-se e autenticar-se na aplicação	69
13.3. Requisitar mensagem	70
13.4. Finalizar serviço	71
13.5. Obter direções	72
13.6. Outras funcionalidades	73
14. Conclusões e Trabalho Futuro	75
Referências Bibliográficas	77
Lista de Siglas e Acrónimos	78
Anexos	79
I. Script de criação da Base de Dados	80
II. Script de povoamento da Base de Dados	82

## Índice de Figuras

Figura 1 - Maqueta do sistema de <i>software</i> .	9
Figura 2 - Calendarização de tarefas.	11
Figura 3 - Diagrama de <i>Gantt</i> .	11
Figura 4 - Modelo de domínio.	18
Figura 5 - Diagrama de Use Case.	20
Figura 6 - Diagrama de atividade de "Autenticar-se na aplicação".	35
Figura 7 - Diagrama de atividade de "Editar perfil".	36
Figura 8 - Diagrama de atividade de "Requisitar serviço".	37
Figura 9 - Diagrama de atividade de "Classificar serviço".	38
Figura 10 - Diagrama de atividade de "Finalizar serviço".	39
Figura 11 - DSS de "Registar-se na aplicação".	40
Figura 12 - DSS de "Autenticar-se na aplicação".	41
Figura 13 - DSS de "Terminar sessão"	41
Figura 14 - DSS de "Editar perfil".	42
Figura 15 - DSS de "Requisitar serviço".	42
Figura 16 - DSS de "Cancelar serviço".	43
Figura 17 - DSS de "Consultar serviço".	43
Figura 18 - DSS de "Classificar serviço".	44
Figura 19 - DSS de "Consultar localização".	44
Figura 20 - DSS de "Consultar catálogo".	45
Figura 21 - DSS de "Consultar funcionamento".	45
Figura 22 - DSS de "Obter direções".	46
Figura 23 - DSS de "Consultar serviço agendado".	46
Figura 24 - DSS de "Finalizar serviço".	47
Figura 25 - DSS de "Registar funcionário".	48
Figura 26 - DSS de "Inativar funcionário".	48
Figura 27 - Diagrama de classes.	49
Figura 28 - Diagrama ER concebido com o software "brModelo".	56
Figura 29 - Modelo lógico concebido com o software "brModelo".	58



Figura 30 - <i>Mockup</i> - Menu inicial.	60
Figura 31 - <i>Mockup</i> - Menu de registo na aplicação.	61
Figura 32 - <i>Mockup</i> - Menu de autenticação na aplicação.	62
Figura 33 - <i>Mockup</i> - Menu de requisição de serviço.	63
Figura 34 - <i>Mockup</i> - Menu de consulta de serviço agendado.	64
Figura 35 - Modelo MVC.	65
Figura 36 - Página inicial.	69
Figura 37 - Registo na aplicação.	69
Figura 38 - Autenticação na aplicação.	70
Figura 39 - Requisitar Massagem.	70
Figura 40 - Serviços agendados de um funcionário.	71
Figura 41 - Serviço selecionado.	71
Figura 42 - Reportar situações durante a finalização de serviço.	72
Figura 43 - Obter direções.	72
Figura 44 - Editar perfil de um cliente.	73
Figura 45 - Classificar um serviço.	73
Figura 46 - Serviços Pendentes/Efetutados de um cliente.	74

## Índice de Tabelas

Tabela 1 - Ficha do projeto <i>Home Massage</i> .	6
Tabela 2 – Dicionário de termos – Modelo de domínio.	19
Tabela 3 – Especificação do <i>use case</i> “Registrar-se na aplicação”.	22
Tabela 4 – Especificação do <i>use case</i> “Autenticar-se na aplicação”.	23
Tabela 5 – Especificação do <i>use case</i> “Terminar sessão”.	23
Tabela 6 – Especificação do <i>use case</i> “Editar perfil”.	24
Tabela 7 – Especificação do <i>use case</i> “Requisitar serviço”.	25
Tabela 8 – Especificação do <i>use case</i> “Cancelar serviço”.	26
Tabela 9 – Especificação do <i>use case</i> “Consultar serviço”.	26
Tabela 10 – Especificação do <i>use case</i> “Classificar serviço”.	27
Tabela 11 – Especificação do <i>use case</i> “Consultar localização”.	28
Tabela 12 – Especificação do <i>use case</i> “Consultar catálogo”.	29
Tabela 13 – Especificação do <i>use case</i> “Consultar funcionamento”.	29
Tabela 14 – Especificação do <i>use case</i> “Obter indicações”.	30
Tabela 15 - Especificação do <i>use case</i> "Consultar serviço agendado".	31
Tabela 16 - Especificação do <i>use case</i> "Finalizar serviço".	32
Tabela 17 - Especificação do <i>use case</i> "Registrar funcionário".	33
Tabela 18 - Especificação do <i>use case</i> "Inativar funcionário".	34
Tabela 19 - Dicionário de dados dos atributos das entidades.	54
Tabela 20 – Informação sobre a tabela “Cliente”.	57
Tabela 21 – Informação sobre a tabela “Funcionario”.	57
Tabela 22 – Informação sobre a tabela “Servico”.	57
Tabela 23 - Informação sobre a tabela "Massagem".	58

# 1. Introdução

## 1.1. Contextualização

A sociedade atual é caracterizada, cada vez mais, pelo seu ritmo acelerado e agitado, induzido principalmente pela constante competição e pressão exercida sobre os cidadãos no seu dia-a-dia. Os adultos, como resultado de demasiadas horas de trabalho, das tarefas relacionadas com a educação dos filhos, entre outros, debatem-se muitas vezes com dores no corpo e stress causado por todos os problemas que enfrentam no dia-a-dia.

Assim, é crucial a existência de meios que ajudem a contrariar este estilo de vida e a reduzir os altos níveis de stress e ansiedade de todos os cidadãos. Surgem, deste modo, inúmeras empresas que promovem estilos de vida mais saudáveis, tais como centros de massagens que, para além de promoverem o relaxamento, combatem o stress acumulado e ajudam no tratamento de diversas lesões e problemas musculares.

A *Zen+* é uma empresa com vários centros de massagens espalhados por várias zonas do país, e cuja sede se encontra em Braga, que, com o intuito de inovar o seu negócio e se destacar da concorrência, realizou um inquérito aos seus clientes habituais no sentido de averiguar qual o interesse destes em que os serviços por eles oferecidos fossem prestados nos seus domicílios. Com este pequeno estudo, a empresa apercebeu-se de que muitos dos seus clientes tinham interesse em usufruir de mais serviços, mas tal não lhes era possível por questões de horários ou de outras situações relacionadas com as suas vidas, tais como, falta de meios para se deslocarem às suas instalações ou até mesmo falta de mobilidade, no caso de pessoas mais idosas.

Perante estes resultados, a empresa decidiu iniciar um pequeno experimento num dos seus centros de massagens mais movimentados, oferecendo aos seus clientes a oportunidade de requisitarem a prestação dos serviços no seu domicílio, sendo que nesses casos, um terapeuta certificado ficaria responsável por se deslocar à localização fornecida pelo cliente, levando consigo todo o material necessário para a terapêutica solicitada. Isto revelou-se um grande sucesso, pelo que a empresa decidiu expandir a ideia aos vários centros.

De forma a facilitar todo o processo de requisição de serviços neste novo projeto ao qual chamou *Home Massage*, a empresa decidiu contratar um conjunto de jovens estudantes de engenharia informática para a criação de uma página *Web* onde os seus clientes podem ter acesso a toda a informação necessária à requisição das várias terapêuticas, sendo que esta

poderá ser igualmente utilizada pelos terapeutas que realizam os serviços para reportarem todas as informações relacionadas com estes.

## **1.2. Apresentação do Caso de Estudo**

Com o principal objetivo viabilizar o seu mais recente projeto *Home Massage*, e tornar a interação com os seus clientes mais fácil, a empresa *Zen+* decidiu lançar uma plataforma *Web* onde poderá dar a conhecer aos visitantes desta o seu novo conceito de massagens ao domicílio. Nesta página, acessível em qualquer browser, os utilizadores deverão encontrar a história por de trás do projeto, os vários serviços oferecidos e os seus custos, as várias localizações onde os serviços podem ser requisitados e uma área restrita a clientes autenticados onde estes podem proceder à requisição dos vários serviços.

No que diz respeito à requisição de serviços, o cliente deve preencher um formulário onde indica todas as informações necessárias, tais como a terapêutica a realizar, local onde o serviço deve ser prestado e quando deve ser prestado. Após a indicação destas informações, o agendamento do serviço é concluído com o envio de um *email* ou SMS ao cliente a confirmar a marcação.

Para facilitar o acompanhamento dos funcionários, a empresa deseja ainda que esta seja concebida de tal forma que cada profissional possua uma área dedicada à gestão de todas as tarefas a desempenhar. Assim, nesta área, este poderá saber, para cada dia de trabalho, as marcações que tem agendadas, onde se deve deslocar, que serviços deve prestar e ainda indicar informações que considere convenientes sobre os serviços prestados. Esta deve ainda possibilitar ao terapeuta proceder à cobrança e recebimento do pagamento do serviço prestado, com a emissão e envio via *email* da fatura e envio da sua referência via SMS ao cliente.

## **1.3. Motivação e Objetivos**

A razão pela qual a empresa *Zen+* decidiu avançar com este projeto prende-se com o facto de esta ter percebido que, em Portugal, ainda não existe nenhuma outra empresa que ofereça este serviço, pelo que esta ideia se destaca num mercado cada vez mais competitivo e onde apenas vingam as empresas capazes de inovar e agradar aos seus clientes.

Para além disso, do ponto de vista dos que irão usufruir da aplicação, concluímos que esta será bastante útil para todos: para os clientes que já estão familiarizados com os serviços oferecidos pela *Zen+*, será uma forma de os requisitar a qualquer momento e poder desfrutar dos mesmos na comodidade das suas casas; para os que ainda não os conhecem será uma oportunidade de ter toda a informação relacionada com estes num só local evitando assim deslocações desnecessárias aos estabelecimentos físicos com o intuito de obterem informações ou realizar marcações.

Já do ponto de vista dos funcionários, esta será uma mais valia no que diz respeito à gestão do seu trabalho. Esta página irá permitir-lhes ter todas as informações relacionadas com os serviços que deverão prestar ao longo do dia, o que por si só será uma mais valia na organização eficiente do seu trabalho, no sentido que irá evitar, por exemplo, atrasos na prestação dos serviços por falta de informações relacionadas com a localização do cliente, ou ainda erros na faturação dos serviços, uma vez que esta será pensada para que não haja lugar para esse tipo de falhas.

Assim sendo, podemos afirmar que com o desenvolvimento desta aplicação *Web*, não só será beneficiada a empresa *Zen+*, que irá facilmente alcançar um número muito maior de clientes, como também os próprios clientes, que irão ter o processo de consulta e requisição de serviços facilitados, e ainda os funcionários, que terão nesta página uma ferramenta de trabalho bastante útil para que este seja prestado com a máxima qualidade.

## **1.4. Estrutura do Relatório**

Este documento tem como principal objetivo transmitir uma ideia clara e concisa de todo o trabalho realizado. Assim sendo, ao longo dos próximos capítulos poderá encontrar informação relacionada quer com a fundamentação, planificação e especificação do sistema de software a desenvolver.

No capítulo 2, poderá encontrar toda a informação relacionada com a justificação do sistema, desde a sua utilidade e viabilidade e ainda o estabelecimento da identidade do sistema.

Já no capítulo 3, iremos abordar aspetos relacionados com o planeamento do projeto, tais como os recursos que serão necessários para a sua implementação, a maqueta do sistema, a definição de medidas de sucesso e a elaboração de um calendário que nos permitirá estabelecer datas limite para a elaboração das tarefas.

De seguida, no 4º capítulo, iremos proceder à análise de requisitos e de funcionalidades da aplicação, pelo que iremos focar-nos não só nos requisitos do utilizador, mas também nos requisitos do sistema.

No capítulo 5, iremos apresentar o modelo de domínio que caracteriza o nosso sistema, assim com um dicionário de termos que facilitará a sua compreensão.

Já nos capítulos 6, 7, 8 e 9, iremos apresentar vários diagramas UML, tais como diagrama de use cases e respetivos diagramas de sequência de subsistemas, diagramas de atividade e diagrama de classes, cujo objetivo é definir e especificar claramente todo o nosso sistema.

Posteriormente, no capítulo 10, iremos abordar todas as informações relacionados com a base de dados que pretendemos implementar, pelo que poderá encontrar nesse capítulo informações relacionadas quer com a sua modelação conceptual quer com a sua modelação lógica.

No capítulo 11, poderá encontrar algumas figuras relacionadas com a idealização da interface gráfica da aplicação.

O capítulo 12 é onde se encontra toda a informação relativa com a implementação do sistema propriamente dito. Lá irá encontrar conteúdo relacionado com a arquitetura do sistema de *software*, implementação da base de dados e ferramentas e API's utilizadas.

De seguida, no capítulo 13, iremos apresentar o produto final, focando-nos essencialmente na interface desenvolvida.

Por fim, iremos refletir sobre o trabalho realizado, discutir os pontos fortes e fracos do mesmo e as nossas perspetivas de trabalho futuro.

Nas últimas páginas deste documento poderá ainda encontrar uma série de referências bibliográficas e uma lista de siglas e acrónimos que serão úteis à compreensão da informação deste documento.

## **2. Fundamentação do Sistema**

### **2.1. Justificação, utilidade e viabilidade do sistema**

A vontade de avançar com o projeto *Home Massage* surge após a *Zen+* verificar que no estrangeiro o conceito de massagens no domicílio já era concreto e utilizado por diversas empresas, mas que, no entanto, em território português isso não se verificava. Desta forma, seriam pioneiros do conceito em Portugal, o que, certamente, traria uma vantagem relativamente às outras empresas, visto que o cliente poderia, na conveniência e comodidade da sua casa, requisitar e usufruir dos serviços da *Zen+* sem necessidade de se deslocar um dos seus estabelecimentos.

De modo a materializar esta vontade, era necessário saber qual a posição dos seus clientes em relação ao novo conceito. Numa fase inicial, a empresa começou por averiguar junto dos seus clientes qual o interesse destes no conceito, não podendo ter recebido um feedback mais positivo. De seguida, a empresa começou por oferecer o serviço a um grupo restrito de clientes com o objetivo de avaliar a viabilidade do projeto, e mais uma vez concluiu que este tinha grande potencial, sendo que todos os clientes ficavam extremamente satisfeitos com o serviço prestado.

Apesar de estarem completamente convencidos do sucesso do projeto, a empresa percebeu de imediato que este estava totalmente dependente do número de pessoas que tivessem conhecimento da sua existência e, por isso, não hesitou na elaboração da plataforma *Web*, uma vez que esta tornaria a interação com o cliente mais acessível. Para além disso, esta irá permitir que o cliente consulte todos os serviços disponíveis e ainda viabiliza a requisição de qualquer um deles de acordo com as preferências de quem o irá usufruir.

Tendo isto, podemos afirmar que os principais motivos que levarão uma pessoa a usufruir do *software* a desenvolver é a comodidade e conveniência que este oferece. Este irá permitir que qualquer utilizador possa, a qualquer momento, requisitar os seus serviços preferidos e, depois de agendados, usufruir destes no conforto das duas casas eliminando assim toda a ansiedade provocada pela necessidade de se deslocar a um dos estabelecimentos da empresa.

## 2.2. Estabelecimento da identidade do projeto

Ficha de Projeto

<b>Nome</b>	<i>Home Massage</i>
<b>Área</b>	Saúde e Lazer
<b>Idioma</b>	Português
<b>Faixa Etária</b>	18+ anos
<b>Descrição</b>	<p>Prestação de serviço ao domicílio que funcionará através de uma página <i>Web</i> onde um utilizador pode consultar os vários serviços de massagem disponibilizados pela empresa <i>Zen<sup>+</sup></i>.</p> <p>Nesta página, um cliente, após realizar o seu registo, pode requisitar vários serviços. Uma vez agendados, estes serão prestados por um funcionário qualificado na localização fornecida pelo cliente.</p> <p>O funcionário possui acesso a uma interface diferente, onde, para além de obter informações sobre os serviços que tem agendados, pode também registar os detalhes dos serviços já prestados e processar a sua faturação.</p>
<b>Cliente</b>	<i>Zen<sup>+</sup></i>
<b>Criadores</b>	André Salgueiro, Hugo Oliveira, Paula Pereira, Rui Calheno

Tabela 1 - Ficha do projeto *Home Massage*.



## 3. Planeamento do projeto

### 3.1. Identificação dos recursos necessários

Na estruturação e planeamento inicial do projeto é necessário tomar consciência dos recursos que serão cruciais para o desenvolvimento das aplicações e que irão sustentar o bom funcionamento do produto final. Assim sendo, apresentamos uma lista dos recursos que identificamos como imprescindíveis para o desenvolvimento do *software*:

- Sistema de validação de códigos de postais: impedirá que os clientes apresentem localizações inválidas ou fora do alcance de prestação de serviços providenciados no projeto *Home Massage*.
- Sistema de representação de mapas: permitirá ao funcionário visualizar o trajeto a realizar desde o seu local inicial até à localização indicada pelos clientes.
- Sistema de localização: permitirá o posicionamento do funcionário no mapa, de modo a que este obtenha direções até à casa dos clientes.
- Sistema de envio de *emails*: permitirá o envio automático dos *emails* de confirmação de agendamento e de faturas relativas à realização do serviço.
- Existência prévia de dados: é fundamental existirem dados relativos aos diversos serviços prestados pela empresa de forma a apresenta-los de forma clara e perceptível aos clientes.
- Camada da apresentação: os clientes e os funcionários terão acesso direto a esta camada. É, portanto, fundamental que o seu design seja intuitivo e de fácil movimentação, oferecendo ao cliente uma experiência agradável e ao funcionário uma ferramenta de trabalho auxiliar.
- Camada de negócio: responsável por processar a informação vinda da camada de apresentação, realizando uma gestão do conteúdo a ser colocado na base de dados e a ser apresentado na interface de utilizador.
- Camada de dados: o armazenamento de dados relativos aos clientes, aos serviços da empresa e às marcações é crucial para o funcionamento da aplicação. Desta forma, o Sistema de Base de Dados permitirá uma resposta rápida aos pedidos de informação e acesso simultâneo e concorrente por parte dos diferentes clientes da empresa.
- Equipa de desenvolvimento: é necessário assegurar que a equipa seja capaz de fazer um bom uso das mais variadas ferramentas, tais como *Microsoft .NET*, *Microsoft SQL*

*Server*, *Microsoft Visual Studio*, entre outras. Para além disso, os vários elementos devem ser capazes de cumprir as diversas tarefas nos prazos estabelecidos e comunicar com o resto da equipa.

## 3.2. Maqueta do sistema

De modo a facilitar a compreensão acerca do funcionamento da aplicação *Web*, foi construída uma maqueta do sistema (Figura 1) que permite ilustrar todos os componentes da aplicação e a forma como estes estão interligados.

De uma forma geral, os clientes, têm acesso ao catálogo de mensagens e a um diverso conjunto de outras informações. De seguida, aqueles que se encontram autenticados podem proceder à requisição de serviços, preenchendo um formulário onde indicam os detalhes dessa mesma requisição.

Esses detalhes são então processados na camada de negócio. Verifica-se a disponibilidade dos terapeutas, a veracidade da localização fornecida e, caso a marcação seja validada, é enviado um *email* para posterior confirmação da mesma.

Finalmente, a marcação é inserida na Base de Dados, de modo a manter o registo dos diversos serviços realizados e por realizar.

Os funcionários, por sua vez, possuem uma interface de utilizador diferente da dos clientes. Nesta é-lhes apresentado as marcações às quais foram atribuídos e, aquando da seleção de uma destas, surge um mapa de navegação que orienta o funcionário até à localização indicada pelo cliente. De salientar que é a camada de negócio que trata de todo o processamento de navegação, sendo apenas apresentado ao funcionário um mapa e direções.

No fim do serviço, o funcionário dá como concluído o mesmo e os registos da Base de Dados são atualizados de modo a guardar informação relevante para a *Zen+*.

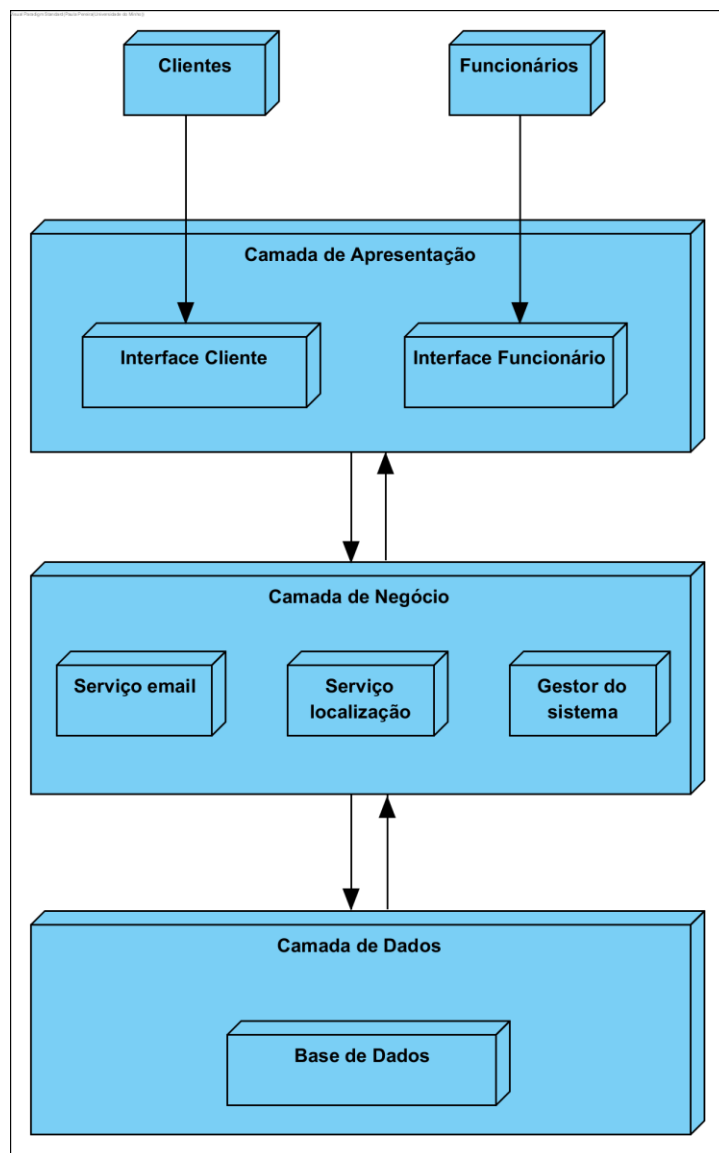


Figura 1 - Maqueta do sistema de *software*.

### 3.3. Definição de medidas de sucesso

Tendo em conta que o serviço *Home Massage* é um serviço inovador em Portugal, podemos considerar que o seu potencial é enorme.

Um dos principais motivos pelo qual a *Zen+* sente que a sua aplicação irá levar a um aumento claro do seu negócio centra-se no facto de cada vez mais, em Portugal, existir um número cada vez maior de jovens e adultos com problemas posturais que necessitam de cuidados terapêuticos.

Aliado a este crescente grupo de pessoas com problemas de saúde encontram-se os desportistas que, quer para tratamento de lesões quer para relaxamento muscular, recorrem frequentemente à realização de massagens, e os idosos, cujos músculos se vão tornando mais débeis com o correr dos anos.

Desta forma, verifica-se que o público alvo da aplicação *Home Massage* é bastante alargado, sendo que os serviços alcançam as áreas da saúde, desporto e lazer.

Com base nas razões acima referidas e apostando numa aplicação *Web* intuitiva e simples de usar, a *Zen+* acredita no sucesso da sua nova aplicação, esperando cerca de 300 clientes assíduos na zona de Braga, gerando com isso cerca 52.000 euros por mês.

Finalmente, a empresa espera conseguir lucro suficiente para expandir o seu negócio a todo Portugal Continental, podendo assim aumentar o número de clientes e consequentemente as suas receitas.

### **3.4. Plano de desenvolvimento**

O desenvolvimento deste projeto foi dividido em três fases principais: fundamentação, especificação e implementação.

A primeira fase durou cerca de 15 dias e consistiu essencialmente na contextualização e fundamentação do projeto.

Na segunda fase, cuja duração foi de 41 dias, concentramo-nos na análise dos requisitos e na especificação do *software* a desenvolver, pelo que utilizamos a linguagem UML na construção de vários diagramas que nos permitiram planejar de forma clara o comportamento do *software* e nos auxiliaram na sua implementação. Também nesta fase procedemos à modelação de uma base de dados que nos permite guardar todos os dados relacionados com os clientes e com os serviços prestados.

A implementação propriamente dita foi realizada ao longo da terceira fase, sendo que, ao longo de 52 dias, nos concentramos em utilizar toda a informação obtida na segunda fase de forma a obter um produto final consistente e de acordo com os requisitos estabelecidos na primeira fase. Assim, procedemos à implementação física da base de dados e desenvolvemos todos os aspetos relacionados com o *backend* e *frontend* do sistema de *software*. Finalmente, procedemos à realização vários testes de forma a averiguar que o sistema funciona de acordo com o esperado.

Como se pode perceber, este processo exigiu uma grande dedicação e coordenação por parte de todos os elementos da equipa de desenvolvimento, pelo que se tornou evidente a necessidade de criar um calendário que nos permitiu estabelecer metas e objetivos de trabalho, assim como atribuir tarefas aos vários membros da equipa.

Para a construção deste calendário, foi utilizada a ferramenta de gestão de projetos *Microsoft Project Professional 2016*, tendo sido igualmente obtido um diagrama de *Gantt*.


























































ID		Modo de Tarefa	Nome da Tarefa	Duração	Início	Conclusão	Predecessoras	Nomes de Recursos
1			<b>Fase I - Fundamentação</b>	<b>15 dias</b>	<b>Dom 18/02/18</b>	<b>Seg 05/03/18</b>		
2			Identificação e caracterização da aplicação	4 dias	Dom 18/02/18	Qui 22/02/18		André;Hugo;Paula;Rui
3			Fundamentação	10 dias	Qui 22/02/18	Dom 04/03/18	2	André;Hugo;Paula;Rui
4			Projeção do calendário	3 dias	Ter 27/02/18	Sex 02/03/18	2	Paula
5			Reunião - Fundamentação	0 dias	Sex 02/03/18	Sex 02/03/18		André;Hugo;Paula;Rui
6			Realização do relatório - Parte I	10 dias	Qui 22/02/18	Dom 04/03/18		André;Hugo;Paula;Rui
7			Entrega do relatório - Parte I	0 dias	Seg 05/03/18	Seg 05/03/18	2;3;4;5;6	Rui
8			<b>Fase II - Especificação</b>	<b>41 dias</b>	<b>Ter 06/03/18</b>	<b>Seg 16/04/18</b>	<b>1</b>	
9			Análise de requisitos	10 dias	Ter 06/03/18	Sex 16/03/18		André;Hugo;Paula;Rui
10			Reunião - Análise de requisitos	0 dias	Sex 16/03/18	Sex 16/03/18	9	André;Hugo;Paula;Rui
11			Modelo de domínio	5 dias	Sex 16/03/18	Qua 21/03/18	10	André;Hugo;Paula;Rui
12			Diagrama de use case e especificações	2 dias	Qua 21/03/18	Sex 23/03/18	11	André;Hugo
13			Diagrama de classe	10 dias	Qua 21/03/18	Sáb 31/03/18	11	André;Hugo;Paula;Rui
14			Diagramas de sequência	17 dias	Qua 21/03/18	Sáb 07/04/18	11	Paula;Rui
15			Diagramas de atividade	4 dias	Sáb 07/04/18	Qua 11/04/18	11	André;Hugo
16			Reunião - Diagramas	0 dias	Seg 09/04/18	Seg 09/04/18	10	André;Hugo;Paula;Rui
17			Base de Dados	20 dias	Sex 16/03/18	Qui 05/04/18	10	Paula;Rui
18			Idealização da interface	8 dias	Dom 08/04/18	Seg 16/04/18		Hugo;André
19			Realização do relatório - Parte II	30 dias	Sex 16/03/18	Dom 15/04/18	10	André;Hugo;Paula;Rui
20			Entrega do relatório - Parte II	0 dias	Seg 16/04/18	Seg 16/04/18	9;10;12;14;15;16;17;19	Rui
21			<b>Fase III - Implementação</b>	<b>55 dias</b>	<b>Ter 17/04/18</b>	<b>Seg 11/06/18</b>	<b>2</b>	
22			Base de Dados - Modelação Física	2 dias	Ter 17/04/18	Qui 19/04/18	17	Paula;Rui
23			Backend	25 dias	Seg 14/05/18	Sex 08/06/18	22	Hugo;Rui
24			Backend - Testes	0 dias	Sex 08/06/18	Sex 08/06/18	23	Hugo;Rui
25			Frontend	19 dias	Dom 20/05/18	Sex 08/06/18	22	André;Hugo;Rui;Paula
26			Testes finais	1 dia	Sáb 09/06/18	Dom 10/06/18	23;25	André;Hugo;Rui;Paula
27			Realização do relatório - Parte III	55 dias	Ter 17/04/18	Seg 11/06/18		André;Hugo;Rui;Paula
28			Entrega do relatório - Parte III	0 dias	Seg 11/06/18	Seg 11/06/18	22;23;24;25;26;27	Rui

Figura 2 - Calendarização de tarefas.

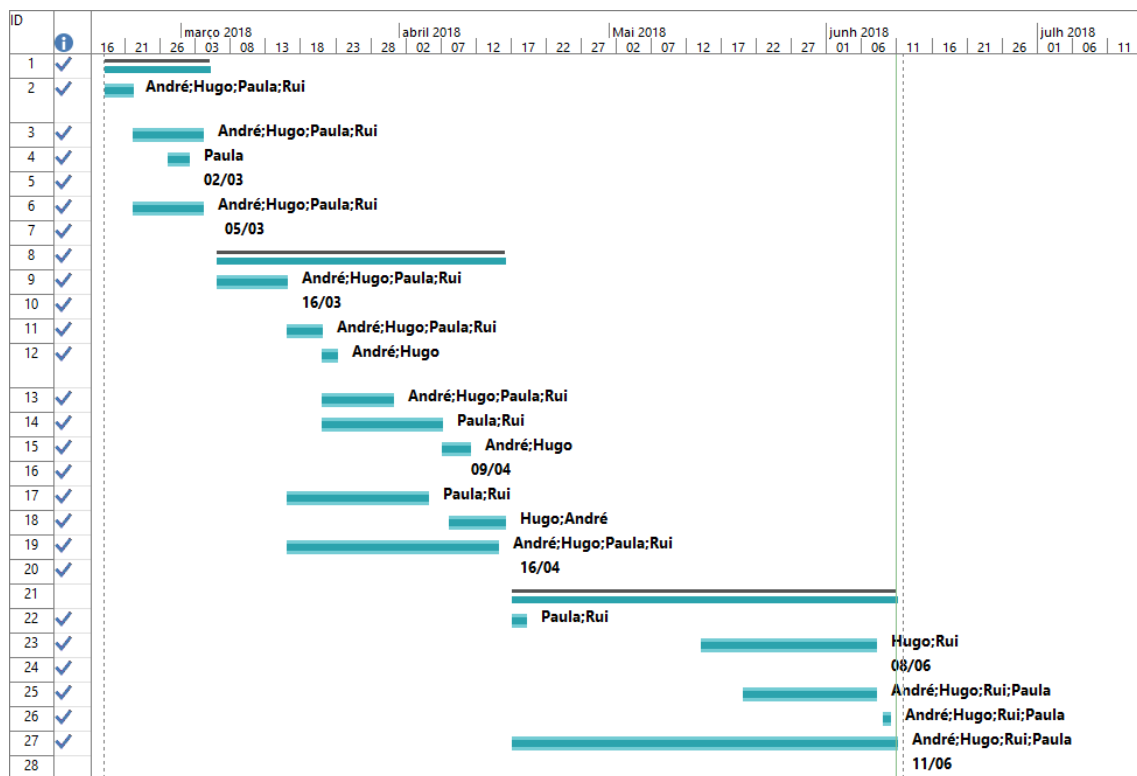


Figura 3 - Diagrama de Gantt.

## 4. Análise de Requisitos e Funcionalidades do Sistema

### 4.1. Registrar utilizadores na aplicação

#### Requisitos de utilizador

- |  |
|--|
| 1. Deverá ser possível o registo de novos utilizadores na aplicação. |
|--|

#### Requisitos de sistema

- |  |
|--|
| <p>1.1. O sistema deverá solicitar as informações necessárias ao registo: nome, <i>username</i>, <i>password</i>, <i>email</i>, número de contribuinte, contacto telefónico e data de nascimento.</p> <p>1.2. O sistema deverá impedir o registo de vários utilizadores com o mesmo <i>username</i>.</p> <p>1.3. O sistema deverá impedir o registo de utilizadores com menos de 18 anos.</p> <p>1.4. O sistema deverá guardar os dados relativos ao registo na base de dados.</p> |
|--|

### 4.2. Autenticar utilizadores na aplicação

#### Requisitos de utilizador

- |   |
|---|
| 2. Deverá ser possível a autenticação de utilizadores na aplicação. |
|---|

#### Requisitos de sistema

- |  |
|--|
| <p>2.1. O sistema deverá solicitar as credenciais necessárias à autenticação na aplicação: <i>username</i> e <i>password</i>.</p> <p>2.2. O sistema deverá verificar e validar as credenciais.</p> |
|--|

### 4.3. Terminar sessão

#### Requisitos de utilizador

- |   |
|---|
| 3. O cliente autenticado deverá poder terminar a sua sessão na aplicação. |
|---|

#### Requisitos de sistema

3.1. O sistema deverá permitir ao cliente terminar a sua sessão na aplicação.

### 4.4. Editar perfil

#### Requisitos de utilizador

4. O cliente autenticado deverá poder editar o seu perfil.

#### Requisitos de sistema

4.1. O sistema deverá impedir um cliente de aceder a outras informações que não as suas.

4.2. O sistema deverá permitir que um cliente autenticado altere as suas informações pessoais, tais como *email*, *password*, nome, data de nascimento e contacto telefónico.

4.3. As novas informações devem ser guardadas na base de dados.

### 4.5. Requisitar serviços

#### Requisitos de utilizador

5. O cliente autenticado deverá poder requisitar serviços através do preenchimento de um formulário.

#### Requisitos de sistema

5.1. O sistema deverá permitir que um cliente autenticado proceda à requisição de serviços.

5.2. O sistema deverá verificar o correto preenchimento dos vários campos do formulário.

5.3. O sistema deverá recorrer a um sistema de validação de códigos postais para validar a localização fornecida.

5.4. O sistema deverá verificar se existem terapeutas disponíveis para prestar o serviço na data e horas indicadas pelo cliente.

5.5. O sistema deverá atribuir, caso esteja disponível, um terapeuta ao serviço.

5.6. O sistema deverá enviar automaticamente um *email* ao cliente a confirmar o agendamento.

5.7. O sistema deverá guardar a informação do serviço requisitado na base de dados.

### 4.6. Consultar localizações

#### Requisitos de utilizador

6. O utilizador deverá poder verificar se são prestados serviços num local com um determinado código postal.

#### **Requisitos de sistema**

**6.1.** O sistema deverá recorrer a um sistema de validação de códigos postais para verificar se nessa localização é possível a prestação de serviços.

### **4.7. Consultar catálogo de mensagens**

#### **Requisitos de utilizador**

**7.** O utilizador deverá poder consultar um catálogo com as informações de todos os serviços prestados.

#### **Requisitos de sistema**

**7.1.** O sistema deverá providenciar todas as informações relevantes sobre os vários serviços, tais como as mensagens disponíveis, em que consistem, preço do serviço, entre outras.

**7.2.** O sistema deverá recorrer aos dados armazenados na base de dados para obter as informações sobre os vários serviços.

### **4.8. Consultar funcionamento**

#### **Requisitos de utilizador**

**8.** O utilizador deverá ter acesso a informação explicativa do funcionamento do serviço.

#### **Requisitos de sistema**

**8.1.** O sistema deverá providenciar informações ao utilizador relacionadas não só com a requisição dos serviços, mas também com a prestação dos mesmos.

### **4.9. Consultar serviços requisitados**

#### **Requisitos de utilizador**

**9.** Um cliente autenticado deverá poder consultar as informações relacionadas com todos os serviços que já requisitou.



#### **Requisitos de sistema**

- 9.1.** O sistema deverá ser capaz de apresentar informações sobre todos os serviços já requisitados pelo cliente.
- 9.2.** O sistema deverá ser capaz de filtrar os serviços de acordo com vários critérios de seleção.
- 9.3.** O sistema deverá recorrer aos dados armazenados na base de dados para obter as informações sobre os vários serviços.

### **4.10. Cancelar serviços**

#### **Requisitos de utilizador**

- 10.** Um cliente autenticado deverá poder cancelar a requisição de um determinado serviço.

#### **Requisitos de sistema**

- 10.1.** O sistema deverá permitir ao cliente autenticado cancelar a requisição de um determinado serviço.
- 10.2.** O sistema deverá registar as informações relacionadas com o cancelamento do serviço na base de dados.

### **4.11. Classificar serviços prestados**

#### **Requisitos de utilizador**

- 11.** O cliente autenticado deverá poder classificar um serviço que lhe foi prestado.

#### **Requisitos de sistema**

- 11.1.** O sistema deverá permitir que um cliente classifique um serviço que lhe foi prestado.
- 11.2.** O sistema deverá verificar se o serviço que o cliente pretende classificar já foi dado como finalizado.
- 11.3.** O sistema deverá guardar a informação relacionada com a classificação do serviço na base de dados.

### **4.12. Consultar serviços agendados**

#### **Requisitos de utilizador**

- 12.** Um funcionário deverá ter acesso a uma lista de serviços agendados.

#### **Requisitos de sistema**

- 12.1.** O sistema deverá ser capaz de gerar uma lista dos serviços agendados para cada funcionário.
- 12.2.** O sistema deverá providenciar sobre cada serviço agendado todas as informações necessárias à sua prestação: cliente, serviço requisitado, localização, entre outros.

### **4.13. Obter direções para localizações**

#### **Requisitos de utilizador**

- 13.** Um funcionário deverá poder obter direções para uma qualquer localização indicada por um cliente para a prestação de um serviço.

#### **Requisitos de sistema**

- 13.1.** O sistema deverá recorrer a um sistema de representação de mapas e de localização para providenciar o trajeto a realizar até à localização indicada pelo cliente.

### **4.14. Finalizar serviços**

#### **Requisitos de utilizador**

- 14.** O funcionário deverá ser capaz de finalizar um serviço através do processamento do seu pagamento e reportação de ocorrências relevantes.

#### **Requisitos de sistema**

- 14.1.** O sistema deverá permitir que um funcionário assinale um serviço como terminado.
- 14.2.** O sistema deverá permitir ao funcionário processar o pagamento do serviço.
- 14.3.** O sistema deverá permitir ao funcionário reportar situações sobre a prestação do serviço.
- 14.4.** O sistema deverá utilizar um serviço de *emails* para proceder ao envio da fatura ao cliente.
- 14.5.** O sistema deverá utilizar um serviço de SMS para proceder ao envio da referência da fatura ao cliente.
- 14.6.** O sistema deverá guardar as informações na base de dados.

## 4.15. Registrar funcionários

### Requisitos de utilizador

15. O administrador deverá poder registar novos funcionários.

### Requisitos de sistema

15.1. O sistema deverá permitir o registo de novos funcionários.

15.2. O sistema deverá solicitar o *username*, *email* e nome do funcionário e verificar se estes são válidos.

15.3. O sistema deverá gerar uma *password* para o novo funcionário.

15.4. O sistema deverá utilizar um serviço de *emails* para enviar ao funcionário as suas credenciais de acesso à aplicação.

15.5. O sistema deverá guardar as informações relativas ao novo funcionário na base de dados.

## 4.16. Inativar funcionários

### Requisitos de utilizador

16. O administrador deverá poder inativar funcionários.

### Requisitos de sistema

16.1. O sistema deverá permitir inativar funcionários.

16.2. O sistema deverá atualizar, na base de dados, o estado do funcionário de ativo para inativo.

## 5. Modelo de domínio

Tendo em conta os requisitos estabelecidos, procedemos à elaboração do modelo de domínio apresentado na Figura 4. Este tem como principal objetivo descrever o panorama em que se insere o nosso projeto através das palavras chaves que o definem e as relações entre as entidades que estas representam.

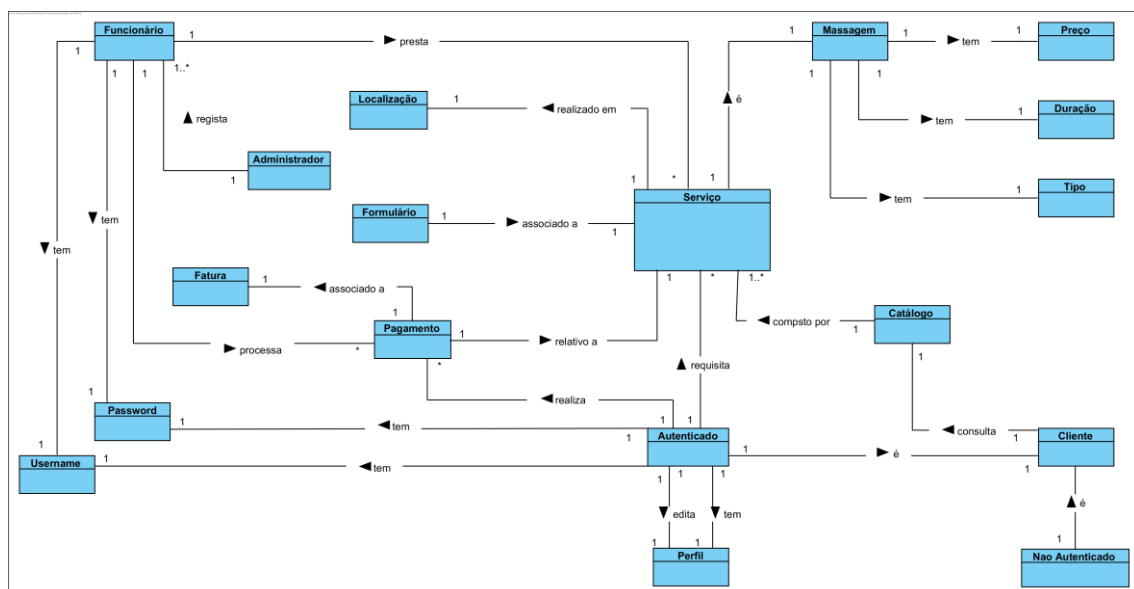


Figura 4 - Modelo de domínio.

### 5.1. Dicionário de termos

Termo	Definição
<b>Utilizador</b>	Representa qualquer utilizador da aplicação.
<b>Autenticado</b>	Representa um utilizador registado e autenticado na aplicação. Também referenciado como “Cliente”.
<b>Cliente</b>	Representa um utilizador registado e autenticado na aplicação.
<b>Não autenticado</b>	Representa um utilizador não autenticado na aplicação.
<b>Funcionário</b>	Representa um funcionário da empresa. Entidade responsável pela prestação dos serviços.

<b>Administrador</b>	Representa o administrador da aplicação. Entidade responsável pelo registo de funcionários.
<b>Perfil</b>	Conjunto de informações pessoais de cada cliente.
<b>Username</b>	Identificador que o utilizador usa para se autenticar na aplicação.
<b>Password</b>	Conjunto de caracteres utilizado pelo utilizador para se autenticar na aplicação.
<b>Catálogo</b>	Conjunto de todos os serviços que a <i>Zen+</i> oferece.
<b>Serviço</b>	Serviço prestado pela <i>Zen+</i> . Diz respeito a uma massagem.
<b>Massagem</b>	Massagem disponível. Consiste na compressão das partes musculares do corpo e das articulações para fins terapêuticos.
<b>Tipo</b>	Tipo de uma massagem.
<b>Duração</b>	Duração de uma massagem.
<b>Preço</b>	Preço associado a uma massagem.
<b>Pagamento</b>	Efeito de pagar o serviço prestado.
<b>Fatura</b>	Documento onde se encontrar todos os dados relacionados com o pagamento de um serviço prestado.
<b>Localização</b>	Local onde o serviço é prestado.
<b>Formulário</b>	Conjunto de campos que o cliente deve preencher aquando da requisição de um determinado serviço.

Tabela 2 – Dicionário de termos – Modelo de domínio.

## 6. Diagrama de Use case

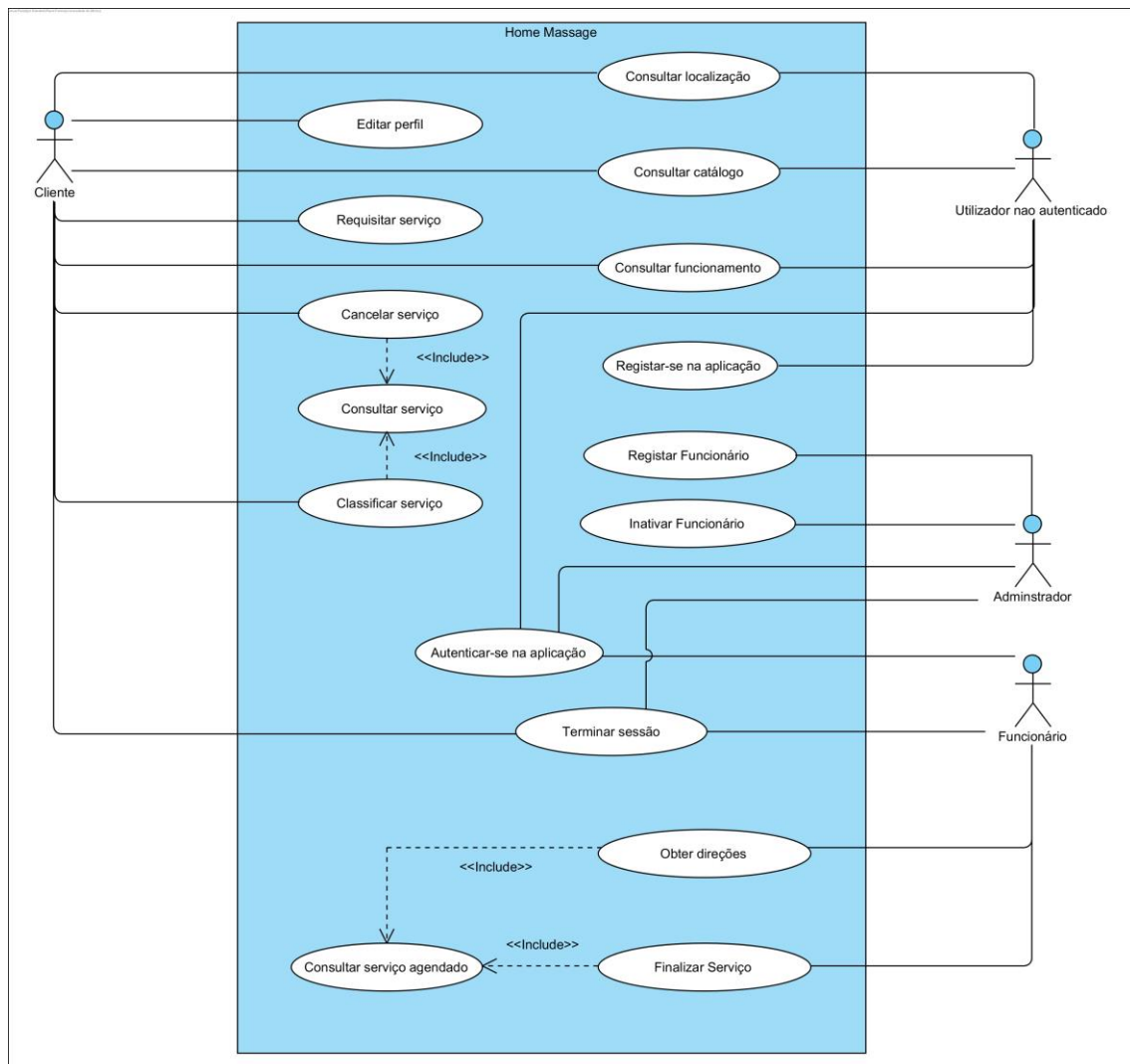


Figura 5 - Diagrama de Use Case.

## 6.1. Atores do sistema

No que diz respeito ao sistema de *software* a desenvolver, identificamos três tipos de atores: o administrador, os funcionários e os utilizadores.

No que diz respeito ao **Administrador**, este é a entidade responsável por registar os funcionários da empresa para que estes possam autenticar-se na aplicação e aceder aos dados relacionados com a prestação dos serviços.

Por sua vez, o **Funcionário** corresponde a um funcionário cujo o registo na aplicação foi realizado pelo administrador e que, estando autenticado na aplicação, pode consultar um conjunto de informações relacionadas com a prestação de serviços.

Por fim, em relação aos utilizadores distinguem-se dois casos:

- **Utilizador não autenticado** – qualquer utilizador da aplicação que não estando autenticado na mesma, apenas pode efetuar ações de consulta de informações, estando assim impedido de requisitar e classificar serviços. Este tem ainda a opção de efetuar um registo ou autenticar-se na aplicação.
- **Cliente** – corresponde a um utilizador registado e autenticado na aplicação, pelo que, para além de poder realizar todas as ações de consulta, pode proceder à requisição/cancelamento/classificação de serviços, e ainda editar as suas informações pessoais.

## 6.2. Especificação dos Use cases

Neste capítulo iremos apresentar a especificação dos vários *use cases* apresentados na Figura 5. Para tal, pela sua simplicidade, iremos utilizar a notação tabular.

### 6.2.1 Registar-se na aplicação

<b>Use Case</b>	<b>Registar-se na aplicação</b>		
<b>Descrição</b>	Utilizador regista-se na aplicação.		
<b>Pré-Condição</b>	Utilizador não autenticado no sistema.		
<b>Pós-Condição</b>	Utilizador registado com sucesso.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende registar-se.	
	2		Solicita dados necessários ao registo.
	3	Fornece dados solicitados.	
	4		Verifica dados.
	5		Valida registo.
	6		Informa que o registo foi realizado com sucesso.
<b>Exceção 1</b> <b>[Dados inválidos]</b> <b>(passo 4)</b>	5.1		Indica que os dados fornecidos são inválidos.

Tabela 3 – Especificação do *use case* “Registar-se na aplicação”.

O *use case* “Registar-se na aplicação” diz respeito à situação em que um qualquer utilizador se regista na aplicação.

Para isso, são lhe solicitados vários dados, tais como nome, *username*, *password*, *email*, numero de contribuinte, contacto telefónico e data de nascimento. O registo só é efetuado se o *email* indicado for válido, se o *username* estiver disponível e se a data de nascimento corresponder a um indivíduo maior de idade.



### 6.2.2 Autenticar-se na aplicação

<b>Use Case</b>	<b>Autenticar-se na aplicação.</b>		
<b>Descrição</b>	Utilizador autentica-se na aplicação.		
<b>Pré-Condição</b>	Utilizador registado e não autenticado.		
<b>Pós-Condição</b>	Utilizador autenticado.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende autenticar-se.	
	2		Solicita credenciais.
	3	Insere credenciais.	
	4		Verifica credenciais.
	5		Valida autenticação.
<b>Exceção 1</b> <b>[Credenciais inválidas]</b> <b>(passo 5)</b>	5.1		Indica que as credenciais são inválidas

Tabela 4 – Especificação do *use case* “Autenticar-se na aplicação”.

O *use case* “Autenticar-se na aplicação” diz respeito à situação em que um qualquer utilizador com registo na aplicação autentica-se na mesma.

Para tal, o utilizador fornece o seu *username* e a sua *password* e, apenas no caso destas serem validadas pelo sistema, o utilizador fica autenticado na aplicação.

### 6.2.3 Terminar sessão

<b>Use Case</b>	<b>Terminar Sessão.</b>		
<b>Descrição</b>	Utilizador termina a sua sessão atual.		
<b>Pré-Condição</b>	Utilizador encontra-se autenticado.		
<b>Pós-Condição</b>	Utilizador não se encontra autenticado.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que quer terminar sessão.	
	2		Termina sessão do utilizador.

Tabela 5 – Especificação do *use case* “Terminar sessão”.

O *use case* “Terminar sessão” diz respeito à situação em que um qualquer utilizador autenticado no sistema termina a sua sessão na aplicação.

Para tal, é apenas necessário que o utilizador indique que pretende terminar sessão.

### 6.2.4 Editar perfil

<b>Use Case</b>	<b>Editar perfil.</b>		
<b>Descrição</b>	Cliente edita as suas informações pessoais.		
<b>Pré-Condição</b>	Cliente autenticado no sistema.		
<b>Pós-Condição</b>	Perfil atualizado com sucesso.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende editar os dados pessoais do seu perfil.	
	2		Apresenta perfil do cliente.
	3	Fornece novos dados.	
	4		Verifica dados.
	5		Processa pedido.
	6		Informa que o perfil foi atualizado com sucesso.
<b>Exceção 1 [Dados inválidos] (Passo 4)</b>	2.1		Informa que a atualização do perfil não é possível

Tabela 6 – Especificação do *use case* “Editar perfil”.

O *use case* “Editar perfil” diz respeito à situação em que um cliente edita as suas informações pessoais.

Para tal, o cliente deve aceder ao seu perfil e atualizar os vários campos passíveis a alterações, sendo eles: *email*, *password*, nome, data de nascimento, contacto telefónico, número do cartão de crédito e número de contribuinte.

### 6.2.5 Requisitar serviço

<b>Use Case</b>	<b>Requisitar serviço.</b>		
<b>Descrição</b>	Cliente requisita serviço.		
<b>Pré-Condição</b>	Cliente autenticado na aplicação.		
<b>Pós-Condição</b>	Serviço requisitado com sucesso.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende requisitar um serviço.	
	2		Apresenta formulário de requisição de serviço.
	3	Preenche formulário.	
	4		Processa pedido.
	5		Regista requisição no sistema.
	6		Envia <i>email</i> para o cliente a confirmar o agendamento.
	7		Informa que o pedido foi requisitado com sucesso.
<b>Exceção 1 [Serviço não pode ser realizado na localização indicada pelo cliente] (Passo 4)</b>	4.1		Informa que o serviço não pode ser agendado devido a localização inválida.
<b>Exceção 2 [Não existem terapeutas disponíveis] (Passo 4)</b>	4.1		Informa que o serviço não pode ser agendado devido à não disponibilidade de terapeutas.

Tabela 7 – Especificação do use case “Requisitar serviço”.

O use case “Requisitar serviço” diz respeito à situação em que um cliente requisita um dos vários serviços disponibilizados na aplicação.

Para isso, o cliente preenche um formulário em que fornece todos os dados necessários: mensagem desejada, localização onde o serviço deve ser prestado, data da prestação do serviço, cartão de crédito a utilizar entre outros. Se a localização indicada for válida e houver terapeutas disponíveis para prestar o serviço, então este é registado e confirmado ao cliente via *email*.

## 6.2.6 Cancelar serviço

<b>Use Case</b>	<b>Cancelar serviço.</b>		
<b>Descrição</b>	Cliente cancela um serviço.		
<b>Pré-Condição</b>	Cliente autenticado no sistema.		
<b>Pós-Condição</b>	Requisição cancelada com sucesso.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	<<include>> Consultar serviço.	
	2	Indica que pretende cancelar o serviço.	
	3		Processa pedido.
	4		Informa que a requisição foi cancelada.

Tabela 8 – Especificação do use case “Cancelar serviço”.

O use case “Cancelar serviço” diz respeito à situação em que um cliente cancela um serviço anteriormente requisitado.

Para tao, basta que o cliente indique ao sistema qual o serviço que pretende cancelar, sendo o cancelamento posteriormente registado no sistema.

## 6.2.7 Consultar serviço

<b>Use Case</b>	<b>Consultar serviço.</b>		
<b>Descrição</b>	Cliente consulta serviço.		
<b>Pré-Condição</b>	Cliente autenticado no sistema.		
<b>Pós-Condição</b>	Cliente consulta informações relativas a um serviço.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende consultar um serviço.	
	2		Apresenta lista de serviços realizados e pendentes.
	3	Seleciona um serviço.	
			Apresenta informações do serviço.

Tabela 9 – Especificação do use case “Consultar serviço”.

O use case “Consultar serviço” diz respeito à situação em que um cliente, autenticado no sistema, consulta toda a informação relativa a um serviço a partir de uma lista de serviços por si já realizados ou requisitados.

Para tal, é necessário apenas que o cliente indique o serviço sobre o qual pretende obter informações.

### 6.2.8 Classificar serviço

Use Case	Classificar Serviço.		
Descrição	Cliente classifica um serviço após a sua realização.		
Pré-Condição	Cliente autenticado e serviço realizado.		
Pós-Condição	Classificação registada no sistema.		
Comportamento Normal		<b>Ator</b>	<b>Sistema</b>
	1	<<include>> Consultar serviço.	
	2	Indica que pretende classificar serviço.	
	3		Verifica se o serviço já foi finalizado.
	4		Apresenta classificações.
	5	Escolhe classificação.	
	6		Regista classificação.
Exceção 1 [Serviço não finalizado] (Passo 3)	3.1		Informa que o serviço não pode ser classificado por ainda não ter sido finalizado.

Tabela 10 – Especificação do *use case* “Classificar serviço”.

O *use case* “Classificar serviço” diz respeito à situação em que um cliente, autenticado no sistema, classifica um serviço.

Para tal, basta que, aquando da consulta da informação desse serviço, indique que pretende proceder à sua classificação. A classificação só será possível se o serviço já tiver sido dado com finalizado, e em caso positivo, esta é registada no sistema.

### 6.2.9 Consultar localização

<b>Use Case</b>	<b>Consultar localização.</b>		
<b>Descrição</b>	Utilizador verifica se, para uma determinada localização, há prestação de serviços.		
<b>Pré-Condição</b>			
<b>Pós-Condição</b>	Utilizador obtém informação sobre a prestação de serviços numa determinada localização.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1.	Indica que pretende consultar localização.	
	2.	Indica código de postal.	
	3.		Verifica código de postal.
	4.		Apresenta informação relativa à prestação de serviços na localização fornecida.
<b>Exceção 1 [Código postal inválido] (Passo 3)</b>	3.1		Informa que o código postal indicado não é válido.

Tabela 11 – Especificação do *use case* “Consultar localização”.

O *use case* “Consultar localização” diz respeito à situação em que um cliente ou utilizador não autenticado pretende averiguar se é possível a prestação de um serviço numa determinada localização.

Para isso, basta que indique o código postal correspondente à localização onde pretende que o serviço seja prestado e o sistema verifica a sua validade ou não.

### 6.2.10 Consultar catálogo

<b>Use Case</b>	<b>Consultar catálogo.</b>		
<b>Descrição</b>	Utilizador consulta catálogo de serviços oferecidos.		
<b>Pré-Condição</b>			
<b>Pós-Condição</b>	Utilizador obtém informação sobre os serviços prestados.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	<b>1</b>	Indica que pretende consultar o catálogo de mensagens.	
	<b>2</b>		Apresenta catálogo de mensagens.

Tabela 12 – Especificação do *use case* “Consultar catálogo”.

O *use case* “Consultar catálogo” diz respeito à situação em que um cliente ou utilizador não autenticado pretende visualizar o catálogo de mensagens disponíveis

Para tal, basta que indique que pretende visualizar o catálogo e o sistema apresenta um conjunto de informações relativas aos vários serviços, tais como: quais as mensagens disponíveis, em que consistem, preço do serviço, entre outras.

### 6.2.11 Consultar funcionamento

<b>Use Case</b>	<b>Consultar funcionamento.</b>		
<b>Descrição</b>	Utilizador consulta informações sobre a requisição e prestação de serviços.		
<b>Pré-Condição</b>			
<b>Pós-Condição</b>	Utilizador obtém as informações pretendidas.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	<b>1</b>	Indica que pretende obter informações sobre o funcionamento do serviço.	
	<b>2</b>		Apresenta informações pretendidas.

Tabela 13 – Especificação do *use case* “Consultar funcionamento”.

O *use case* “Consultar funcionamento” diz respeito à situação em que um cliente ou utilizador não autenticado pretende obter informações relacionadas com a requisição e prestação de serviços.

### 6.2.12 Obter direções

<b>Use Case</b>	<b>Obter direções.</b>		
<b>Descrição</b>	Funcionário obtém direções para localização fornecida pelo cliente.		
<b>Pré-Condição</b>	Funcionário autenticado no sistema.		
<b>Pós-Condição</b>	Funcionário obtém direções para a localização pretendida.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	<<include>> Consultar serviço agendado.	
	2	Indica que pretende obter direções.	
	3		Processa pedido.
	4		Apresenta trajetória até ao local pretendido.
<b>Exceção 1 [Impossível apresentar trajetória] (Passo 4)</b>	4.1		Indica que é impossível apresentar trajetória pretendida.

Tabela 14 – Especificação do *use case* “Obter indicações”.

O *use case* “Obter direções” diz respeito à situação em que um funcionário, autenticado no sistema, solicita direções para uma determinada localização.

Para tal, o funcionário deve, aquando da consulta do serviço agendado, indicar que pretende obter a trajetória a seguir. Esta é então apresentada pelo sistema, a não ser que, por algum motivo externo, tal não seja possível.



### 6.2.13 Consultar serviço agendado

<b>Use Case</b>	<b>Consultar serviço agendado.</b>		
<b>Descrição</b>	Funcionário consulta informações sobre um serviço agendado.		
<b>Pré-Condição</b>	Funcionário autenticado no sistema.		
<b>Pós-Condição</b>	Funcionário obtém informações sobre um serviço agendado.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende consultar serviço agendado.	
	2		Apresenta lista de serviços agendados.
	3	Indica serviço desejado.	
	4		Processa pedido.
	5		Apresenta informações do serviço indicado.

Tabela 15 - Especificação do *use case* "Consultar serviço agendado".

O *use case* "Consultar serviço agendado" diz respeito à situação em que um funcionário, autenticado no sistema, consulta toda a informação relativa a um serviço a partir de uma lista de serviços agendados para si,

Para tal, é necessário apenas que o funcionário indique o serviço sobre o qual pretende obter informações.

### 6.2.14 Finalizar serviço

<b>Use Case</b>	<b>Finalizar serviço.</b>		
<b>Descrição</b>	Funcionário finaliza um serviço, processando o seu pagamento e indicando informações relevantes sobre esse mesmo serviço.		
<b>Pré-Condição</b>	Funcionário autenticado no sistema.		
<b>Pós-Condição</b>	Serviço é dado como concluído e o seu pagamento é processado.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	<<include>> Consultar serviço agendado.	
	2	Indica que pretende finalizar serviço.	
	3		Finaliza Serviço.
	4		Gera fatura relativa ao serviço.
	5		Envia fatura via <i>email</i> para o cliente.
	6		Envia referência da fatura via SMS para o cliente.
	7		Pergunta se pretende reportar alguma situação.
	8	Diz que sim.	
	9	Reporta situação.	
	10		Regista informações no sistema.
	11		Confirma finalização do serviço.
<b>Alternativa 1 [Não pretende reportar nenhuma situação] (Passo 8)</b>	8.1	Diz que não.	
	8.2		Confirma finalização do serviço.

Tabela 16 - Especificação do *use case* "Finalizar serviço".

O *use case* "Finalizar serviço" diz respeito à situação em que um funcionário autenticado sinaliza um serviço como finalizados.

Para isso, o funcionário consulta as informações do serviço e indica que pretende finalizá-lo. Isto faz com que o sistema gere a fatura relativa ao serviço prestado e a envie, por email, ao cliente, assim como a sua respetiva referência, por SMS. É ainda dada a opção ao

funcionário de reportar alguma outra situação que este considere relevante, sendo que essa informação é posteriormente registada no sistema.

### 6.2.15 Registar funcionário

Use Case	Registar Funcionário.		
Descrição	Administrador regista um funcionário no sistema.		
Pré-Condição	Funcionário não registado no sistema.		
Pós-Condição	Funcionário registado com sucesso no sistema.		
Comportamento Normal		Ator	Sistema
	1	Indica que pretende registar funcionário.	
	2		Apresenta campos para realização do registo.
	3	Indica <i>username</i> e <i>email</i> do funcionário.	
	4		Valida <i>username</i> e <i>email</i> .
	5		Gera <i>password</i> aleatória para o funcionário.
	6		Regista funcionário no Sistema.
	7		Envia <i>email</i> ao funcionário com as suas credenciais.
	8		Confirma registo.
Exceção 1 [ <i>Username Inválido</i> ] (Passo 4)	2.1		Indica que <i>username</i> já se encontra em uso.
Exceção 2 [ <i>Email Inválido</i> ] (Passo 4)	2.1		Indica que o <i>email</i> é inválido.

Tabela 17 - Especificação do use case "Registar funcionário".

O use case "Registar funcionário" diz respeito à situação em que o administrador pretende registar um novo funcionário no sistema.

Para tal, o administrador fornece o *username*, *email* e nome correspondentes ao novo funcionário e, se estes forem válidos, ou seja, o *username* não se encontra em uso e o *email* é válido, o sistema gera um *password*, envia ao novo funcionário todas as credenciais para o *email* indicado, e regista-o no sistema.

### 6.2.16 Inativar funcionário

<b>Use Case</b>	<b>Inativar Funcionário.</b>		
<b>Descrição</b>	Administrador inativa um funcionário no sistema.		
<b>Pré-Condição</b>	Funcionário registrado no sistema.		
<b>Pós-Condição</b>	Funcionário inativado com sucesso no sistema.		
<b>Comportamento Normal</b>		<b>Ator</b>	<b>Sistema</b>
	1	Indica que pretende inativar funcionário.	
	2		Apresenta lista de funcionários ativos.
	3	Escolhe funcionário.	
	4		Inativa funcionário.
	5		Indica que funcionário foi se encontra inativo.

Tabela 18 - Especificação do *use case* "Inativar funcionário".

O *use case* "Inativar funcionário" diz respeito à situação em que o administrador dá um funcionário com inativo na empresa. Isto pode dever-se ao despedimento ou aposento do funcionário da *Zen+*.

Para tal, o administrador indica o funcionário que pretende dar como inativo a partir de uma lista com informação de todos os funcionários ativos, sendo essa informação posteriormente registada do sistema.

## 7. Diagramas de Atividade

Nesta secção iremos apresentar os diagramas de atividade relativos às funcionalidades que consideramos mais importantes na nossa aplicação e que foram referidas em Análise de Requisitos e Funcionalidades do Sistema.

### 7.1. Autenticar-se na aplicação

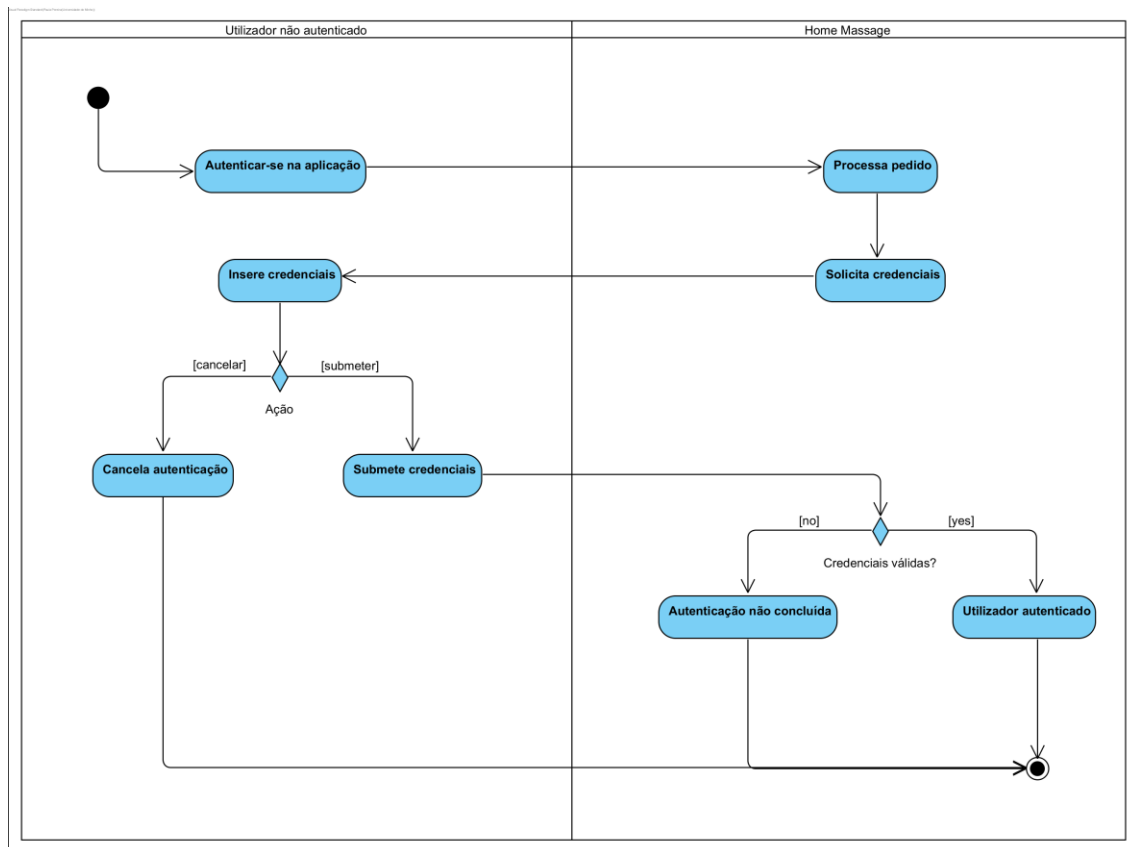


Figura 6 - Diagrama de atividade de "Autenticar-se na aplicação".

Este diagrama de atividade diz respeito à funcionalidade Autenticar utilizadores na aplicação.

## 7.2. Editar perfil

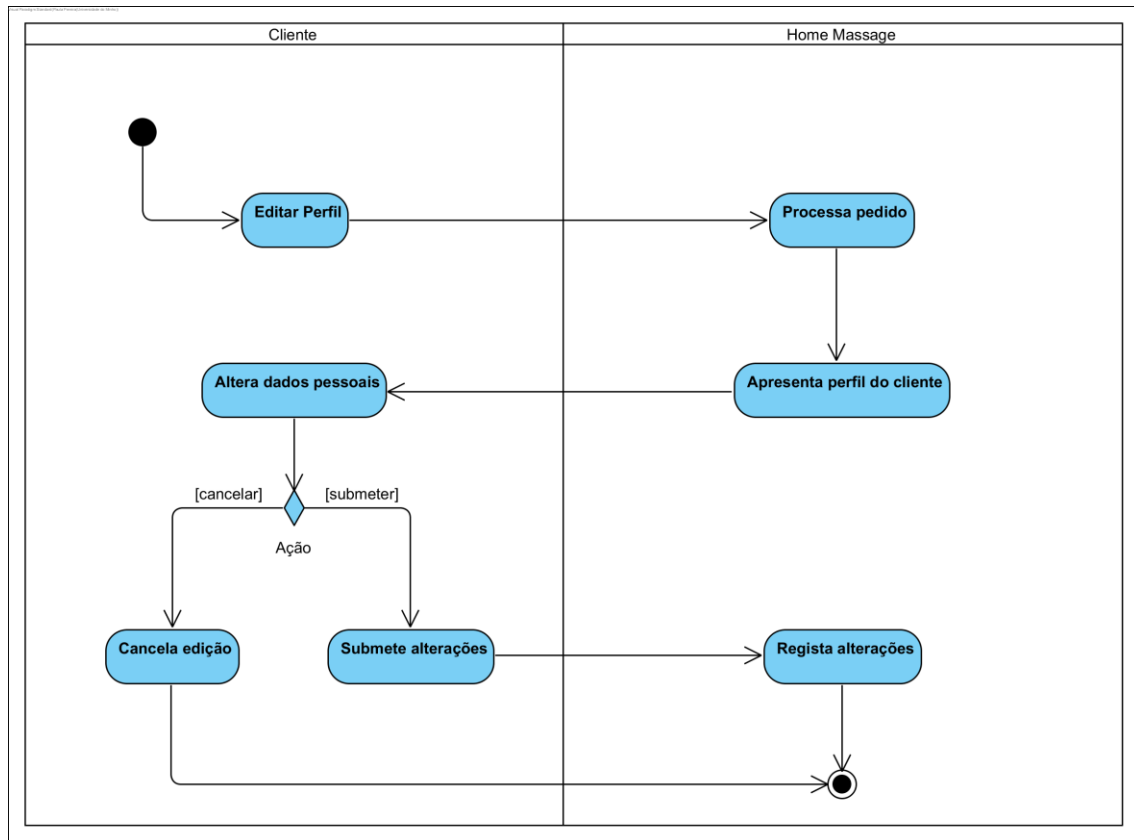


Figura 7 - Diagrama de atividade de "Editar perfil".

Este diagrama de atividade diz respeito à funcionalidade Editar perfil.

### 7.3. Requisitar Serviço

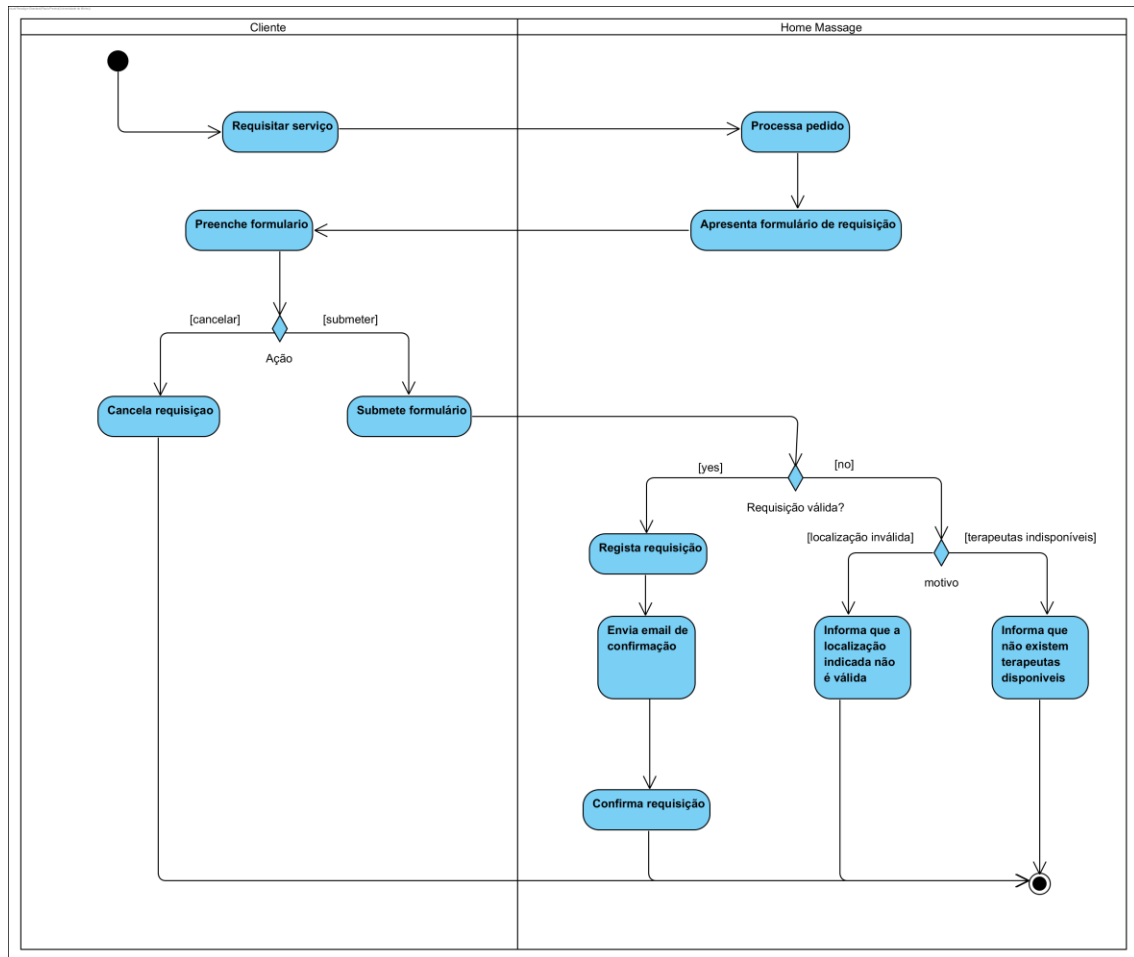


Figura 8 . Diagrama de atividade de "Requisitar serviço".

Este diagrama de atividade diz respeito à funcionalidade Requisitar serviço.

## 7.4. Classificar serviço prestado

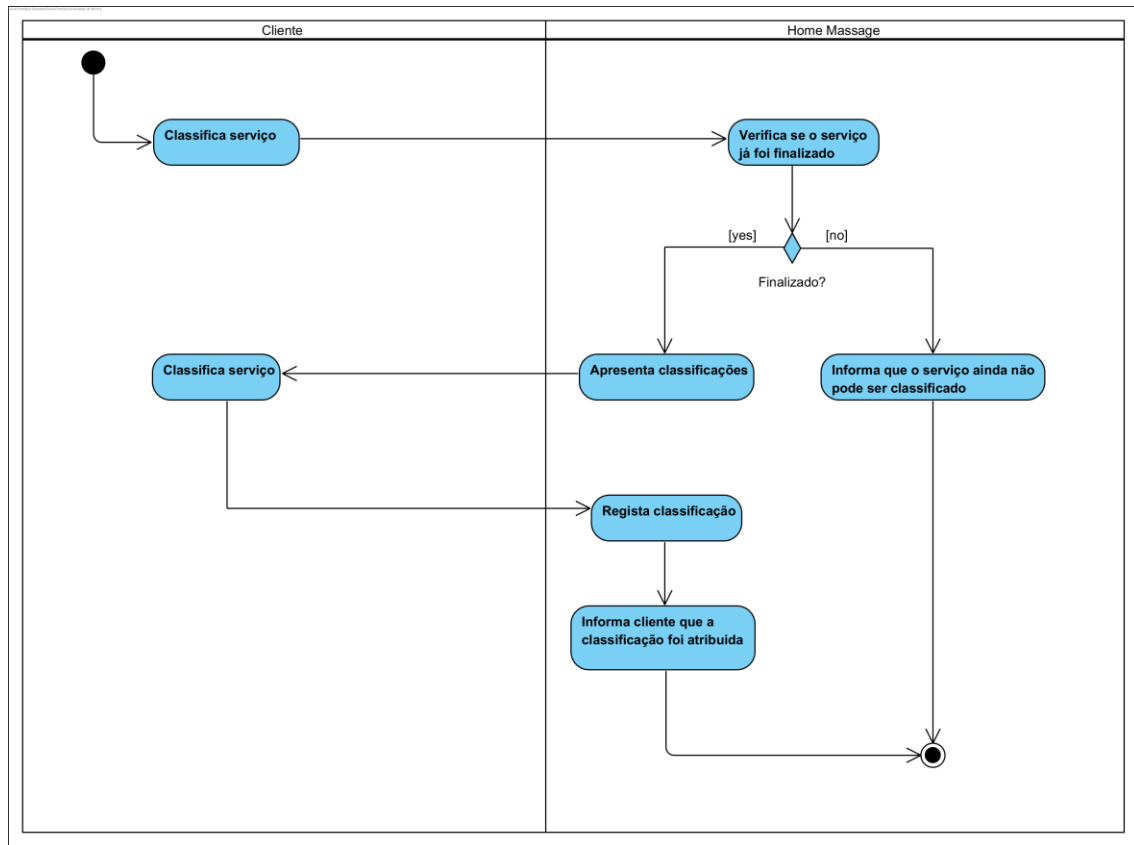


Figura 9 - Diagrama de atividade de "Classificar serviço".

Este diagrama de atividade diz respeito à funcionalidade Classificar serviços prestado.



## 7.5. Finalizar serviço

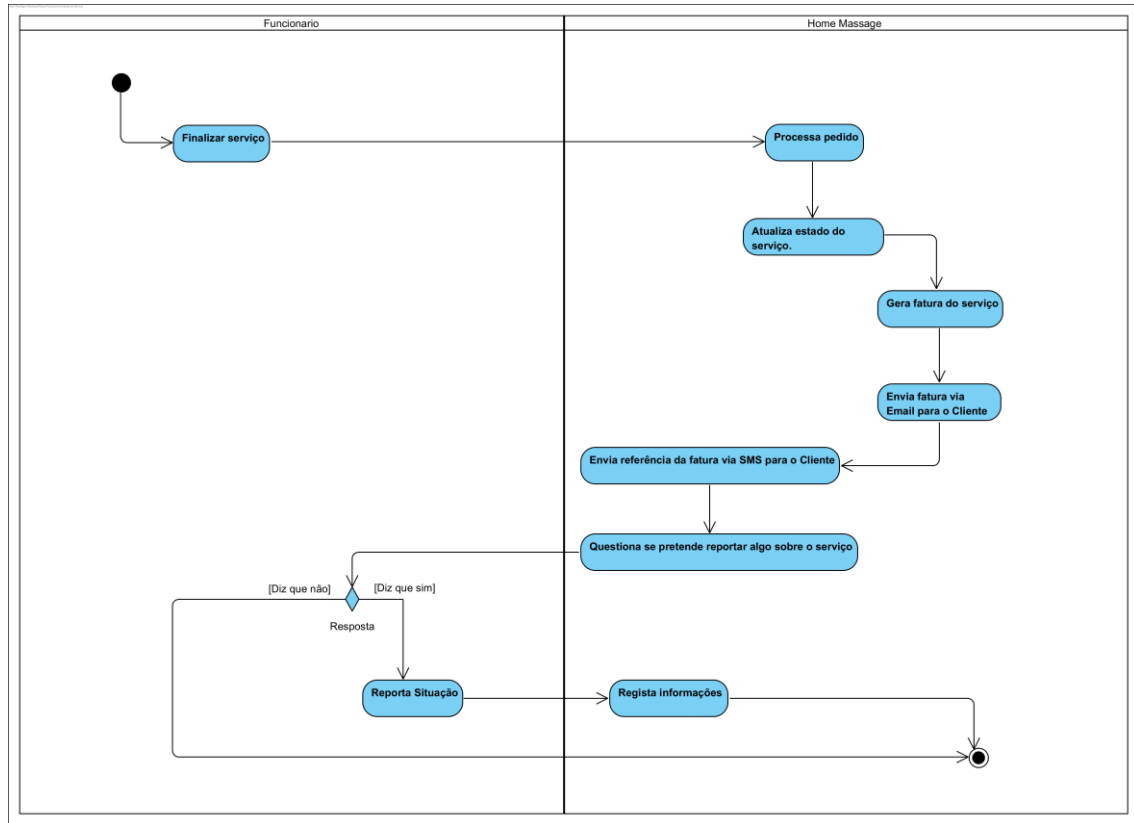


Figura 10 - Diagrama de atividade de "Finalizar serviço".

Este diagrama de atividade diz respeito à funcionalidade Finalizar serviço.



## 8.2. Autenticar-se na aplicação

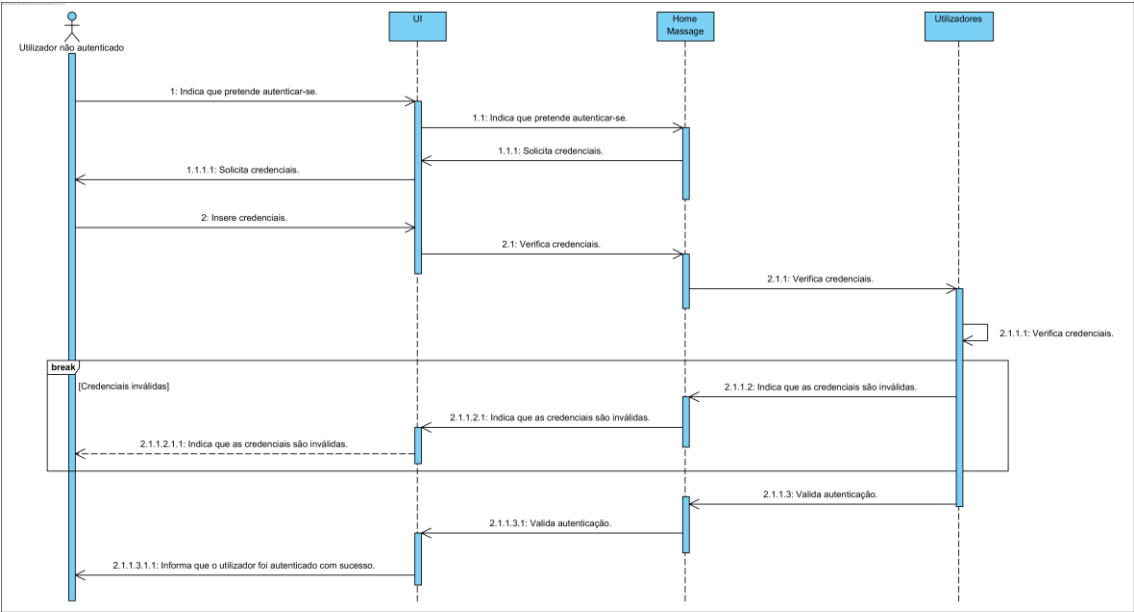


Figura 12 - DSS de "Autenticar-se na aplicação".

## 8.3. Terminar sessão

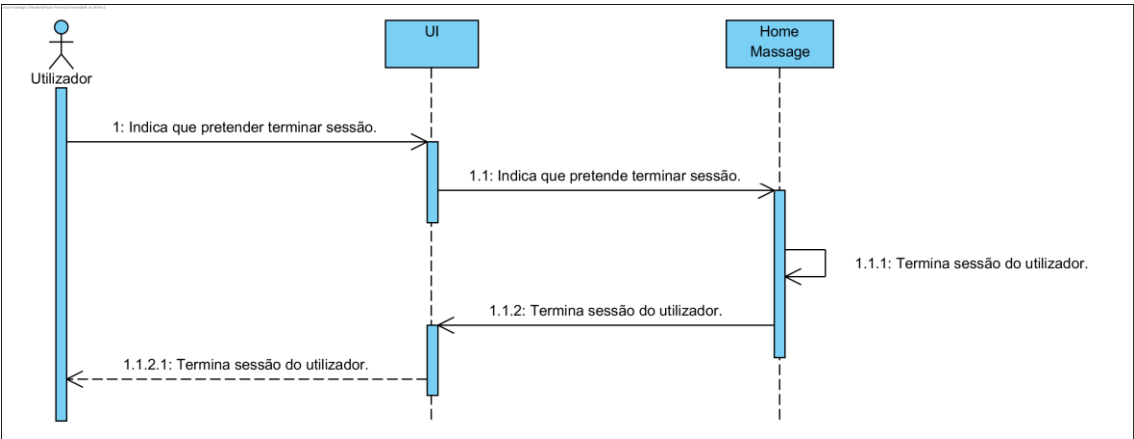


Figura 13 - DSS de "Terminar sessão"

## 8.4. Editar perfil

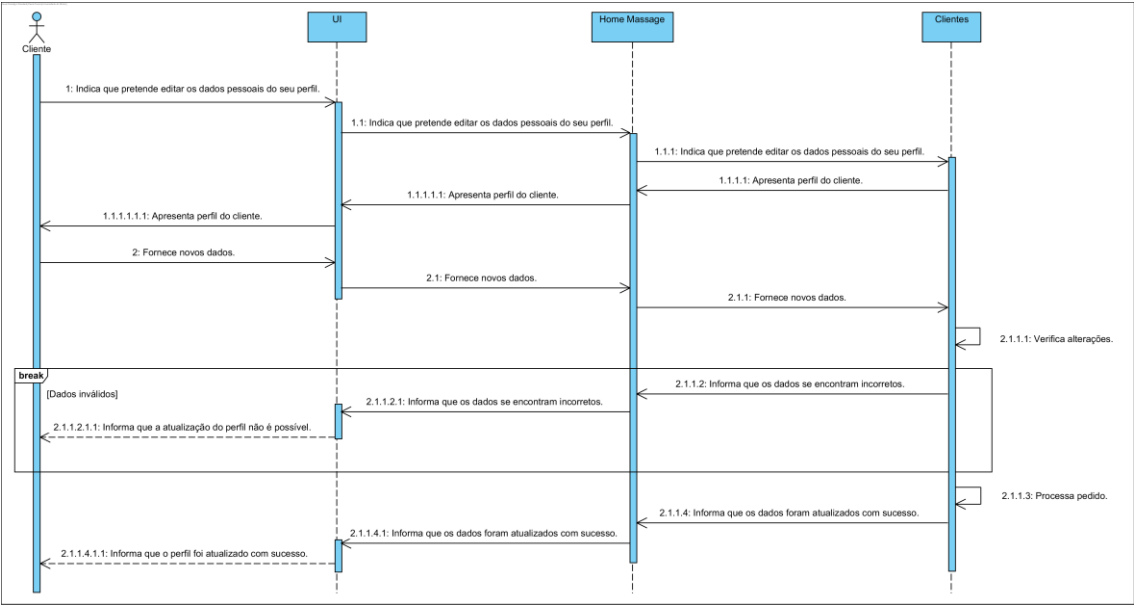


Figura 14 - DSS de "Editar perfil".

## 8.5. Requisitar serviço

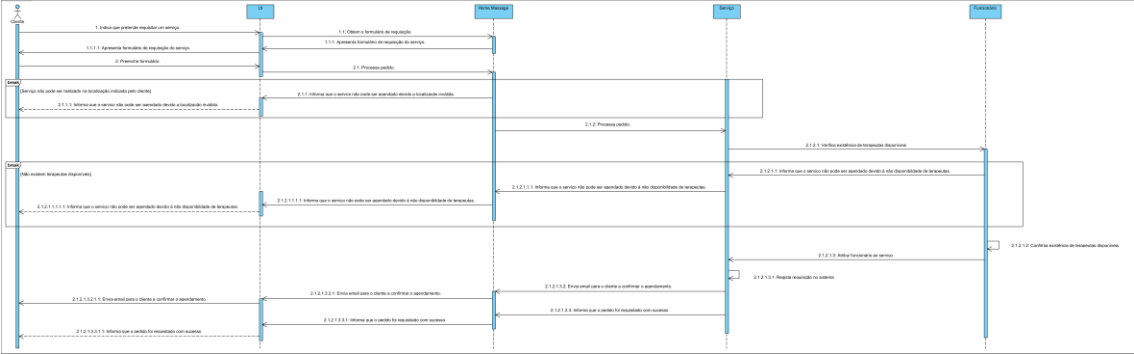


Figura 15 - DSS de "Requisitar serviço".

## 8.6. Cancelar serviço

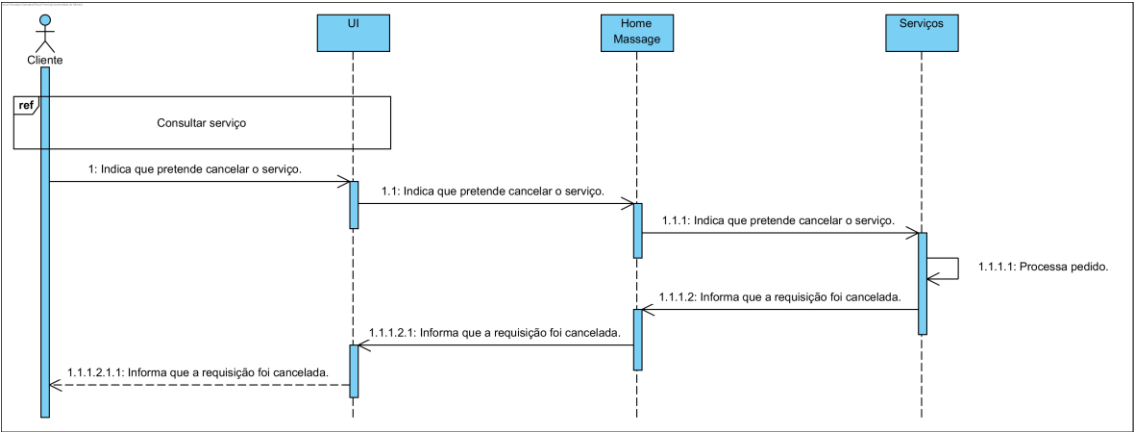


Figura 16 - DSS de "Cancelar serviço".

## 8.7. Consultar serviço

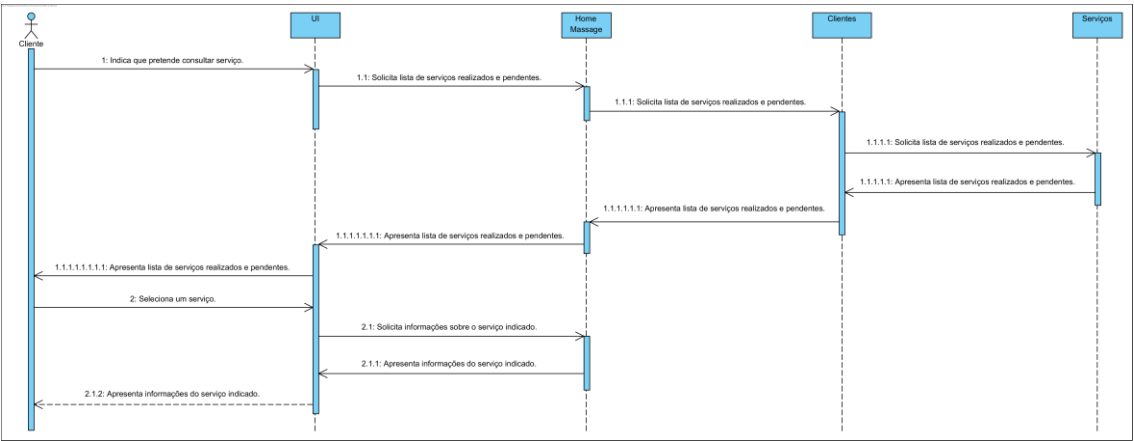


Figura 17 - DSS de "Consultar serviço".

## 8.8. Classificar serviço

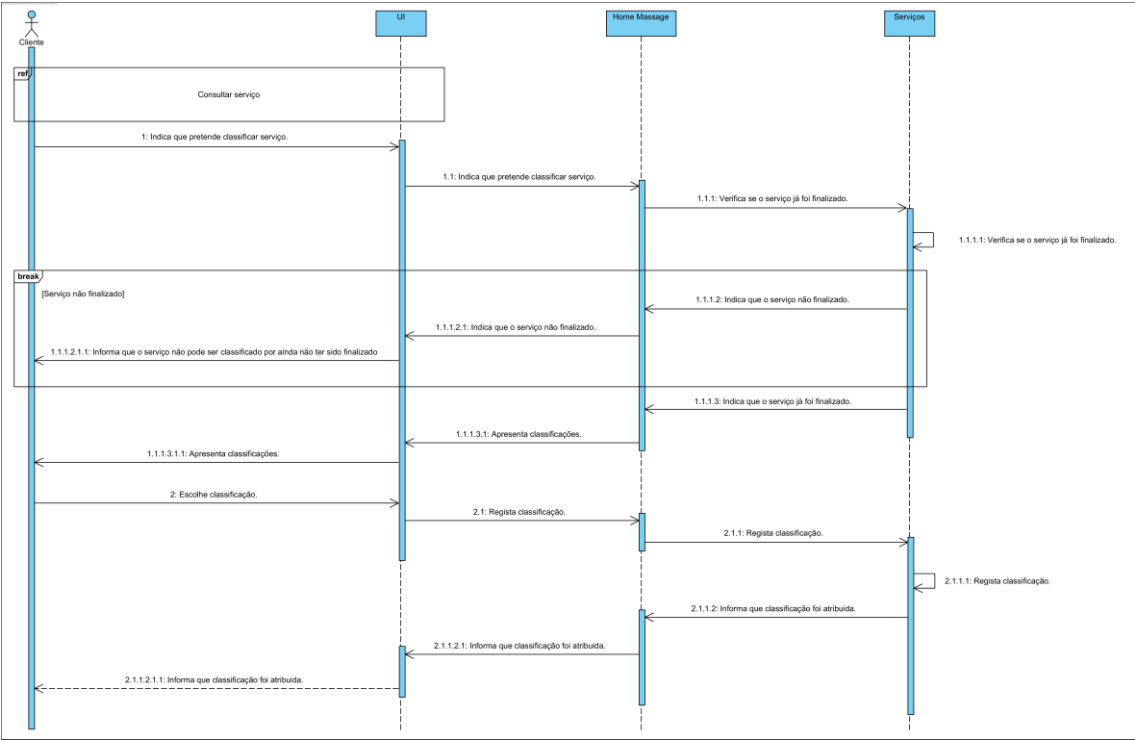


Figura 18 - DSS de "Classificar serviço".

## 8.9. Consultar localização

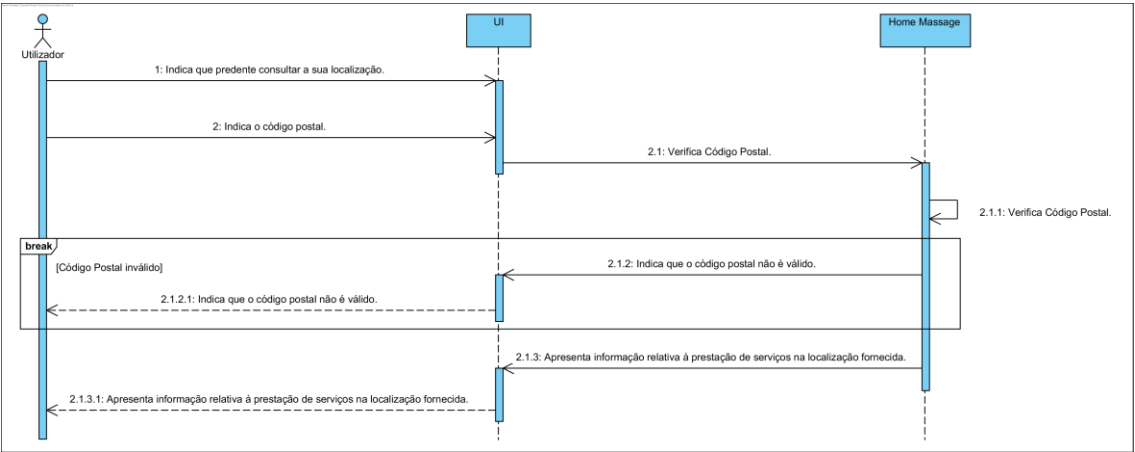


Figura 19 - DSS de "Consultar localização".

# 8.10. Consultar catálogo

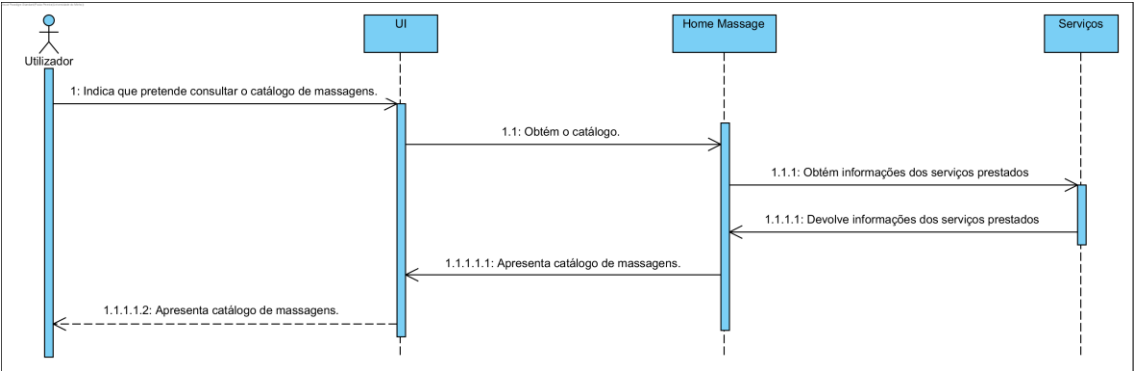


Figura 20 - DSS de "Consultar catálogo".

# 8.11. Consultar funcionamento

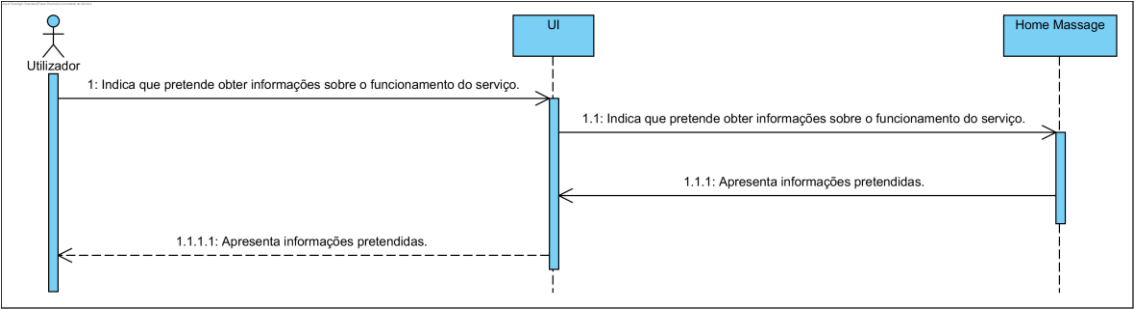


Figura 21 - DSS de "Consultar funcionamento".

### 8.12. Obter direções

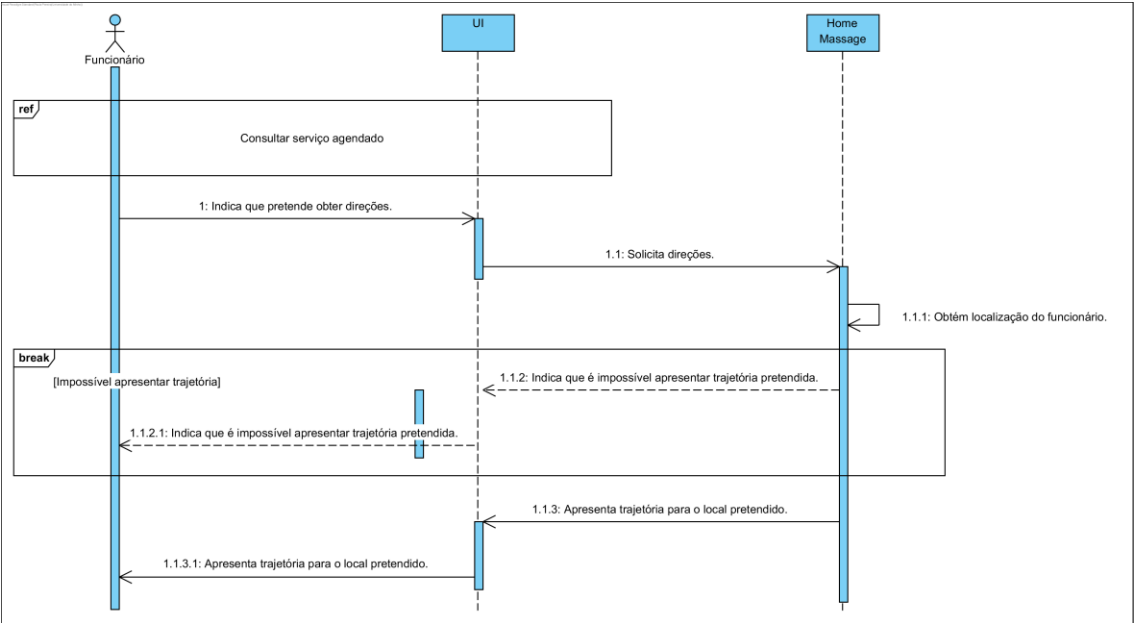


Figura 22 - DSS de "Obter direções".

### 8.13. Consultar serviço agendado

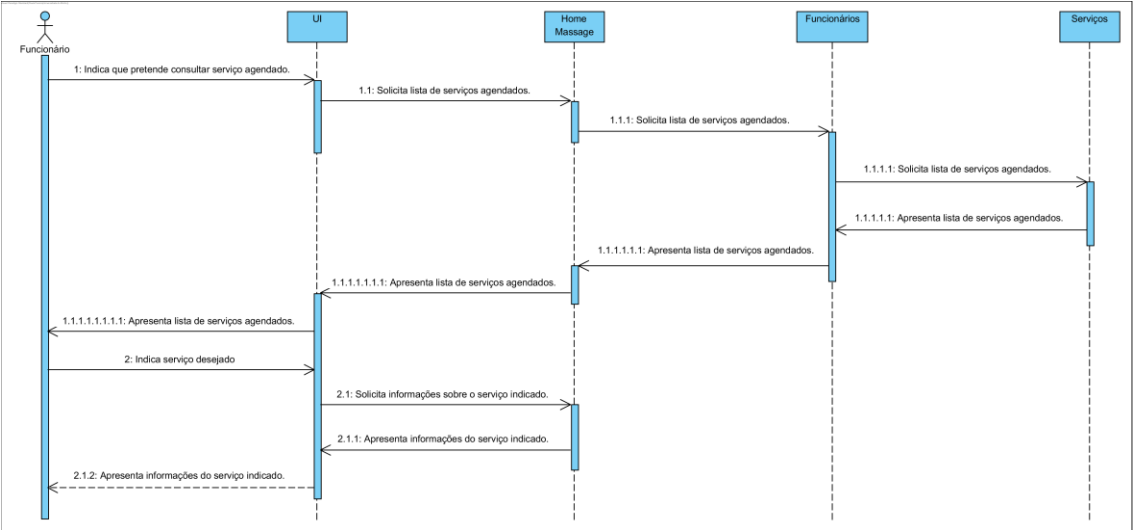


Figura 23 - DSS de "Consultar serviço agendado".



## 8.14. Finalizar serviço

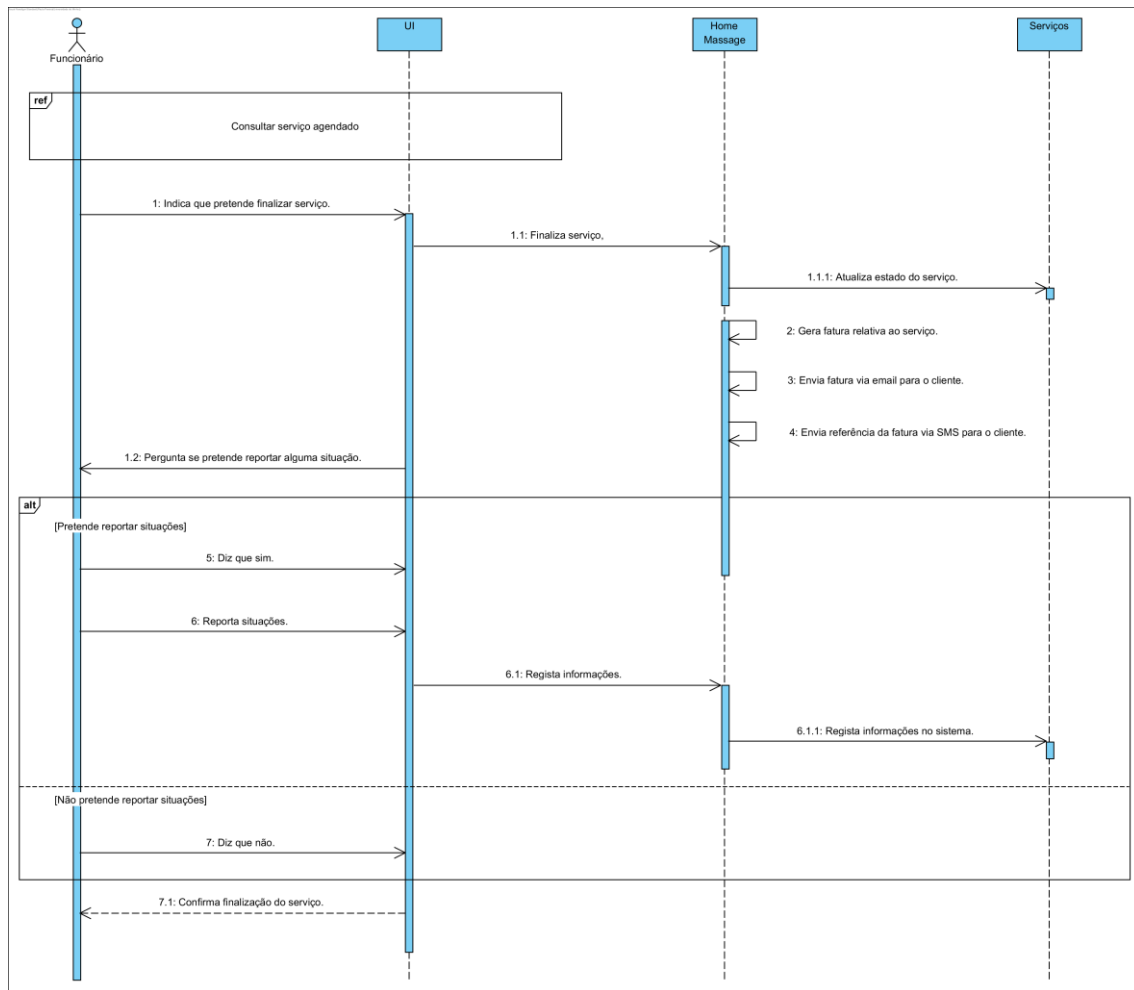


Figura 24 - DSS de "Finalizar serviço".

## 8.15. Registrar funcionário

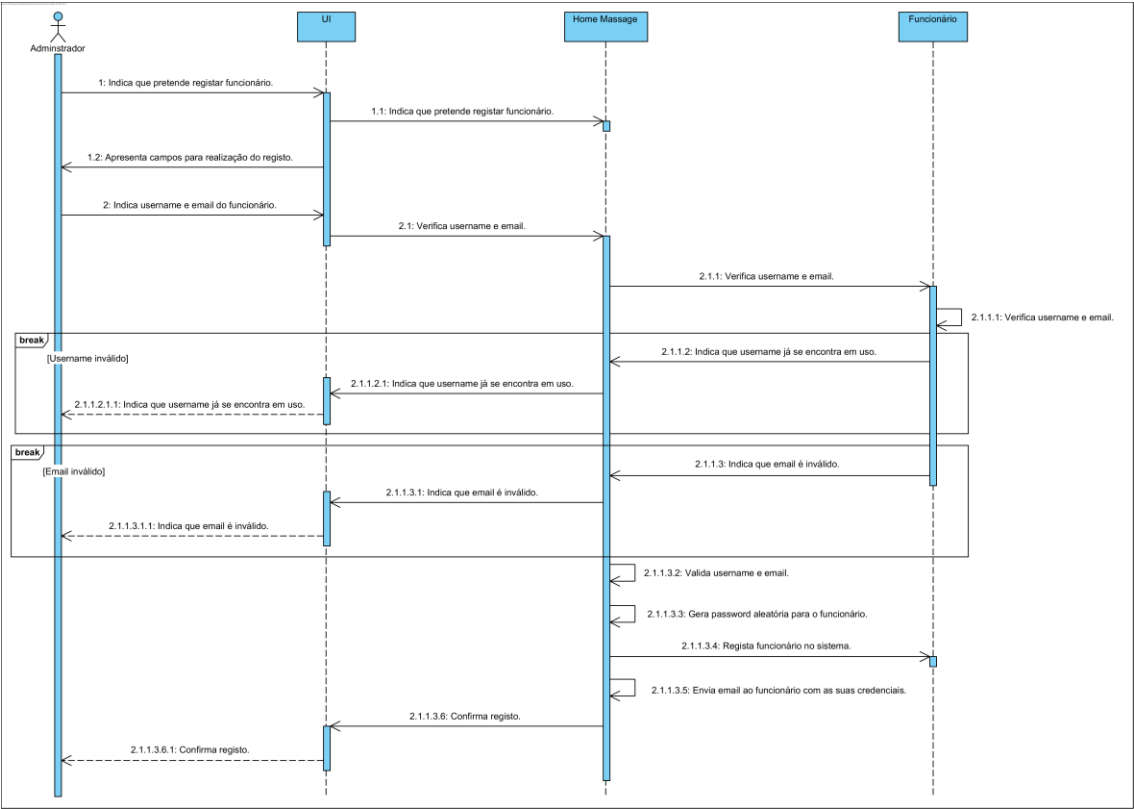


Figura 25 - DSS de "Registrar funcionário".

## 8.16. Inativar funcionário

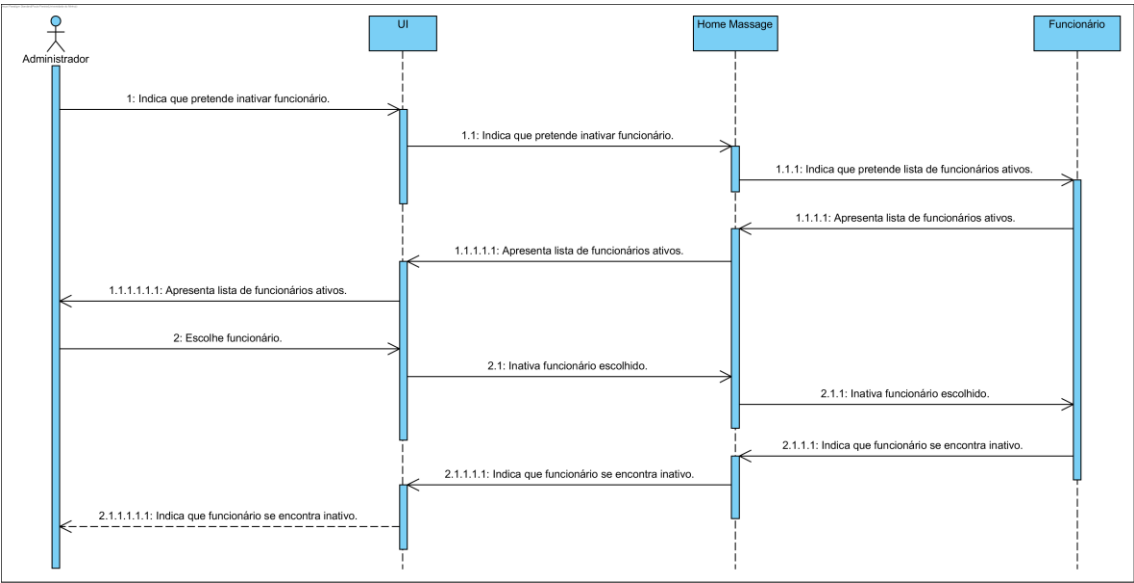


Figura 26 - DSS de "Inativar funcionário".

## 9. Diagrama de classes

Após a análise dos requisitos e a especificação dos vários *use cases*, procedemos à elaboração de um diagrama de classes, onde foram representadas as entidades que consideramos serem potenciais classes que a implementar.

Para além disso, e já a pensar na implementação do código e nas informações que serão necessárias guardar em relação a cada uma destas entidades, foram representados os seus atributos.

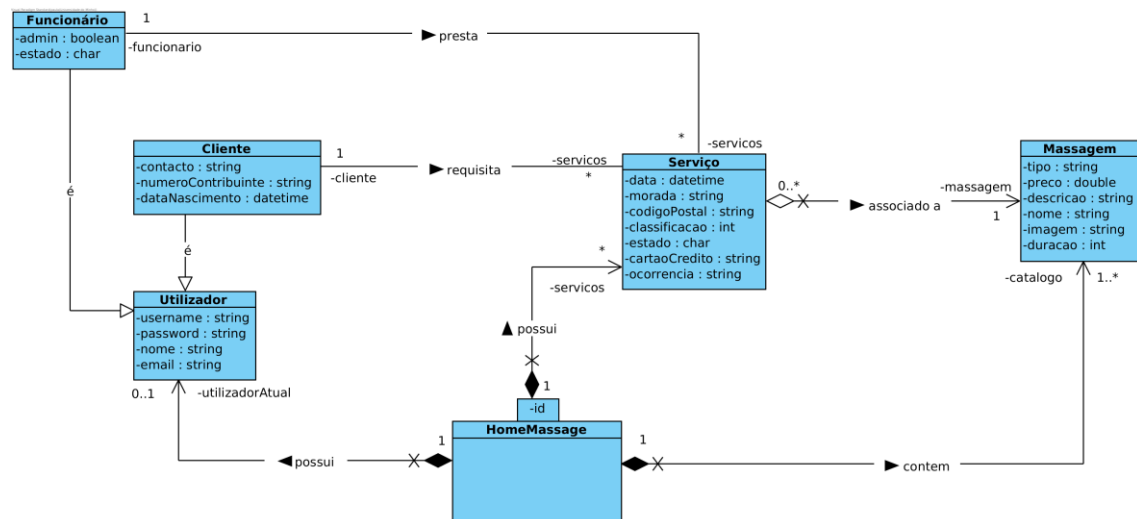


Figura 27 - Diagrama de classes.

## 10. Base de dados

Neste capítulo iremos abordar todos os aspetos relacionados com a modelação da base de dados que irá suportar a nossa aplicação.

Tendo em conta a aplicação que pretendemos desenvolver, e tal como já foi referido nos capítulos anteriores, será necessária uma base de dados capaz de armazenar todos os dados relacionados com os clientes, funcionários, serviços oferecidos e serviços prestados.

### 10.1. Levantamento e análise de requisitos

Nesta secção iremos proceder ao levantamento e análise dos requisitos que a base de dados desenvolvida deve verificar. Estes requisitos foram levantados a partir de várias conversas com o cliente e através da análise das várias funcionalidades que a aplicação irá suportar.

#### 10.1.1 Requisitos de descrição

A base de dados a implementar deverá permitir armazenar um conjunto alargado de informações relacionados quer com os vários utilizadores da aplicação, quer com os serviços oferecidos e respetivas requisições. Assim, temos que:

- Para cada **utilizador** devem ser guardadas as seguintes informações: *username*, *password*, nome e *email*; no caso de um **funcionário**, deverão ainda guardar-se informação sobre os serviços já prestados e os serviços agendados e ainda se se encontra ativo ou não na empresa; se se tratar de um **cliente** registado deverá guardar-se ainda toda a informação relacionada com os serviços já realizados e os serviços agendados, assim como o seu contacto telefónico, e ainda número de contribuinte e data de nascimento;
- Para cada **serviço** devem ser guardadas as seguintes informações: cliente que requisita o serviço, funcionário que irá prestar o serviço, data de prestação do serviço, localização onde o serviço será prestado (endereço e código postal),

mensagem a realizar, cartão de crédito utilizado para o pagamento e estado do serviço;

- Para cada **mensagem** disponível deverá guardar-se o seu nome, preço, duração, descrição e imagem descritiva.

### **10.1.2 Requisitos de exploração**

As informações guardadas na base de dados deverão permitir obter vários tipos de informações, tais como:

- quais os serviços associados a um determinado cliente;
- quais os serviços associados a um determinado funcionário;
- quais as mensagens disponíveis;
- a informação base de um serviço;
- os dados necessários à emissão de uma fatura;
- quais os funcionários ativos e inativos;
- as informações relacionadas com cada mensagem;
- quais os funcionários disponíveis numa determinada data.

### **10.1.3 Requisitos de controlo**

De modo a garantir a segurança e o bom funcionamento da base de dados, é necessário que os diversos tipos de utilizador da mesma possuam diferentes privilégios.

Assim, para a nossa base de dados consideramos que cada um dos três atores do sistema possui diferentes permissões. São estas:

#### **Administrador**

- Inserir/Atualizar informação dos funcionários.

#### **Funcionário**

- Atualizar Serviços (aquando da finalização dos mesmos).

#### **Cliente**

- Inserir/Atualizar informação do seu perfil.
- Inserir novos Serviços (aquando da requisição dos mesmos).

## 10.2. Modelação conceptual

Para a modelação conceptual, foi construído um diagrama Entidade-Relacionamento com o auxílio do programa “*brModelo*”. Este modelo é o resultado de várias etapas, sendo elas:

- Identificação das entidades do projeto;
- Identificação dos relacionamentos do projeto;
- Associação dos atributos às entidades e/ou relacionamentos;
- Determinação do domínio dos atributos;
- Determinação de chaves candidatas e primárias;
- Identificação da existência ou não de redundância no modelo.

### 10.2.1 Identificação e caracterização das entidades

Analisando os requisitos necessários para a criação da base de dados, identificamos as seguintes entidades: Utilizador, Cliente, Funcionário, Serviço e Mensagem.

Entidade	Descrição
Cliente	Representa um cliente da Zen*.
Funcionário	Representa um funcionário da Zen*.
Utilizador	Representa um utilizador da aplicação.
Serviço	Representa um serviço requisitado por um cliente e prestado por um funcionário.
Mensagem	Representa uma das mensagens disponíveis.

De maneira sucinta temos que: “São oferecidos vários serviços, associados a mensagens, que são requisitados pelos clientes e prestados pelos funcionários”.

### 10.2.2 Identificação e caracterização dos relacionamentos

Após termos definidas quais as entidades da nossa base de dados, prosseguimos para o estabelecimento de relações entre as mesmas. Assim sendo, têm-se os seguintes relacionamentos:

#### Cliente “Requisita” Serviço

Relacionamento entre Cliente e Serviço. Utilizado para associar os clientes aos serviços por ele requisitados.

Possui cardinalidade um para muitos, indicando que um cliente pode requisitar vários serviços e que um serviço é sempre requisitado por um cliente.

Tem participação opcional em Cliente e obrigatória em Serviço, indicando que um cliente pode ou não requisitar serviços, mas um serviço está sempre associado a uma requisição de serviço.

#### **Funcionário “Realiza” Serviço**

Relacionamento entre Funcionário e Serviço. Utilizado para associar os funcionários aos serviços agendados e prestados por si.

Possui cardinalidade um para muitos, indicando que um funcionário pode ter agendados/prestar vários serviços e que um serviço é sempre agendado/prestado por um funcionário.

Tem participação opcional em Funcionário e obrigatória em Serviço, indicando que um funcionário pode ou não realizar ou não serviços, mas um serviço é sempre realizado por um funcionário.

#### **Serviço “Associado a” Mensagem**

Relacionamento entre Serviço e Mensagem. Utilizado para associar os Serviço à Mensagem correspondente.

Possui cardinalidade um para muitos, indicando que um serviço é sempre associado a uma única mensagem e que uma mensagem pode ser associada a vários serviços.

Tem participação obrigatória em Serviço e opcional em Mensagem, uma vez que um serviço está sempre associado a uma mensagem, mas uma mensagem pode não estar associada a nenhum serviço.

Entidade 1	Relação	Entidade 2	Multiplicidade
Cliente	requisita	Serviço	(1, 1) – (0, N)
Funcionário	realiza	Serviço	(1, 1) – (0, N)
Serviço	Associado a	Mensagem	(0, N) – (1, 1)

### 10.2.3 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Definidas as entidades e os relacionamentos estabelecidos entre elas, procedemos à identificação e caracterização dos seus atributos.

Entidade	Atributo	Descrição	Tipo	Domínio	Extras
Utilizador	Id_Utilizador	Identificador único do utilizador	Inteiro	Qualquer inteiro positivo	
	Username	Identificador único do utilizador	Texto	Qualquer texto	
	Password	Password	Texto	Qualquer texto	
	Nome	Nome do utilizador	Texto	Qualquer nome	
	Email	Endereço eletrónico do utilizador	Texto	Formato <i>email</i>	
Cliente	Contacto	Contacto telefónico do cliente	Texto	Qualquer contacto telefónico	
	Número Contribuinte	Número de contribuinte a utilizado na faturação dos serviços	Texto	Qualquer nº de contribuinte	
	Data de Nascimento	Data de nascimento do utilizador	Data	Qualquer data	
Funcionário	Estado	Estado do funcionário na empresa	Caracter	'A' se ativo, 'I' se inativo	
Serviço	Id_Serviço	Identificador único do serviço	Inteiro	Qualquer Inteiro positivo	
	Data	Data em que é prestado o serviço	Data	Qualquer data	
	Cartão Crédito	Número de cartão de crédito utilizado para o pagamento	Texto	Qualquer nº de cartão de crédito	
	Localização	Local onde é prestado o serviço	Texto	Qualquer localização	Composto (Endereço e Código Postal)
	Estado	Estado do serviço	Caracter	0 se 'Pendente', 1 se 'Terminado'	
	Classificação	Classificação atribuída ao serviço	Inteiro	Inteiro de 0 a 5	
	Ocorrência	Ocorrências na prestação do serviço	Texto	Qualquer texto	
Massagem	Id_Massagem	Identificador único da massagem	Inteiro	Qualquer inteiro positivo	
	Nome	Nome da massagem	Texto	Qualquer nome	
	Preço	Preço da massagem	Texto	Qualquer decimal. 2 casas decimais	
	Duração	Duração da massagem	Inteiro	Qualquer inteiro positivo	
	Descrição	Descrição da massagem	Texto	Qualquer texto	
	Imagem	Imagem ilustrativa da massagem	Imagem	Qualquer imagem	

Tabela 19 - Dicionário de dados dos atributos das entidades.

Como se pode verificar na Tabela 19, foram utilizados os atributos iniciados por "Id", cujo tipo de dados é um número inteiro, para identificar unicamente cada instância das diversas entidades. Estes atributos são, portanto, as **chaves primárias** de cada entidade.



No entanto, em certas entidades, existem outros atributos que podem identificar unicamente cada instância da tabela, os quais se designam por **chaves alternativas**. Estes atributos são:

#### **Cliente**

- Username - cada cliente possui um username único.
- Email - cada cliente possui um email único.
- Contacto - cada cliente possui um contacto único.

#### **Funcionario**

- Username - cada funcionário possui um username único.
- Email - cada funcionário possui um email único.

#### **Servico**

- Funcionario e Data - em conjunto formam uma chave candidata composta, visto que um registo em Servico pode ser identificado pela data e o funcionário em questão.

#### **Massagem**

- Nome - cada mensagem possui um nome único.

Apesar dos atributos referidos acima serem únicos para cada entrada das entidades referidas, optamos por usar os atributos inicialmente referidos como chaves primárias devido ao facto de serem atributos simples com valores numéricos (ao contrário dos outros, que são valores textuais).

### **10.2.4 Detalhe ou generalização de entidades**

Com base na recolha da informação dos requisitos e na identificação das entidades do nosso sistema de base de dados, percebemos que as entidades “Cliente” e “Funcionário”, deve herdar todos os atributos da entidade “Utilizador”.

Assim sendo, foi tomada a decisão de proceder à generalização das entidades com o objetivo de, aquando a modelação lógica da base de dados, surgirem duas tabelas independentes relativas às informações dos clientes e funcionários.

## 10.2.5 Diagrama ER

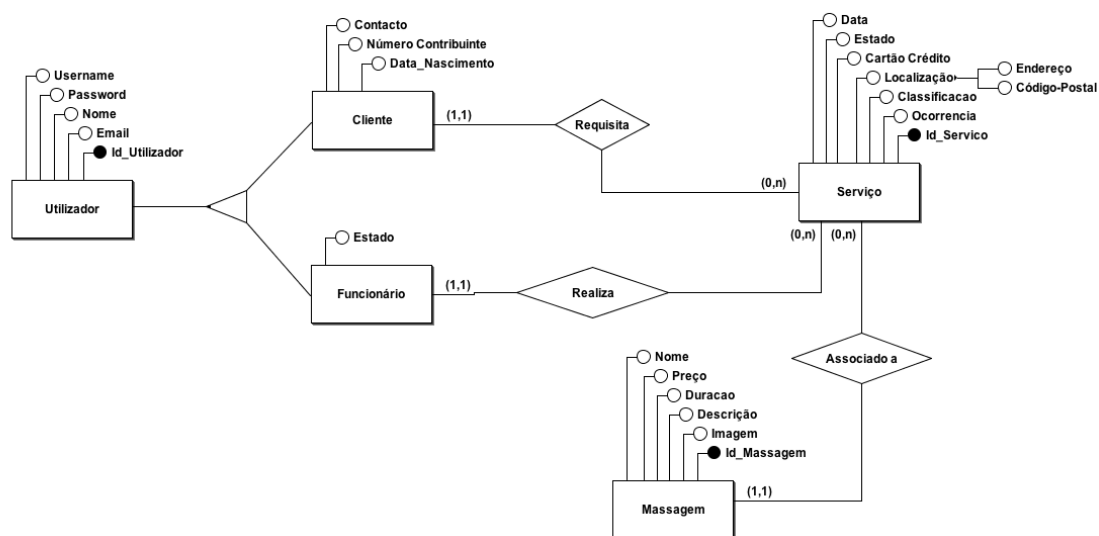


Figura 28 - Diagrama ER concebido com o software “brModelo”.

Através do diagrama apresentado, é possível observar as Entidades, os Relacionamentos e os Atributos apresentados nas seções anteriores, obtendo-se assim uma melhor percepção da base de dados a desenvolver e do conjunto de informações que esta permitirá armazenar.

## 10.3. Modelação lógica

Nesta secção iremos apresentar informação relativa à modelação lógica da nossa base de dados.

### 10.3.1 Construção e validação de modelo de dados lógicos

O modelo lógico, foi concebido com base no modelo conceptual criado inicialmente. Para isso, foram cumpridas as seguintes regras:

- Cada entidade gera uma tabela;
- Relacionamentos 1 para N, levam a implementação de uma chave estrangeira na tabela que possui maior cardinalidade.

Obtiveram-se assim as seguintes tabelas:

**Cliente:**

Coluna	Tipo de Dados	Chave	Null
<b>Id_Cliente</b>	int	Candidata, Primária	Não
<b>Username</b>	varchar(50)	Candidata	Não
<b>Password</b>	varchar(50)		Não
<b>Nome</b>	varchar(50)		Não
<b>Email</b>	varchar(50)	Candidata	Não
<b>Contacto</b>	int	Candidata	Não
<b>Numero_Contribuinte</b>	varchar(9)		Não
<b>Data_Nascimento</b>	date		Não

Tabela 20 – Informação sobre a tabela “Cliente”.

**Funcionario:**

Coluna	Tipo de Dados	Chave	Null
<b>Id_Funcionario</b>	int	Candidata, Primária	Não
<b>Username</b>	varchar(50)	Candidata	Não
<b>Password</b>	varchar(50)		Não
<b>Nome</b>	varchar(50)		Não
<b>Email</b>	varchar(50)	Candidata	Não
<b>Estado</b>	bit		Não

Tabela 21 – Informação sobre a tabela “Funcionario”.

**Servico:**

Coluna	Tipo de Dados	Chave	Null
<b>Id_Servico</b>	int	Candidata, Primária	Não
<b>Cliente</b>	int	Estrangeira	Não
<b>Funcionario</b>	int	Estrangeira	Não
<b>Massagem</b>	int	Estrangeira	Não
<b>Data</b>	datetime		Não
<b>Cartao_Credito</b>	varchar(19)		Não
<b>Estado</b>	bit		Não
<b>Endereco</b>	varchar(100)		Não
<b>Codigo_Postal</b>	varchar(8)		Não
<b>Classificacao</b>	Int		
<b>Ocorrencia</b>	Varchar(250)		

Tabela 22 – Informação sobre a tabela “Servico”.

## Massagem:

Coluna	Tipo de Dados	Chave	Null
Id_Massagem	int	Candidata, Primária	Não
Nome	varchar(15)	Candidata	Não
Preço	money		Não
Duracao	Int		Não
Descricao	nvarchar(250)		Não
Imagem	image		Não

Tabela 23 - Informação sobre a tabela "Massagem".

### 10.3.2 Desenho do modelo lógico

O modelo lógico obtido foi o seguinte:

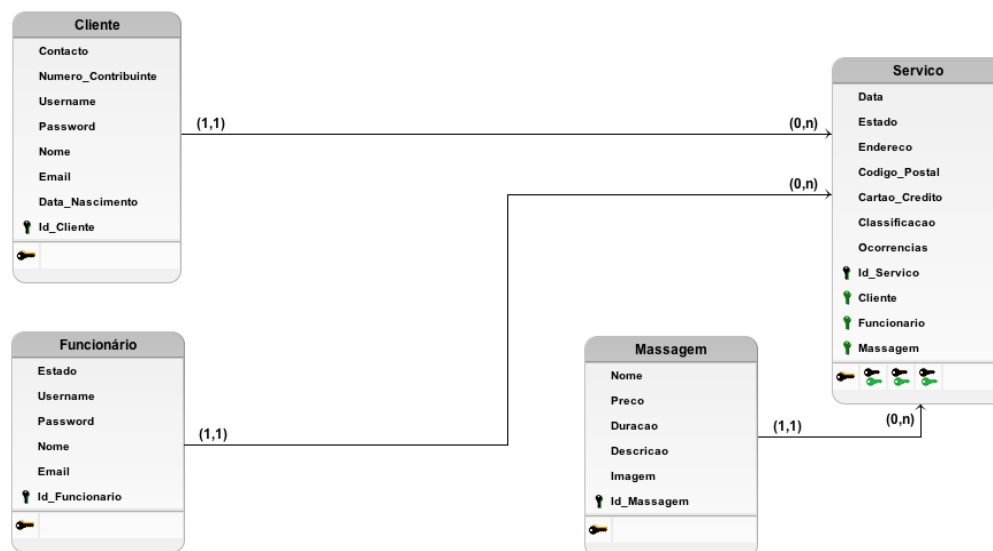


Figura 29 - Modelo lógico concebido com o software "brModelo".

### 10.3.3 Validação do modelo através da normalização

De modo a evitar redundâncias na base de dados e minimizar anomalias nas operações de modificação de dados o modelo tem de responder a um conjunto de regras, chamadas “formas normais”.

O modelo lógico apresentado encontra-se normalizado, uma vez que verifica as três formas normais.

### **Primeira Forma Normal**

Todos os atributos presentes nas tabelas da Base de Dados são atômicos. De modo a que esta forma normal fosse verificada, os atributos compostos foram "decompostos" em atributos simples. Assim, garantimos que a Base de Dados se encontra na 1FN.

### **Segunda Forma Normal**

Visto que na Base de Dados não se verificam dependências parciais, devido à inexistência de chaves primárias compostas, e tendo em conta que esta já se respeita a 1FN, garantimos que se encontra na 2FN.

### **Terceira Forma Normal**

Sabendo que a Base de Dados se encontra 2FN, verificaram-se as dependências transitivas. Tendo em conta que todos os atributos não-chave dependem apenas da chave primária (ou chaves candidatas), podemos garantir que estas dependências são inexistentes e que a Base de Dados se encontra na 3FN.

## 11. Idealização da interface

Nesta secção apresentamos algumas figuras que permitem desde já ter uma ideia de como será a interface gráfica da aplicação.

### 11.1. Menu inicial

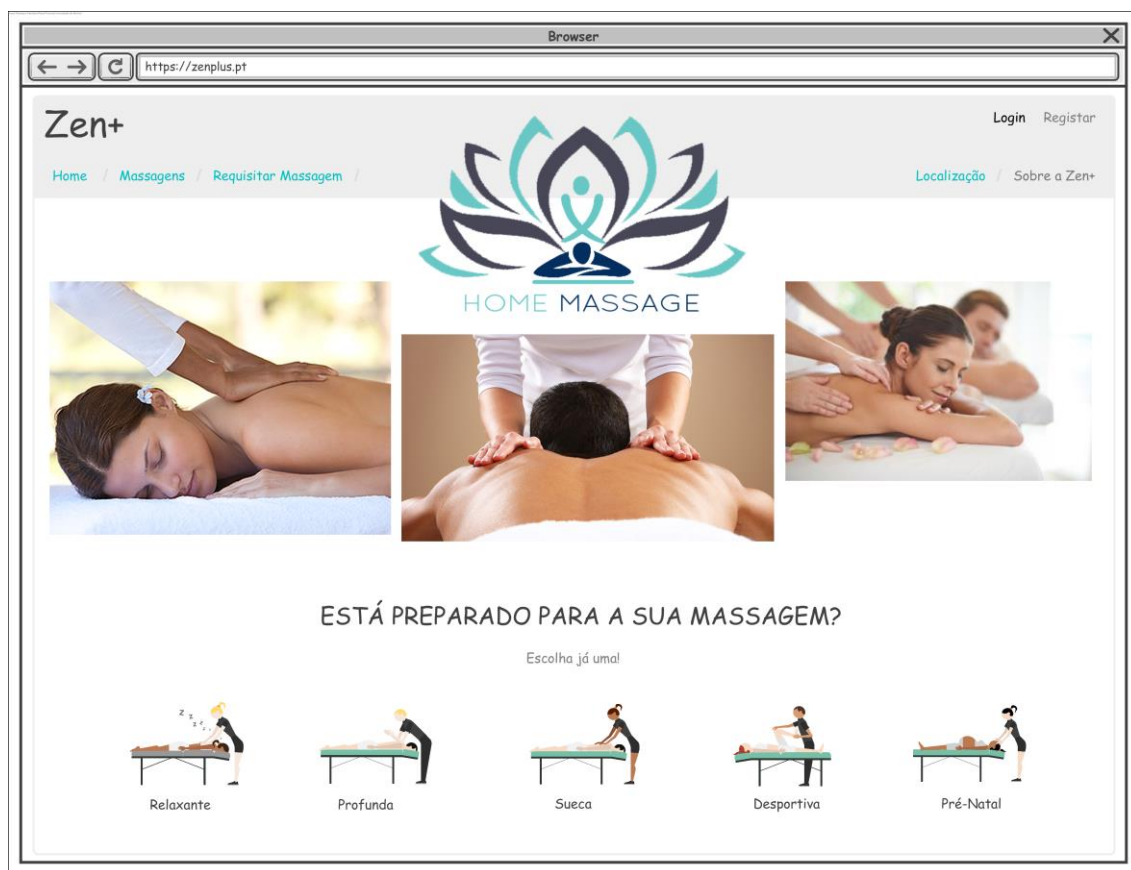
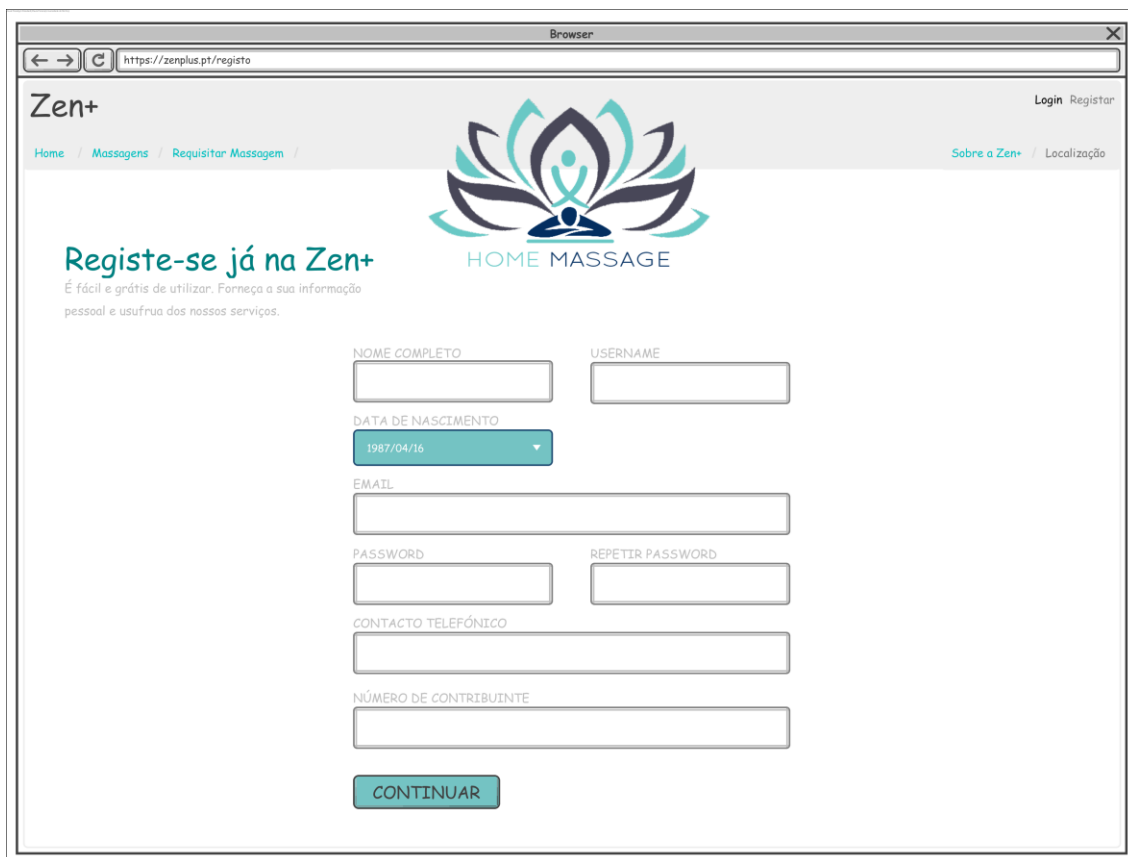


Figura 30 - Mockup - Menu inicial.

A Figura 30 diz respeito à página principal da aplicação, onde o utilizador terá acesso a alguma informação sobre os serviços oferecidos e onde terá acesso a várias páginas associadas às várias funcionalidades da aplicação.

## 11.2. Menu de registo na aplicação

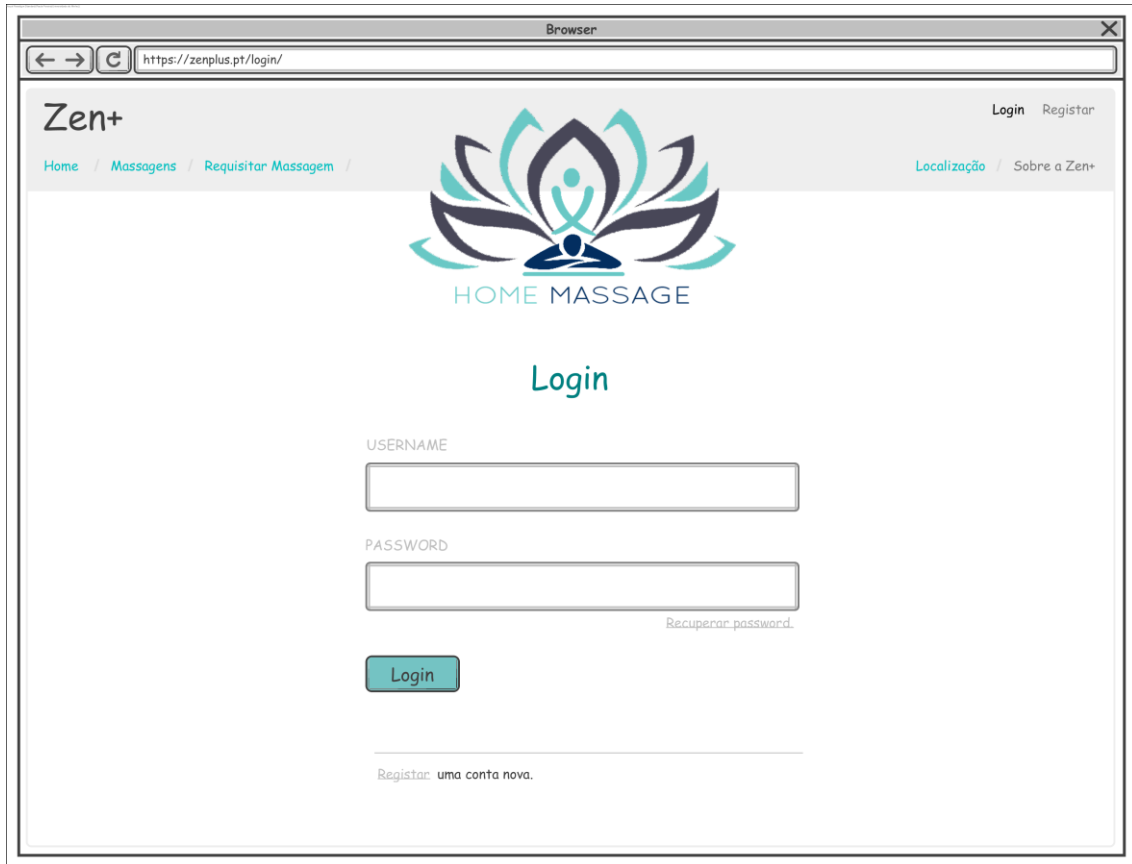


The image shows a web browser window displaying the registration page for Zen+ HOME MASSAGE. The browser's address bar shows the URL <https://zenplus.pt/registo>. The page header includes the Zen+ logo and navigation links: Home, Massagens, Requisitar Massagem, Sobre a Zen+, and Localização. The main heading is "Registe-se já na Zen+", followed by a subtext: "É fácil e grátis de utilizar. Forneça a sua informação pessoal e usufrua dos nossos serviços." The registration form consists of the following fields: NOME COMPLETO, USERNAME, DATA DE NASCIMENTO (with a date picker showing 1987/04/16), EMAIL, PASSWORD, REPETIR PASSWORD, CONTACTO TELEFÓNICO, and NÚMERO DE CONTRIBUINTE. A "CONTINUAR" button is located at the bottom of the form.

Figura 31 - *Mockup* - Menu de registo na aplicação.

A Figura 31 diz respeito à funcionalidade Registrar utilizadores na aplicação, pelo que é apresentado um pequeno formulário no qual o utilizador indica todas as informações necessárias ao seu registo.

### 11.3. Menu de autenticação na aplicação



The image shows a web browser window displaying the login page of an application named 'Zen+'. The browser's address bar shows the URL 'https://zenplus.pt/login/'. The page has a light gray header with the 'Zen+' logo on the left and 'Login' and 'Registar' links on the right. Below the header, there is a navigation menu with links: 'Home', 'Massagens', 'Requisitar Massagem', 'Localização', and 'Sobre a Zen+'. The main content area features a large, stylized logo of a person in a yoga pose, with the text 'HOME MASSAGE' below it. The word 'Login' is displayed in a large, teal font. Below this, there are two input fields: 'USERNAME' and 'PASSWORD'. A teal 'Login' button is positioned below the password field. To the right of the password field, there is a link that says 'Recuperar password.'. At the bottom of the page, there is a link that says 'Registar: uma conta nova.'.

Figura 32 - *Mockup* - Menu de autenticação na aplicação.

A Figura 32 diz respeito à funcionalidade Autenticar utilizadores na aplicação, pelo que é apresentado um pequeno formulário no qual o utilizador fornece as suas credenciais, que após serem validadas pelo sistema lhe permitem aceder à sua área pessoal.



## 11.4. Menu de requisição de serviço

**Zen+**  
Home / Massagens / Requisitar Massagem /

Olá, Utilizador

Sobre a Zen+ / Localização

**HOME MASSAGE**

### Requisitar massagem

Preencha todos os campos.

NÚMERO DO CARTÃO DE CRÉDITO:

VALIDADE:

LOCAL DO SERVIÇO:

Apenas disponível na zona de Braga, Portugal.

SELECIONE O DIA E HORA

Junho 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

10:00H

SELECIONE O TIPO DE MASSAGEM

- ☒ RELAXANTE
- ☐ PROFUNDA
- ☐ SUECA
- ☐ DESPORTIVA
- ☐ PRÉ-NATAL

**VALIDAR**

Figura 33 - Mockup - Menu de requisição de serviço.

A Figura 33 diz respeito à funcionalidade Requisitar serviços, pelo que é apresentado um pequeno formulário no qual o utilizador indica todas as informações necessárias à solicitação do serviço.

## 11.5. Menu de consulta de serviço agendado

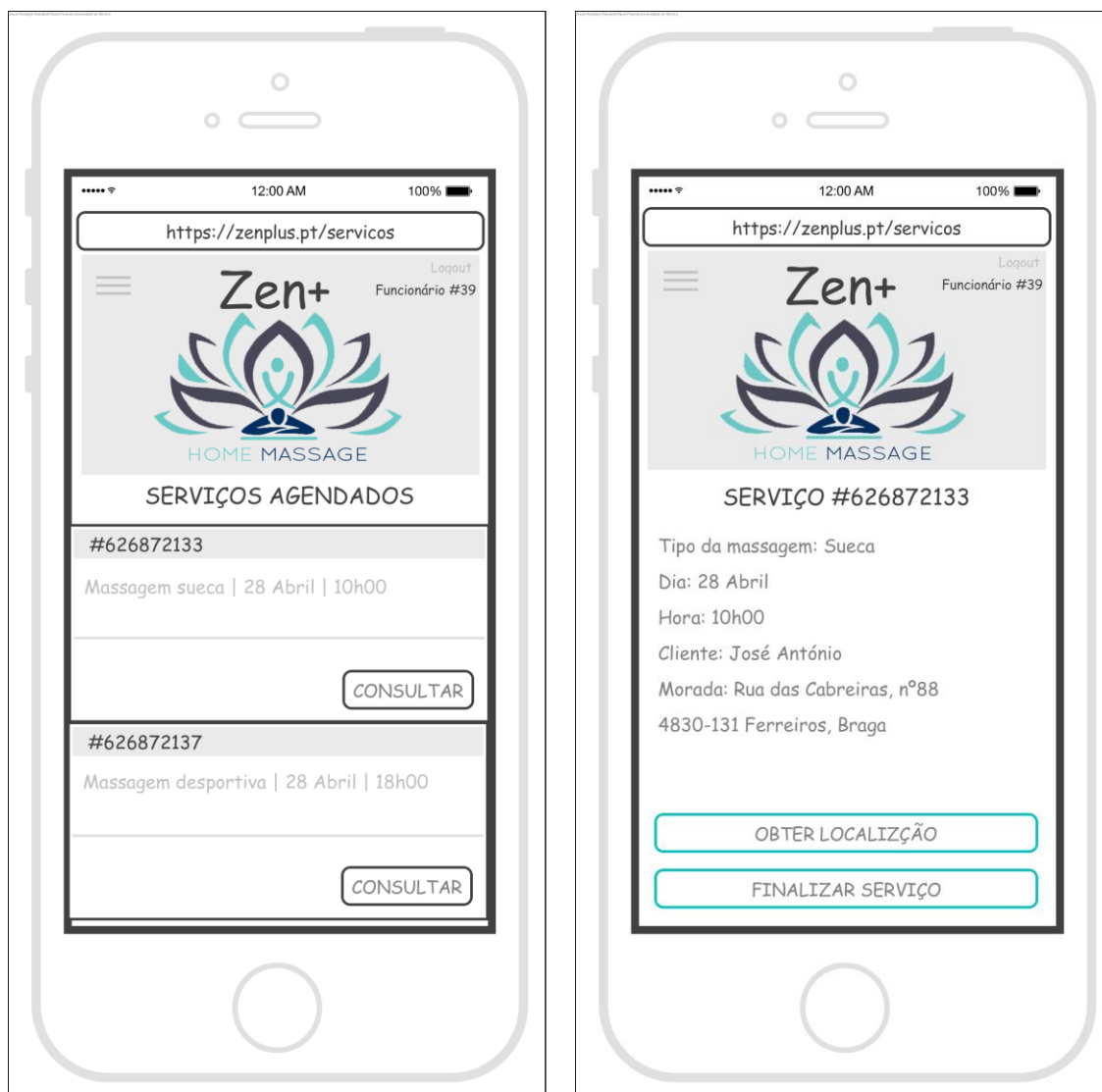


Figura 34 - Mockup - Menu de consulta de serviço agendado.

A Figura 34 diz respeito à funcionalidade Consultar serviços agendados, que permite a um funcionário autenticado na aplicação obter informações relativas aos serviços agendados.

## 12. Implementação

Nesta secção poderá encontrar todo um conjunto de informações relativa à implementação do sistema de *software*. Assim sendo, iremos apresentar a arquitetura do sistema, informação relacionada com a base de dados implementada, API's e ferramentas utilizadas, entre outros.

### 12.1. Modelo MVC

Para a implementação do sistema de *software*, baseamo-nos no padrão MVC (*Model-View-Controller*), um padrão de arquitetura que, promovendo a reutilização, organização e desempenho do código, consiste em separar a aplicação em três áreas distintas:

- **Model** – Representa os dados que a aplicação utiliza;
- **View** – Representa a interface gráfica com que o utilizador interage;
- **Controller** – Representa a lógica aplicacional que relaciona as três áreas distintas.

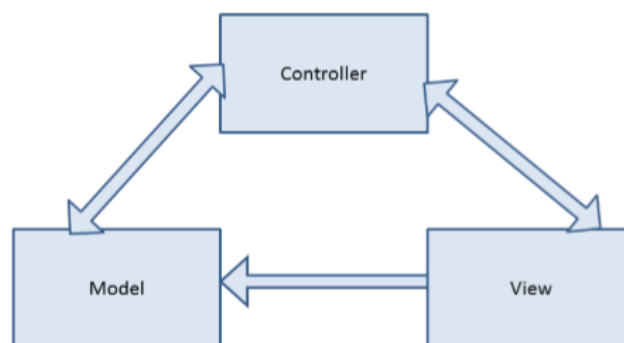


Figura 35 - Modelo MVC.

O *model* representa assim a camada de negócio e consiste nos dados da aplicação, lógica e funções. Para que a aplicação funcionasse corretamente foram então criados tantos modelos quantas entidades existem no sistema, para que cada um deles fosse responsável pela informação necessária à sua representação.

Por sua vez, a *view* pertence à camada de apresentação e possui todas as funcionalidades para a apresentação da informação aos utilizadores da aplicação, sendo que

essa informação é lhe fornecida por um controlador. Por cada entidade existente no sistema existe pelo menos uma *view*.

Por fim, o *controller* é o responsável por controlar todo o fluxo de informação no sistema, funcionando como um intermediário entre a informação do *model* e as respetivas *views*. Em cada controlador foi desenvolvido um método por cada operação necessária.

## 12.2. Base de dados

Para implementar a base de dados idealizada anteriormente recorreremos à ferramenta *Microsoft SQL Server*. Para tal, desenvolvemos um *script* de criação de acordo com a informação apresentada no capítulo Base de dados que pode ser consultado no anexo 0.

Para além do *script* de criação, foi ainda desenvolvido um *script* de povoamento com as informações base necessárias ao funcionamento do sistema, tais como as informações relativas às várias mensagens disponíveis, funcionários e clientes. Este *script* pode ser consultado no anexo II.

## 12.3. Ferramentas utilizadas

De modo a sermos capazes de obter o produto final que iremos apresentar no capítulo seguinte tivemos de recorrer ao uso de várias ferramentas.

O projeto foi desenvolvido utilizando a ferramenta *Microsoft Visual Studio 2017*, recorrendo para isso à *framework* .NET e utilizando as linguagens C#, Javascript e HTML.

### 12.3.1 Sistema de Representação de Mapas

Para que fosse possível apresentar a um funcionário a trajetória a seguir até um local indicado por um cliente para a prestação de um serviço, foi necessário recorrer à API *Bing Maps*.

Através desta API é possível obter a localização atual do funcionário e depois utilizá-la para determinar a trajetória a ser realizada até aos locais indicados pelos clientes para a realização dos vários serviços.

A implementação é realizada através da utilização de *scripts* javascript no HTML. Assim, obtém-se a localização do funcionário:

```
navigator.geolocation.getCurrentPosition(function (position) {  
    loc = new Microsoft.Maps.Location(  
        position.coords.latitude,  
        position.coords.longitude);  
});
```

E, cria-se uma rota:

```
Microsoft.Maps.loadModule('Microsoft.Maps.Directions', function () {
    directionsManager = new Microsoft.Maps.Directions.DirectionsManager(map);
    var origem = new Microsoft.Maps.Directions.Waypoint({ location: loc });
    directionsManager.addWaypoint(origem);
    var destino = new Microsoft.Maps.Directions.Waypoint({address: '...' });
    directionsManager.addWaypoint(destino);
    ...
    directionsManager.calculateDirections();
});
```

### 12.3.2 Sistema de envio de emails

Dada a necessidade de enviar *emails* automáticos de confirmação das requisições de serviços, envio das faturas associadas a cada um deles e ainda o envio das credenciais do funcionário após registo pelo administrador, foi necessário assegurar que o sistema era capaz de *emails* com as respetivas informações.

Assim sendo, recorreremos à classe *SmtpClient* para estabelecer a conexão ao servidor *smtp-mail.outlook.com*.

```
SmtpClient smtpClient = new SmtpClient();
smtpClient.Host = "smtp-mail.outlook.com";
smtpClient.Port = 587;
smtpClient.UseDefaultCredentials = false;
smtpClient.Credentials = new System.Net.NetworkCredential(...);
smtpClient.DeliveryMethod = SmtpDeliveryMethod.Network;
smtpClient.EnableSsl = true;
```

De seguida, através da classe *MailMessage* criamos o e-mail pretendido, adicionando-lhe remetente, destinatário, assunto e corpo.

```
MailMessage mail = new MailMessage();
mail.From = new MailAddress("homemessage_li4@hotmail.com");
mail.Subject = "Confirmação da requisição da mensagem";
mail.To.Add(new MailAddress(clientEmail));
mail.Body = "...";
smtpClient.Send(mail);
```

### 12.3.3 Sistema de envio de SMS

Dada a necessidade de enviar SMS's com as referências das faturas associadas aos serviços prestados, foi necessário recorrer a uma API que fosse capaz de enviar de forma automática as mensagens contendo as informações pretendidas. Para isso, recorreremos à API *Nexmo.API*.

```
SMS.Send(new SMS.SMSRequest{from= rem, to = dest, text= msg});
```

## 13. Produto final

Ainda que tivesse sido necessária a realização de pequenas alterações face à especificação realizada na segunda fase deste projeto, consideramos que a arquitetura final do sistema de *software* desenvolvido vai de encontro ao que foi idealizado.

Terminada a fase de implementação, fomos capazes de apresentar um *software* que permite a qualquer utilizador interagir com o sistema através de um *browser*. Houve, por isso, a preocupação de apresentar aos vários utilizadores uma interface intuitiva e *user-friendly*, que permite uma navegação fácil e eficaz pelas várias páginas disponíveis. Assim sendo, a implementação que apresentamos, de forma geral, corresponde a um conjunto de botões e formulários que vão abrindo novas páginas de acordo com as funcionalidades implementadas.

Uma das principais vantagens da aplicação, prende-se com o facto de esta funcionar em qualquer dispositivo, desde que este possua um *browser*. Isto mostra-se como um ponto forte no que diz respeito ao número de utilizadores da aplicação, pois esta passa a estar acessível a todas as pessoas sem grandes esforços.

De seguida, apresentam-se várias figuras representativas da interface alcançada relativamente às principais funcionalidades.

### 13.1. Página Inicial

Na Figura 36 encontra-se a página inicial apresentada a qualquer utilizador que aceda à aplicação. Nesta, o utilizador tem acesso a vários botões associados às várias funcionalidades permitidas. Assim sendo, é possível, premindo o respetivo botão, aceder a um catálogo de mensagens, verificar se numa determinada localização é possível a prestação de serviços, consultar informações gerais sobre a requisição e prestação de serviços, requisitar mensagens, registar-se ou autenticar-se no sistema.



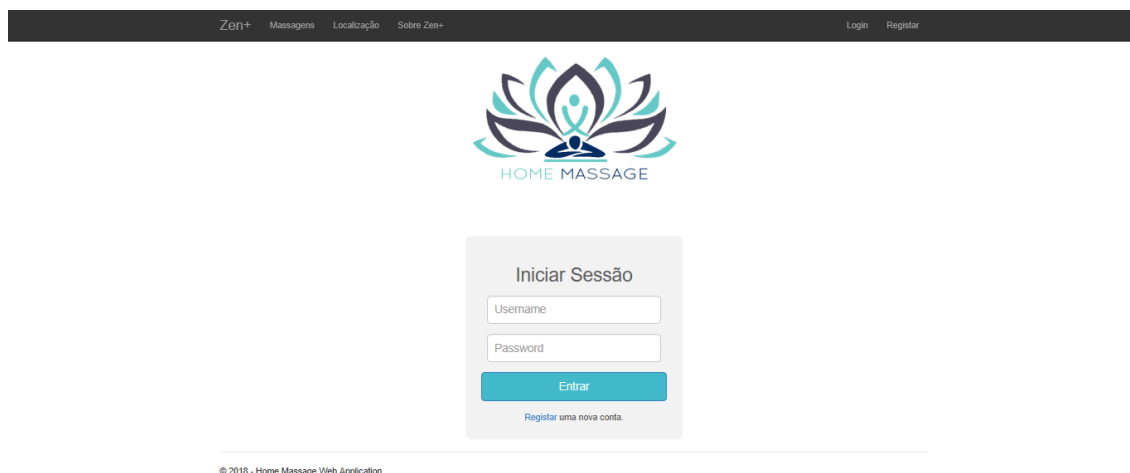
Figura 36 - Página inicial.

## 13.2. Registrar-se e autenticar-se na aplicação

Na Figura 37 encontra-se a página que é apresentada a um utilizador que pretenda registar-se no sistema. Como se pode verificar, este registo implica apenas que o utilizador preencha um pequeno formulário com as suas informações pessoais.

Figura 37 - Registo na aplicação.

Já na Figura 38 encontra-se a página de autenticação na aplicação. Para isso, é solicitado ao utilizador que indique o seu *username* e *password*, e caso estes sejam confirmados pelo sistema, o utilizador fica autenticado com sucesso.

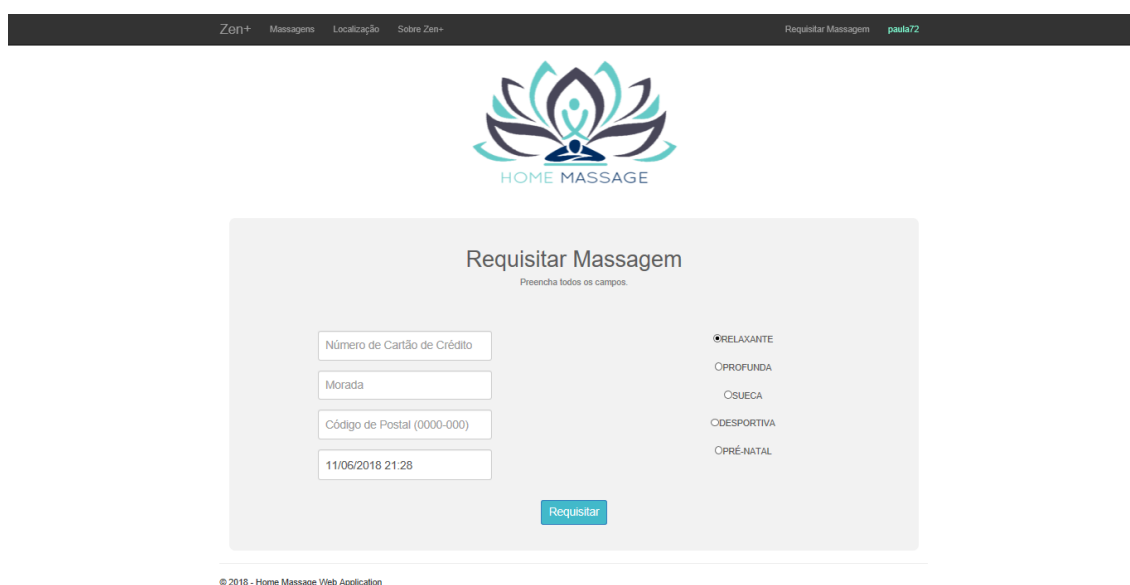


The screenshot shows the login page of the Home Massage application. At the top, there is a dark navigation bar with links: Zen+, Massagens, Localização, Sobre Zen+, Login, and Registrar. Below the navigation bar is the Home Massage logo, which features a stylized lotus flower in teal and blue. The main content area is a light gray box titled "Iniciar Sessão". It contains two input fields: "Username" and "Password". Below these fields is a blue "Entrar" button. Under the button, there is a link that says "Registrar uma nova conta." At the bottom of the page, there is a small copyright notice: "© 2018 - Home Massage Web Application".

Figura 38 - Autenticação na aplicação.

### 13.3. Requisitar massagem

Na Figura 39 encontra-se a página que deve ser utilizada por um cliente quando este pretende requisitar uma massagem. Esta página só é acessível quando o utilizador se encontra autenticado no sistema. A requisição consiste no preenchimento de um formulário em que são indicadas as preferências do cliente, tais como, massagem a realizar, cartão de crédito a utilizar na faturação, endereço e data.



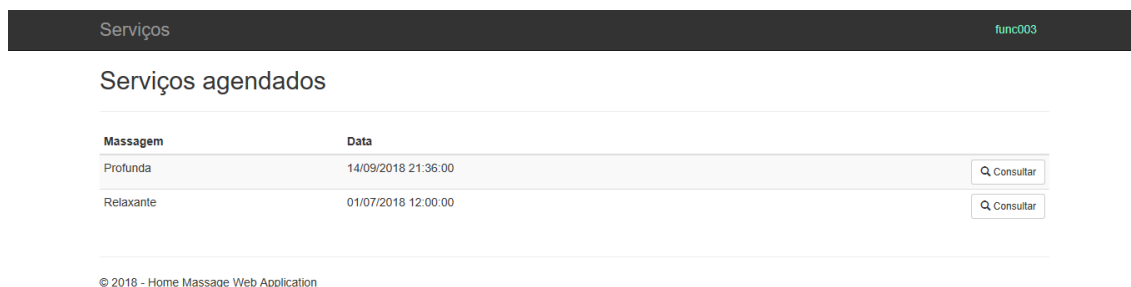
The screenshot shows the "Requisitar Massagem" page of the Home Massage application. At the top, there is a dark navigation bar with links: Zen+, Massagens, Localização, Sobre Zen+, and "Requisitar Massagem" (which is highlighted in green). Below the navigation bar is the Home Massage logo. The main content area is a light gray box titled "Requisitar Massagem" with the subtitle "Preencha todos os campos." It contains a form with four input fields: "Número de Cartão de Crédito", "Morada", "Código de Postal (0000-000)", and "11/06/2018 21:28". To the right of these fields is a list of radio buttons for selecting a massage type: "RELAXANTE" (selected), "OPROFUNDA", "OSUECA", "ODESPORTIVA", and "OPRÉ-NATAL". Below the form fields is a blue "Requisitar" button. At the bottom of the page, there is a small copyright notice: "© 2018 - Home Massage Web Application".

Figura 39 - Requisitar Massagem.



## 13.4. Finalizar serviço

No que diz respeito à finalização de serviços, esta acontece na área reservada de cada funcionário, idêntica à representada na Figura 40. Assim sendo, após a sua autenticação são lhe apresentados os vários serviços agendados, e algumas informações sobre estes.



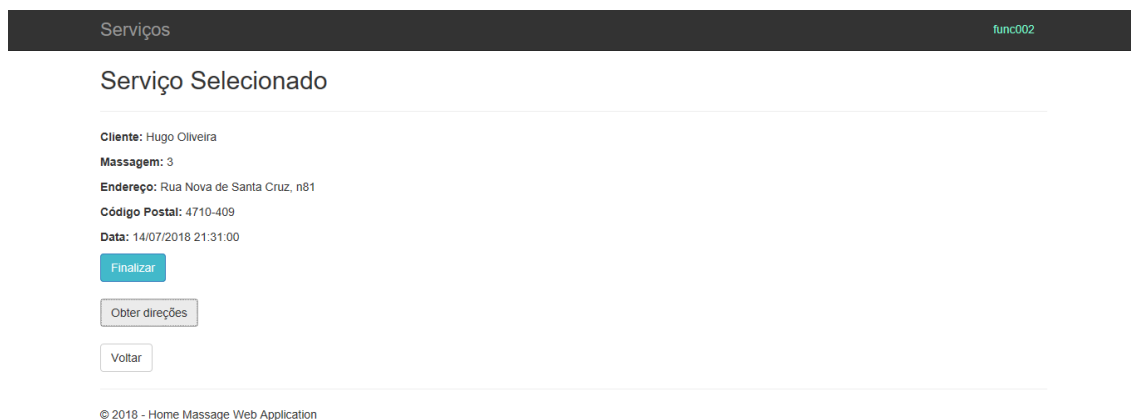
The screenshot shows a web application interface for a user named 'func003'. The page title is 'Serviços'. Below the title, there is a section 'Serviços agendados'. It contains a table with two columns: 'Massagem' and 'Data'. There are two rows of data. Each row has a 'Consultar' button to its right. At the bottom of the page, there is a copyright notice: '© 2018 - Home Massage Web Application'.

Massagem	Data	
Profunda	14/09/2018 21:36:00	<a href="#">Consultar</a>
Relaxante	01/07/2018 12:00:00	<a href="#">Consultar</a>

© 2018 - Home Massage Web Application

Figura 40 - Serviços agendados de um funcionário.

Ao selecionar um dos serviços é apresentada uma nova página como a representada na Figura 41, onde o funcionário pode consultar uma informação mais detalhada do serviço, dá-lo como finalizado, ou obter direções. Ao finalizar é ainda apresentada uma nova página semelhante à da Figura 42, em que o funcionário pode reportar alguma situação relativa ao serviço.



The screenshot shows a web application interface for a user named 'func002'. The page title is 'Serviço Selecionado'. It displays the following information: 'Cliente: Hugo Oliveira', 'Massagem: 3', 'Endereço: Rua Nova de Santa Cruz, n81', 'Código Postal: 4710-409', and 'Data: 14/07/2018 21:31:00'. Below this information, there are three buttons: 'Finalizar' (highlighted in blue), 'Obter direções', and 'Voltar'. At the bottom of the page, there is a copyright notice: '© 2018 - Home Massage Web Application'.

Cliente: Hugo Oliveira  
Massagem: 3  
Endereço: Rua Nova de Santa Cruz, n81  
Código Postal: 4710-409  
Data: 14/07/2018 21:31:00

[Finalizar](#)  
[Obter direções](#)  
[Voltar](#)

© 2018 - Home Massage Web Application

Figura 41 - Serviço selecionado.

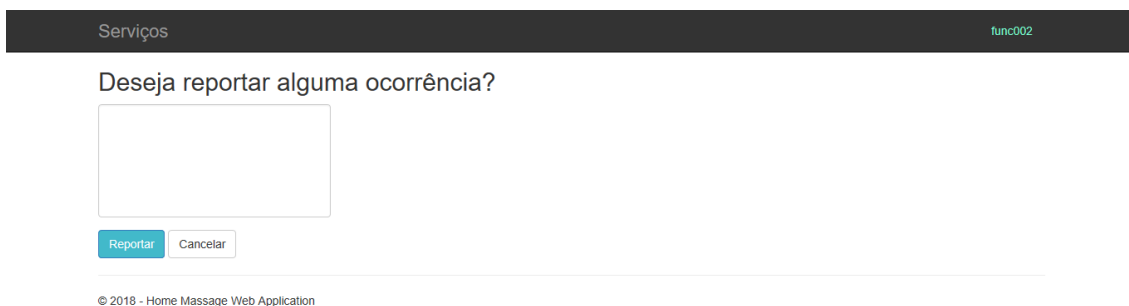


Figura 42 - Reportar situações durante a finalização de serviço.

## 13.5. Obter direções

Na Figura 43 encontra-se representada a página que é mostrada ao funcionário quando este indica que pretende obter direções para o local onde um determinado serviço deverá ser prestado. Tal como se pode verificar, a trajetória entre um determinado local A e um local B encontra-se representada a azul. É ainda possível fazer zoom e alterar o modo de apresentação do mapa.

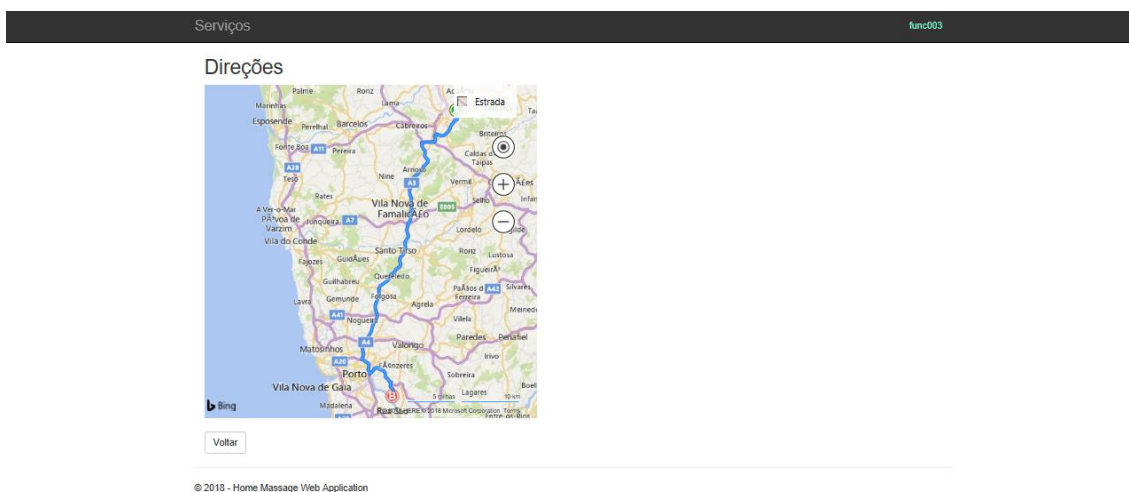


Figura 43 - Obter direções.

## 13.6. Outras funcionalidades

De seguida, apresentam-se figuras relativas a páginas associadas a outras funcionalidades de menor relevância.

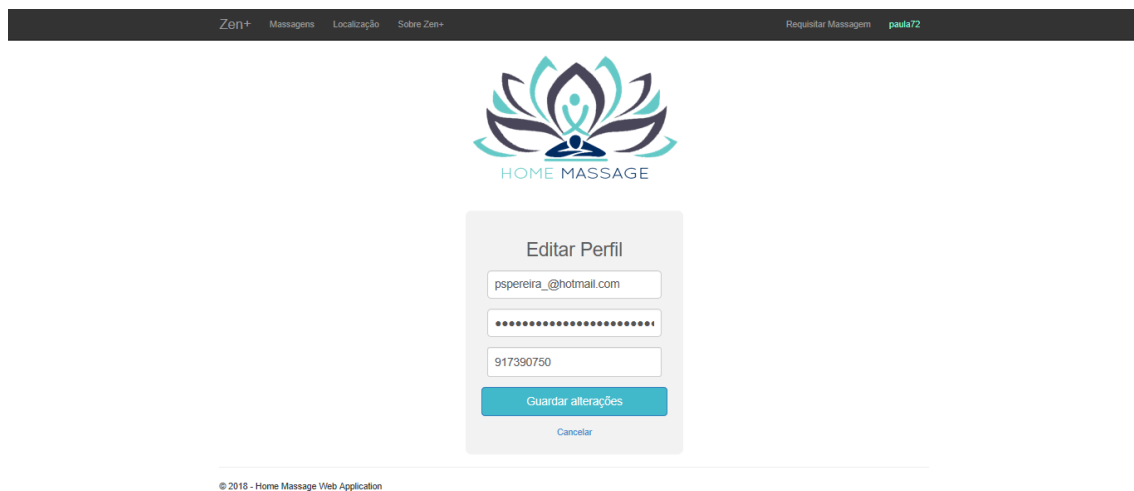


Figura 44 - Editar perfil de um cliente.

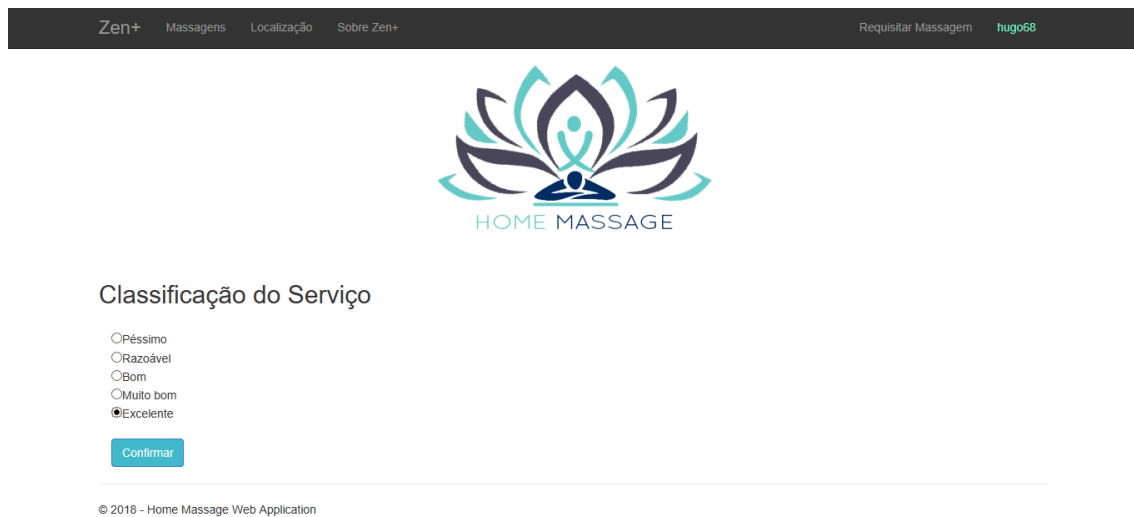


Figura 45 - Classificar um serviço.



### Serviços Pendentes/Efetuosos

Massagem	Data	Endereço	Estado	Classificação
Relaxante	01/07/2018 12:00:00	Rua Nova de Santa Cruz, n81	Pendente	<input type="button" value="✕ Cancelar"/>
Relaxante	01/06/2018 12:00:00	Rua Sao Domingos n186	Efetuada	Excelente

© 2018 - Home Massage Web Application

Figura 46 - Serviços Pendentes/Efetuosos de um cliente.

## 14. Conclusões e Trabalho Futuro

Na primeira fase deste projeto percebemos, desde logo, a importância de fundamentar e idealizar detalhadamente todos os aspetos do projeto. Esta fase permitiu que os vários membros da equipa de desenvolvimento entrassem em sintonia, o que se revelou extremamente importante não só na implementação do *software*, mas principalmente na sua especificação. Consideramos, portanto, que só assim fomos capazes de, trabalhando em aspetos distintos do projeto, contribuir positivamente para o produto final.

Um dos principais focos no decorrer desta etapa, centrou-se, sobretudo, no cuidado de estabelecer objetivos de trabalho, através da estipulação de datas para a sua conclusão. Acreditamos assim, que todo o planeamento desenvolvido com a ferramenta *Microsoft Project Professional 2016*, permitiu incutir um ritmo de trabalho eficiente.

Quando terminada a fundamentação do sistema, seguimos o calendário desenvolvido na fase anterior, iniciando assim a fase de especificação do sistema. Durante a elaboração desta segunda fase, sabíamos que esta seria das fases mais cruciais para o projeto, pois iria influenciar totalmente o nosso produto final.

Com este pensamento, a definição de requisitos foi a etapa mais demorada e na qual tivemos mais dificuldades, pois ambicionamos que todos os requisitos fossem de encontro ao que o grupo pretendia e que chegassem a um equilíbrio das ideias do mesmo.

No que diz respeito à última fase deste projeto, a fase de implementação, várias foram as dificuldades enfrentadas, sendo que o maior entrave à sua realização foi a má gestão do tempo, uma vez que, ainda que há semelhança das outras fases tivéssemos um calendário para nos ajudar no estabelecimento de metas para a finalização das várias tarefas, não foi fácil cumpri-las. Esse incumprimento deveu-se principalmente ao facto, como estudantes do 3º ano do curso de Engenharia Informática, de não termos sido capazes de conciliar o desenvolvimento deste projeto com todos os trabalhos, avaliações e aulas das restantes unidades curriculares. Isto refletiu-se na incapacidade de, ao contrário do que pretendíamos no início deste projeto, desenvolver todas as funcionalidades da forma pretendida.

Tendo tudo isto em conta, temos a perfeita noção de que ainda existe muito espaço para melhorias e que esta é uma oportunidade de adicionarmos complexidade ao sistema de *software* desenvolvido, pela inclusão de novas funcionalidades.

No entanto, consideramos que este foi, sem dúvida alguma, um dos projetos mais enriquecedores que já desenvolvemos ao longo do nosso percurso enquanto estudantes de engenharia informática, uma vez que, não só nos permitiu reforçar conhecimentos

anteriormente adquiridos, tais com linguagem UML e concepção e implementação de bases de dados relacionais, como adquirir novos conhecimentos relacionados com uso de ferramentas *Microsoft*.

## Referências Bibliográficas

M.Connolly, T. & E.Begg, C., 2015. Database Systems A Pratical Approach to Design, Implementation, and Management. 6th Edition ed. England: Person Education Limited.

Edmacauley, Data types (Transact-SQL). *Microsoft Docs*. Available at: <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017> [Accessed March 27, 2018].

SQL Server Functions. Available at: [https://www.w3schools.com/sql/sql\\_ref\\_sqlserver.asp](https://www.w3schools.com/sql/sql_ref_sqlserver.asp) [Accessed April 12, 2018].

"How to Send SMS Messages with ASP.NET MVC Framework." February 14, 2018. Available at: <https://www.nexmo.com/blog/2017/03/23/send-sms-messages-asp-net-mvc-framework-dr/> [Accessed May 18, 2018].

## Lista de Siglas e Acrónimos

<b>DSS</b>	Diagrama de sequência de subsistemas
<b>1FN</b>	Primeira Forma Normal
<b>2FN</b>	Segunda Forma Normal
<b>3FN</b>	Terceira Forma Normal



## **Anexos**

## I. Script de criação da Base de Dados

```
CREATE DATABASE homemassage;
```

```
USE homemassage;
```

```
CREATE TABLE Cliente (
```

```
    Id_Cliente          INT NOT NULL IDENTITY(1,1),
    Username            VARCHAR(50) NOT NULL UNIQUE,
    Password            VARCHAR(50) NOT NULL,
    Nome                VARCHAR(50) NOT NULL,
    Email               VARCHAR(50) NOT NULL UNIQUE,
    Contacto            INT NOT NULL,
    Numero_Contribuinte VARCHAR(9) NOT NULL,
    Data_Nascimento     DATE NOT NULL,
    Role                VARCHAR (60) DEFAULT 'user' NULL,
    CONSTRAINT PK_Client PRIMARY KEY (Id_Cliente)
```

```
);
```

```
CREATE TABLE Funcionario (
```

```
    Id_Funcionario      INT NOT NULL IDENTITY(1,1),
    Username            VARCHAR(50) NOT NULL UNIQUE,
    Password            VARCHAR(50) NOT NULL,
    Nome                VARCHAR(50) NOT NULL,
    Email               VARCHAR(50) NOT NULL UNIQUE,
    Estado              BIT NOT NULL,
    Role                VARCHAR (60) DEFAULT 'employee' NULL,
    CONSTRAINT PK_Funcionario PRIMARY KEY (Id_Funcionario)
```

```
);
```

```

CREATE TABLE Massagem (
    Id_Massagem          INT NOT NULL IDENTITY(1,1),
    Nome                 VARCHAR(15) NOT NULL UNIQUE,
    Preco                MONEY NOT NULL,
    Duracao              INT NOT NULL,
    Descricao            VARCHAR(250) NOT NULL,
    Imagem               IMAGE,
    CONSTRAINT PK_Massagem PRIMARY KEY (Id_Massagem)
);

CREATE TABLE Servico (
    Id_Servico           INT NOT NULL IDENTITY(1,1),
    Cliente              INT NOT NULL,
    Funcionario          INT NOT NULL,
    Massagem             INT NOT NULL,
    Data                DATETIME NOT NULL,
    Cartao_Credito       VARCHAR(19) NOT NULL,
    Estado               BIT NOT NULL,
    Endereco             VARCHAR(100) NOT NULL,
    Codigo_Postal        VARCHAR(8) NOT NULL,
    Ocorrencias          VARCHAR(250) NULL,
    Classificacao        INT NULL,
    CONSTRAINT PK_Servico PRIMARY KEY (Id_Servico),
    CONSTRAINT FK_Cliente FOREIGN KEY (Cliente)
    REFERENCES Cliente(Id_Cliente) ON UPDATE CASCADE,
    CONSTRAINT FK_Funcionario FOREIGN KEY (Funcionario)
    REFERENCES Funcionario(Id_Funcionario) ON UPDATE CASCADE,
    CONSTRAINT FK_Massagem FOREIGN KEY (Massagem)
    REFERENCES Massagem(Id_Massagem) ON UPDATE CASCADE
);

```

## II. Script de povoamento da Base de Dados

```
INSERT INTO Massagem (Nome,Preco,Duracao,Descricao,Imagem)
```

```
VALUES
```

```
('Relaxante',40,45,'',NULL),  
( 'Profunda',60,60,'',NULL),  
( 'Sueca',40,45,'',NULL),  
( 'Desportivo',70,60,'',NULL),  
( 'Pré-Natal',90,90,'',NULL);
```

```
INSERT INTO Funcionario (Username>Password,Nome>Email,Estado,Role)
```

```
VALUES
```

```
('admin','7260bedd7813b31b27c204450a065136','admin','homemassage@hotmail.com',1,'employee'),  
( 'func002','168cef729f6c0bbb7c2f3bec98857975','func002','funcionario2@hotmail.com',1,'employee'),  
( 'func003','b4fd4e310410cfa6c31661e66c425345','func003','funcionario3@hotmail.com',1,'employee'),  
( 'func004','2698a3f9752cb7a4a438e658a479a569','func004','funcionario4@hotmail.com',0,'employee'),  
( 'func005','f3ed85b558e1f3f1c72e68f52286de1a','func005','funcionario5@hotmail.com',1,'employee');
```

```
INSERT INTO Cliente  
(Username>Password,Nome>Email>Contacto>Numero_Contribuinte>Data_Nascimento,Role)
```

```
VALUES
```

```
('andre62','a4ad2a25dcf6ab4220606dce6982fd63','André Salgueiro','andre@gmail.com',926231675,'250704293','1996-07-09','user'),  
( 'hugo68','12cbab1ee557242c13b6396ef164701e','Hugo Oliveira','oliveirahugo.97@gmail.com',966765650,'250704293','1996-07-09','user'),
```

```

('paula72','f1370fb56962fddf6ff58d97dbaa4224','Paula
Pereira','pspereira@hotmail.com',917390750,'250704293','1996-07-
09','user'),
('rui133','b84e1c3b002c87fa8955ccc4c5e3defc','Rui
Calheno','ruic66@hotmail.com',915049712,'250704293','1997-06-
06','user');

```

```

INSERT                                INTO                                Servico
(Cliente,Funcionario,Massagem,Data,Cartao_Credito,Estado,Endereco,Codi
go_Postal)
VALUES
(2,2,1,'2018-07-01 12:00:00','401200103714111123',0,'Rua Nova de Santa
Cruz, n81','4710-409');

```