



**Universidade do Minho**  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## **Unidade Curricular de Laboratórios de Informática IV**

Ano Letivo de 2017/2018

### **GuguDadah**

**Carlos Campos A74745**

**Diana Costa A78985**

**Marcos Pereira A79116**

**Sérgio Oliveira A77730**

**Vítor Castro A77870**

Abril de 2018

Data de Recepção	
Responsável	
Avaliação	
Observações	

## GuguDadah

**Carlos Campos A74745**

**Diana Costa A78985**

**Marcos Pereira A79116**

**Sérgio Oliveira A77730**

**Vítor Castro A77870**

Abril de 2018

## Resumo

Este relatório descreve a primeira e segunda etapa do projeto “GuguDadah”, que surge da necessidade de implementar uma solução informática que permita gerir um serviço de babysitting ao domicílio, e respetivos babysittings.

A primeira fase consiste na fundamentação do projeto, onde o mesmo será contextualizado, feita a definição da identidade do sistema e explicitadas as motivações e respetivos objetivos. Nestes tópicos, será descrito o enquadramento do projeto e seu modelo de funcionamento, tal como os principais razões que levam à implementação e propósitos a alcançar no término do mesmo.

De seguida, é feita uma análise de viabilidade, dos recursos necessários e maquete do sistema. Assim, é possível estabelecer o custo aproximado do projeto, bem como descrever a forma como vão ser obtidos os dados para a aplicação e representar a arquitetura do sistema idealizado. Identificam-se também os principais constituintes do sistema e respetivas interações.

Fechando o primeiro capítulo, é obtido um conjunto de medidas de sucesso a cumprir, bem como o plano de desenvolvimento. Com estes, será possível avaliar o programa final e a eficácia da equipa nas diversas fases do desenvolvimento.

Já na segunda parte, foram definidos os requisitos de utilizador e de sistema, fundamentais ao prosseguimento do projeto, e, sobre estes, foram elaborados diagramas (modelação UML) para melhor orientar o grupo e definir metas e tarefas. Para sustentar e armazenar os dados dos clientes, funcionários e respetivos trabalhos realizados, construiu-se uma BD, onde se descreveu, etapa a etapa, a modelação conceptual e lógica e todo o raciocínio envolvente. Por fim, elaboraram-se os *mockups* do projeto, de maneira a projetar a interface do mesmo.

**Área de Aplicação:** Desenvolvimento de um sistema de babysitting ao domicílio

**Palavras-chave:** Desenvolvimento de Software, Engenharia de Software, aplicação, babysitting, infantário, localização, gestão de projeto, diagrama de Gantt, Base de Dados, Mockups, Diagramas, UML

# Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
1.1. Contextualização	1
1.2. Motivação e Objetivos	2
1.3. Definição da Identidade do Sistema a Desenvolver	3
1.4. Análise de Viabilidade	4
1.5. Identificação dos Recursos Necessários	6
1.6. Modelo de Sistema	7
1.7. Definição de Medidas de Sucesso	8
1.8. Plano de Desenvolvimento	9
2.1. Requisitos do utilizador e de sistema funcionais	11
2.2. Requisitos de sistema não-funcionais	17
2.2.1 Organizacionais	17
2.2.2 Externos	18
2.2.3 Produto	18
3.1. Diagrama de Use Cases	19
3.1.1 Especificação do Use Case “Aceitar proposta de trabalho”	21
3.1.2 Especificação do Use case “Avaliar trabalho”	22
3.1.3 Especificação do Use Case “Registo”	23
3.1.4 Especificação do Use Case “Agendar trabalho”	24
3.2. Diagramas de sequência	26
3.2.1 Exemplo de Diagrama de Sequência - Aceitar proposta de trabalho	26
3.2.2 Exemplo de Diagrama de Sequência - Avaliar trabalho	27
3.2.3 Exemplo de Diagrama de Sequência - Registo	29
3.2.4 Exemplo de Diagrama de Sequência – Agendar Trabalho	30
3.3. Diagramas de Máquinas de Estado	32
3.3.1 GuguDadah	32
3.3.2 Agendar Trabalho	33
3.3.3 Avaliar Trabalho	33

3.3.4 Registo de Clientes	34
3.3.5 Registo de Profissionais	34
3.3.6 Gerir pagamentos	35
3.3.7 Marcar Clientes Golden	35
3.3.8 Verificar Propostas de Trabalho	36
3.3.9 Verificar Trabalhos Pendentes	36
3.4. Diagrama de atividades	37
3.5. Diagrama de Classes	39
4.1. Modelo Conceptual	41
4.1.1 Identificar os tipos de identidades	41
4.1.2 Identificar tipos de relacionamento	41
4.1.3 Identificar e associar atributos com os tipos de entidades e relacionamentos	42
4.1.4 Atributos simples/compostos	42
4.1.5 Atributos derivados	42
4.1.6 Atributos multivalor	42
4.1.7 Associação entre atributos e entidades	43
4.1.8 Determinar domínio dos atributos	45
4.1.9 Determinar chaves primárias, candidatas e alternativas	46
4.1.10 Desenho do diagrama ER	47
4.1.11 Revisão do modelo de dados com o utilizador	48
4.2. Modelo Lógico	49
4.2.1 Construção e validação do modelo de dados lógico	49
4.2.1.1 Entidades fortes	49
4.2.1.2 Relacionamentos binários 1:N	51
4.2.2 Desenho do modelo lógico	53
4.2.3 Validação do modelo através da normalização	53
4.3 Análise do crescimento futuro	54
4.4 Revisão do modelo lógico com o utilizador	54

## Índice de Figuras

Figura 1 – Maquete do Sistema	7
Figura 2 - Diagrama de Gantt: Fundamentação	9
Figura 3 - Diagrama de Gantt: Especificação	9
Figura 4 - Diagrama de Gantt: Implementação	9
Figura 5 - Especificação do Use Case "Aceitar proposta de trabalho"	21
Figura 6 - Especificação do Use Case "Avaliar trabalho"	22
Figura 7 - Especificação do Use Case "Registo"	23
Figura 8 - Especificação do Use Case "Agendar trabalho"	25
Figura 9 - Diagrama de Sequência "Aceitar Proposta de Trabalho"	27
Figura 10 - Diagrama de Sequência "Avaliar Trabalho"	28
Figura 11 - Diagrama de Sequência "Registo"	29
Figura 12 - Diagrama de Sequência "Agendar Trabalho"	31
Figura 13 - Diagrama geral de máquinas de estado, com as respectivas sub-máquinas	32
Figura 14 - Diagrama de máquinas de estado - Agendar Trabalho	33
Figura 15 - Diagrama de máquinas de estado - Avaliar trabalho	33
Figura 16 - Diagrama de máquinas de estado - Registo de clientes	34
Figura 17 - Diagrama de máquinas de estado - Registo de profissionais	34
Figura 18 - Diagrama de máquinas de estado - Gerir pagamentos	35
Figura 19 - Diagrama de máquinas de estado - Marcar Clientes Golden	35
Figura 20 - Diagrama de máquinas de estado – Verificar Propostas de Trabalho	36
Figura 21 - Diagrama de máquinas de estado - Verificar Trabalhos Pendentes	36
Figura 22 - Diagrama de atividades	38
Figura 23 - Diagrama de classes	39
Figura 24 - Diagrama de classes - DAOs	40
Figura 25 - Desenho do Diagrama ER	47
Figura 26 - Concetual para lógico (Relacionamento Trabalho Funcionario)	52
Figura 27 - Homepage	56
Figura 28 - Homepage com sessão iniciada	57
Figura 29 – Login	58

Figura 30 – Registo	59
Figura 31 - Cliente - minha área	60
Figura 32 - Cliente – Babysitters	61
Figura 33 - Cliente - Agendar passo 1	62
Figura 34 - Cliente - Agendar passo 2	63
Figura 35 - - Cliente - Agendar passo 3	64
Figura 36 - Cliente - Agendar passo 4	65
Figura 37 - Profissional - Minha área	66
Figura 38 - Profissional - Ver localização	67
Figura 39 - Admin - Registrar profissional	68
Figura 40 - Admin - utilizadores do Sistema	69
Figura 41 - Admin - Lista geral de trabalhos	70

# **1. Definição do Sistema**

## **1.1. Contextualização**

Em pleno século XXI, com o desenvolvimento industrial e tecnológico, cresce a necessidade de produção contínua e rápida por parte das empresas, para singrar nos mercados económicos. Com isso aumenta a carga horária dos trabalhadores, desde o nível mais laboral, até ao executivo. Tudo isto gera um problema sério para quem tem filhos, visto que cria um défice de tempo livre e, por vezes, com reuniões até tarde ou a condição de trabalhar horas extra, torna-se difícil arranjar alguém para tomar conta dos filhos. Por vários motivos, ora os avós moram longe, ora o casal amigo tem um jantar importante, e como os infantários não trabalham até horas tardias, é causado o pânico.

A empresa Bebés & Companhia, que possui uma cadeia de infantários espalhados pela grande Lisboa, tendo em vista essa adversidade, com o reparo que a maioria dos seus funcionários estão na faixa etária dos vinte, viram aqui uma ótima oportunidade. Surge, assim, a ideia de fazer um serviço ao domicílio de babysitting.

Depois da realização de um inquérito por todos os jardins de infância, em que foram abordadas várias questões, como a disposição para trabalhar horas extra, disponibilidade para horas tardias e para deslocações ao domicílio, foi concluído que a adesão foi de acordo com as expectativas a nível de idades e com isso ficaria acessível um grande volume de mão-de-obra, por parte da própria empresa, não havendo a intenção de contratar agentes externos.

Quando a equipa de desenvolvimento de software foi contactada, após algumas reuniões e um estudo do mercado, ficou acordado que a ideia seria patenteada e que a mesma seria viável e suscetível a grande escalabilidade. Seria assim criada a plataforma “GuguDadah”.



## **1.2. Motivação e Objetivos**

O principal motivo que levou o grupo à realização deste projeto, é fundamentalmente a dificuldade de encontrar alguém que seja responsável para tomar conta dos nossos filhos a horas tardias, ou a falta de disponibilidade dos infantários para cuidar de crianças adoentadas. Seja por causa de uma reunião, ou também por causa de um jantar muito importante, por vezes não existe auxílio por parte de algum ente querido, e como não existe por norma infantários abertos em horário noturno, o problema é agravado. A política atual dos infantários também inclui que crianças doentes, para não afetar as restantes, deverão ficar em casa.


Outro fundamento para esta motivação, é a inexistência de um sistema que permita a requisição de serviços de babysitting online, em Portugal. O mais próximo que poderemos encontrar são os anúncios no OLX, e lembrando que este é essencialmente um site de venda de produtos em segunda mão. Assim sendo, não existe um controlo da atividade prestada, nem qualquer tipo de garantia ou de qualidade por parte da pessoa que presta a função exigida.

Uma condição que suporta a iniciativa da equipa é que a maioria dos trabalhadores dos Infantários Bebés & Companhia possuem idade no intervalo dos vinte aos trinta. Como faz parte do senso comum, cidadãos nestas idades têm tendência para querer angariar mais fundos monetários, visto que estão em inícios de vidas autónomas, e como a maioria também não possuiu crianças, faz com que tenham uma maior disponibilidade e vontade, para se prontificarem a trabalhar a horas tardias, e assim são um grande suporte a esta ideia de negócio.

Com uma ideia original, que é apoiada por estes vários fatores, foi tomada a decisão de fazer um estudo do mercado para consolidar a ambição do grupo, que contactou a “MarkUp”, uma empresa especializada em fazer esse tipo de análise. Depois de realizada a devida observação do relatório final, concluiu-se que este projeto iria obter sucesso, seguindo várias condicionantes que foram referidas anteriormente.

### 1.3. Definição da Identidade do Sistema a Desenvolver

O projeto a desenvolver pela equipa, denominado "GuguDadah", é um sistema de gestão de serviços de babysitting personalizados, oferecido pelo infantário "Bebés & Companhia", que decidiu expandir o seu serviço diurno e localizado para um que funcionasse non-stop e ao domicílio. Os pais das crianças, quer inscritas no infantário, quer externas, poderão requisitar um educador profissional e certificado para se deslocar a sua casa, em qualquer horário, conforme o tipo de préstimo que optaram. Os profissionais serão avaliados pelos pais conforme os seus serviços anteriores, e os pais obterão descontos de acordo com a frequência com que requerem préstimos ou caso tenham filho(s) inscritos no "Bebés & Companhia". Por último, é de salientar que um educador tem sempre acesso à navegação por GPS sempre que estiver a executar a rota até ao domicílio combinado. Resumem-se de seguida algumas características identificativas do sistema a desenvolver, com um objetivo meramente elucidativo, uma vez que as mesmas serão desenvolvidas à frente no relatório.

Nome:	GuguDadah
Slogan:	De Pais, Para pais
Categoria:	Babysitting
Características:	Implementação de descontos em serviços
	Reserva/Requerimento de profissionais para serviço
	Avaliação dos profissionais
	Navegação GPS
Faixa Etária:	20-50
Logótipo:	

## 1.4. Análise de Viabilidade

De maneira a analisar a viabilidade deste novo serviço, coube ao grupo cumprir o papel de adicionar, sobre a análise de mercado inicial da empresa contratante, uma outra análise ou estudo da adequação deste empreendimento às circunstâncias do serviço ao domicílio.

Um sistema de entregas ao domicílio 24/365 é um modelo de negócio cada vez mais comum no tempo presente, que demonstra uma abertura do cliente à adaptação contínua e adoção de novas formas de comércio. A popularidade deste modelo indica, à partida, a viabilidade da solução. Foi realizada, assim mesmo, uma pesquisa breve relativa à presença destes tipos de negócio no mercado atual.

Constatou-se rapidamente que a adoção das entregas ao domicílio é cada vez mais geral, havendo anúncios da abertura destes serviços por parte de grandes marcas com uma frequência elevada, como a *Amazon*, que melhora continuamente as suas opções de entrega. Seguem-se alguns outros exemplos:

- *"The owner of KFC and Taco Bell is teaming up with Grubhub to expand home delivery [...]"* (8 Fev, 2018)
- *"McDonald's cheers home delivery success as UK spurs sales rise"* (Jan 30, 2018)
- *"UberEats Starts Delivering Food in Detroit Today"* (Nov 30, 2017)
- *"Burger King já faz entregas ao domicílio em Lisboa"* (Jan 6, 2017)

De uma forma mais geral, o termo de pesquisa "home delivery" tem visto um crescimento estável já há vários anos, sendo por isso um alvo seguro para um negócio que já esteja estabelecido e queira aumentar as suas chances de se manter relevante numa sociedade que está continuamente em desenvolvimento.

Foi demonstrado acima, de maneira resumida, que o modelo de negócio projetado é popular. No entanto, um leitor atento pode já ter reparado que o tipo negócio que mais participa nesta adoção tende a estar relacionado com a restauração.

A criação de um serviço de babysitting ao domicílio posiciona-se, então, como uma opção de entrada num mercado estável e de crescimento contínuo, mas ainda assim inovadora o suficiente para se manter um movimento original, mais do que um investimento trivial.

Foram feitos estudos de mercado com recurso a inquéritos por telefone, que demonstraram uma elevada recetividade à possibilidade de agendar um(a) baby-sitter a qualquer altura do dia, durante todo o ano. O facto de o serviço estar associado a um lugar físico onde se podem conhecer os profissionais, aumenta também a confiança por parte dos pais.

Foi também descoberto que a perda de confiança comum ao movimento inicial de um serviço para um meio digital é compensada pela presença de avaliações pessoais e rankings no sistema, que aumentam o nível de transparência face ao consumidor.

Uma posterior avaliação da proposta por parte de um comité de gestores concluiu que o orçamento estipulado é adequado ao nível de complexidade do projeto. Havendo a possibilidade de manutenção a longo termo, e tendo em conta as projeções positivas da receção do novo modelo de negócio, este empreendimento é dado como uma mais valia para a organização à qual o grupo pertence.

## **1.5. Identificação dos Recursos Necessários**

Para este projeto estima-se a necessidade de alocação de um engenheiro de software e quatro programadores. O engenheiro de software irá assumir também o papel extraordinário de gestor de projeto e de gestor de bases de dados, para além de ser responsável pela especificação do projeto. Os programadores irão trabalhar 2h/dia, daí a necessidade de quatro profissionais envolvidos no projeto, que dá o equivalente a 1homem/mês.

Não serão utilizados quaisquer equipamentos para a realização deste projeto, para além dos computadores disponibilizados pela software house associada aos engenheiros e programadores.

Numa fase inicial serão feitas duas reuniões com CEO do infantário Bebés & Companhia. Na primeira vai-se proceder ao levantamento de requisitos e, no fim do período de planeamento do projeto, a confirmação do modelo elaborado. Ambas as reuniões desta fase serão feitas nas instalações da empresa que pretende contratar o serviço, exigindo um veículo ligeiro para deslocação da equipa em ambas as ocasiões. No intervalo das reuniões será também necessário fazer um estudo de mercado, encomendado a uma empresa externa.

A equipa será alocada ao projeto durante três meses, com algumas interrupções para participação em outros projetos. Por este motivo, é natural que surjam períodos de pausa no desenvolvimento.

É também necessário alocar um servidor que estará na software house e que será o responsável pelo alojamento dos serviços 24/7 assim como se prevê a inevitabilidade de se adquirir tablets para a equipa realizar testes nas diferentes plataformas existentes no mercado.

Os diferentes softwares e suas licenças estarão, à partida, garantidos pelos trabalhadores da empresa pelo que não acarretam nenhum encargo financeiro para a empresa.

Por fim, o projeto será desenvolvido inteiramente nos escritórios da empresa, levando à ocupação dos espaços habituais dos colaboradores.

## 1.6. Modelo de Sistema

O sistema consiste num servidor web ASP.NET ligado a uma base de dados SQL Server, onde serão guardados todos os dados dos intervenientes no sistema. O servidor expõe uma API pública que é disponibilizada através do site, assim como uma API privada que é utilizada pela aplicação móvel.

O site web é usado por clientes de maneira a escolher baby-sitters, agendar trabalhos, e submeter avaliações. A aplicação móvel é usada pelos trabalhadores e permite a verificação do perfil profissional e dos trabalhos agendados, indicando também o caminho até estes (deslocação GPS até ao domicílio pretendido).

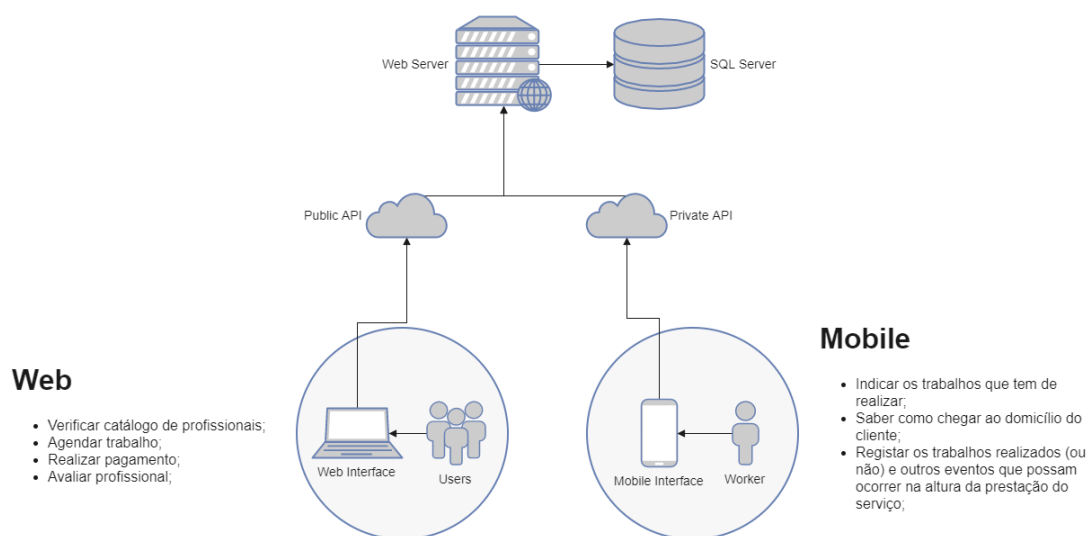


Figura 1 – Maquete do Sistema

## **1.7. Definição de Medidas de Sucesso**

Para proceder a uma avaliação de qualidade do projeto definiram-se, desde logo, um conjunto de pontos a verificar. Assim, segue a lista com os tópicos a cumprir para uma boa execução do plano traçado:

- A aplicação deve ser classificada pelos utilizadores como de fácil utilização;
- Devem ser atingidos 1000 usos da aplicação no período de 365 dias;
- O nível de satisfação dos utilizadores do infantário, avaliado através dos inquéritos anuais, deve subir, pelo menos, 3 pontos percentuais;
- O número de utilizadores do infantário deve subir 5 pontos percentuais, um ano após a implementação.

## 1.8. Plano de Desenvolvimento

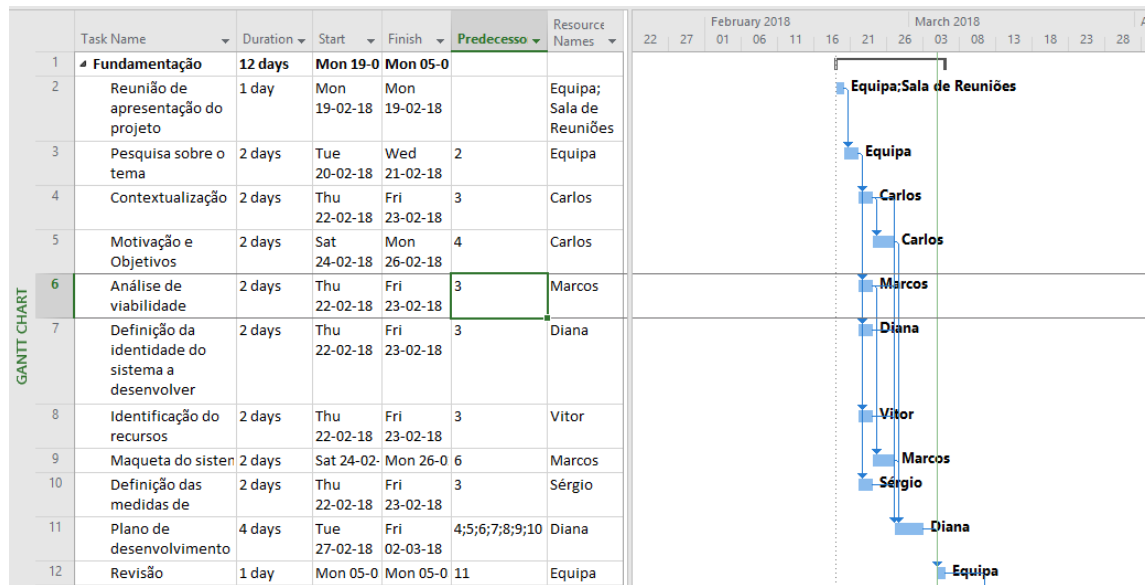


Figura 2 - Diagrama de Gantt: Fundamentação

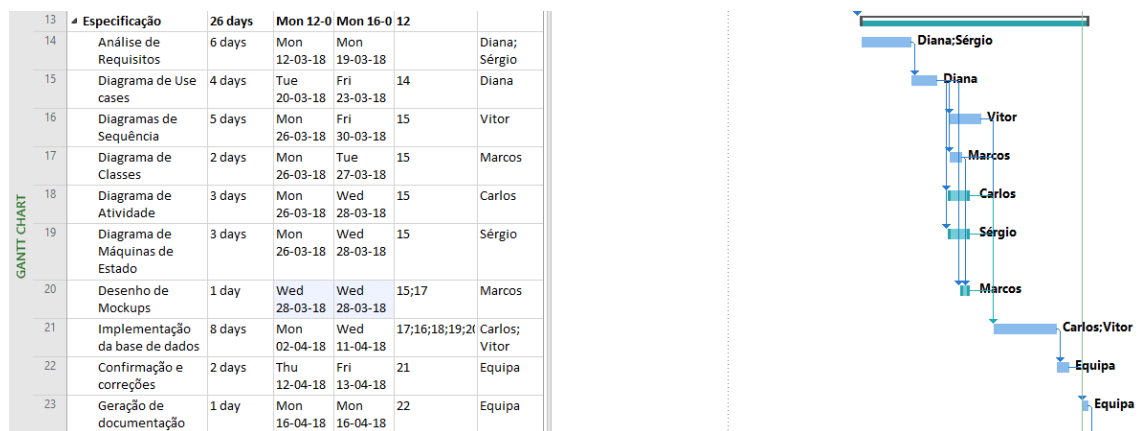


Figura 3 - Diagrama de Gantt: Especificação

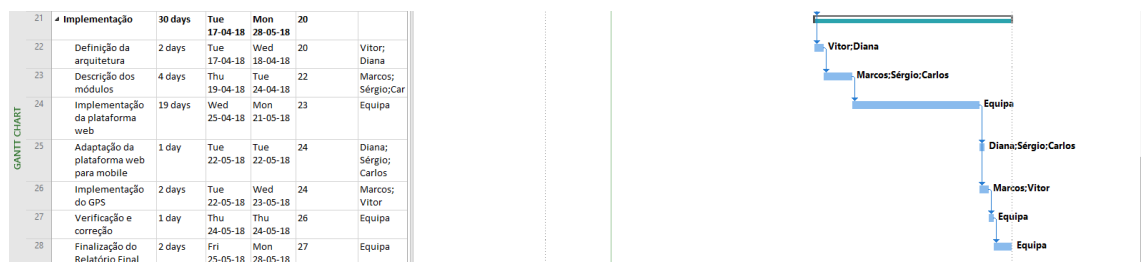


Figura 4 - Diagrama de Gantt: Implementação



Na primeira fase foi feita a fundamentação do tema. Após feita uma intensiva pesquisa inicial, com vista à contextualização correta com o problema, foi possível fazer a divisão das diversas tarefas de acordo com os pontos fortes dos elementos. Assim, apenas a fase de pesquisa sobre o tema e de revisão ficaram como trabalho de equipa.

O tempo dado a cada um dos contribuidores, apesar de por vezes poder ser maior (não tendo implicações no prazo de entrega da primeira parte), vai de acordo com as estimativas do gestor de projeto.

Nas fases seguintes haverá a necessidade de ter toda a equipa reunida em torno de uma tarefa ou outra tarefa, dada a importância da mesma. Todas as outras serão distribuídas segundo o mesmo critério que na primeira fase.

## 2. Requisitos

Por forma a que a *software house* começasse a definir uma concordância com o cliente (infantário Bebés&Companhia), entre a própria equipa de programadores, e para restringir, priorizar e padronizar o que o sistema deveria fazer e fornecer, foram definidos um conjunto de requisitos. Estes requisitos, fulcrais ao projeto para descrever o funcionamento do sistema, obrigaram a várias reuniões entre a equipa e o infantário, e estudos analíticos na área de serviços de acompanhamento e educação de bebés e crianças.

Nas próximas secções apresentar-se-ão os requisitos de utilizador, bem como os respetivos de sistema, que se dividem, por sua vez, em funcionais e não funcionais.

### 2.1. Requisitos do utilizador e de sistema funcionais

Os requisitos devem ter diferentes níveis de detalhe, uma vez que têm, também, diferentes tipos de leitores, que os usam de maneiras diferentes. Assim, para cada requisito de utilizador, apresentam-se os de sistema que lhe correspondem, em formato tabular, de forma a facilitar a leitura. Como maneira a clarificar o raciocínio, dividem-se, também, os requisitos em temas – requisitos correspondentes ao cliente, profissional e administrador.

- **Cliente**

- **Verificar lista de profissionais**

Definição do requisito de Utilizador
1. O utilizador deve poder consultar a lista de profissionais existentes no sistema.
Especificação do(s) requisito(s) de Sistema
1.1. O sistema mostrará a lista dos profissionais registados no sistema, bem como os seus rankings e horário de trabalho (turno).
1.2. A lista de profissionais deverá ser apresentada ordenada por ranking, anos de trabalho ou número de serviços prestados.
1.3. Por cada profissional, deverão ser fornecidos alguns dados pessoais.

○ **Agendar trabalho**

Definição do requisito de Utilizador
2. O utilizador deve poder agendar um trabalho, com base na data, hora e morada requerida, e com base de profissionais disponíveis para a hora mencionada. O cliente terá também à sua disposição um conjunto de serviços extra que poderá, ou não, optar por agendar, acrescentando ao preço total do trabalho.
Especificação do(s) requisito(s) de Sistema
2.1. O sistema deve permitir seleccionar as datas em que o serviço irá ser realizado, assim como a hora desejada para o mesmo ser realizado.
2.2. O sistema deve recorrer à API externa do Bing Maps para conhecer a localização do utilizador ou então obriga a que a localização seja introduzida manualmente.
2.3. Deverá autorizar a inserção de serviços complementares por parte do utilizador, para especificar algumas exigências do mesmo no que toca ao serviço.
2.4. O sistema deve permitir ao utilizador seleccionar o profissional que pretende.
2.5. Os profissionais estão agrupados de acordo com o horário/turno em que trabalham: Manhã (8h-16h), Tarde (16h-24h) ou Noite (24h-8h).
2.6. O sistema deve permitir ao utilizador verificar o custo do agendamento, antes de finalizar a operação, que será debitado depois do profissional aceitar.
2.7. O sistema deverá permitir que seja mandada uma proposta de serviço devidamente especificada ao profissional, que, posteriormente, poderá aceitar ou recusar.

○ **Avaliar trabalho**

Definição do requisito de Utilizador
3. O utilizador deve poder avaliar um trabalho, já realizado, de um profissional.
Especificação do(s) requisito(s) de Sistema
3.1. O sistema deve mostrar os parâmetros classificativos essenciais à avaliação e deve permitir que o utilizador os consiga preencher.
3.2. A classificação é um decimal positivo, desde 0-5★.
3.3. Armazenar a média dos parâmetros no sistema para, posteriormente, outros clientes poderem consultar o ranking.

- **Realizar pagamento**

Definição do requisito de Utilizador
4. O utilizador deve poder realizar o pagamento do serviço (em dinheiro ou online/cartão), antes da realização do mesmo, e depois de confirmado o agendamento).
Especificação do(s) requisito(s) de Sistema
4.1. O sistema deverá permitir que seja elaborado um método de pagamento (online), que será sempre a favor do infantiário (que acordará depois o pagamento com os respetivos profissionais). Este método é utilizado aquando do agendamento do trabalho.

- **Beneficiar de descontos – ser golden ou externo**

Definição do requisito de Utilizador
5. O utilizador deve poder beneficiar de descontos, conforme o seu estatuto de cliente golden (com filho(a) inscrito(a) no infantiário) ou externo.
Especificação do(s) requisito(s) de Sistema
5.1. O sistema deverá permitir marcar os clientes Golden com o referido estatuto, para futuros descontos em agendamentos.

- **Registo no sistema**

Definição do requisito de Utilizador
1. O utilizador deve poder registar-se no sistema.
Especificação do(s) requisito(s) de Sistema
1.1 O sistema deve solicitar email, username, nome, password e contacto.
1.2 O sistema não deve permitir o registo de utentes com um username/email já registado.
1.3 O sistema deve armazenar os dados.

- **Profissional**

- **Ver direções até à morada do cliente**

Definição do requisito de Utilizador
1. O utilizador deve poder ver direções até à casa do cliente. Deve também ser apresentado um mapa com representativo da morada. Para além disso, deve poder visualizar a trajetória até à residência do cliente.
Especificação do(s) requisito(s) de Sistema
1.1. O sistema deve recorrer-se da API externa disponibilizada pelo Bing Maps de forma a representar a localização do cliente, bem como a trajetória até este.

- **Cliente e Profissional**

- **Verificar trabalhos anteriores**

Definição do requisito de Utilizador
1. O utilizador deve conseguir aceder a informação detalhada de serviços já realizados anteriormente.
Especificação do(s) requisito(s) de Sistema
1.1. O sistema deverá permitir observar uma listagem de todos os trabalhos realizados anteriormente à data de um cliente específico.

- **Verificar trabalhos agendados**

Definição do requisito de Utilizador
2. O utilizador poderá analisar os trabalhos agendados, com o objetivo de verificar os detalhes de agendamento.
Especificação do(s) requisito(s) de Sistema
2.1. O sistema deverá permitir a disponibilização de uma lista de serviços de babysitting agendados, com os respetivos detalhes.

- **Verificar trabalhos propostos**

Definição do requisito de Utilizador
3. O utilizador deve poder analisar os trabalhos propostos (ainda não aceites), com o intuito de verificar os detalhes de agendamento (hora, data, ...), ou até cancelar o mesmo.
Especificação do(s) requisito(s) de Sistema
3.1. O sistema deverá permitir a disponibilização de uma lista de serviços de babysitting propostos, juntamente com os respetivos detalhes, e permitir a opção de cancelamento do mesmo.

- **Verificar perfil**

Definição do requisito de Utilizador
4. O utilizador deve poder verificar o seu perfil, que deverá conter a informação pessoal inserida aquando do registo, assim como as listagens de trabalhos propostos, agendados e já realizados.
Especificação do(s) requisito(s) de Sistema
4.1 O sistema deverá fornecer o perfil de utilizador completo e intuitivo.

- **Administrador**

- **Marcar clientes golden**

Definição do requisito de Utilizador
1. O utilizador deverá poder marcar, dentro da lista dos clientes, os que são golden ou externos.
Especificação do(s) requisito(s) de Sistema
1.1 O sistema disponibilizará a opção de poder marcar clientes Golden.

- **Gerir pagamentos**

Definição do requisito de Utilizador	
2.	O utilizador deverá poder gerir a lista de trabalhos já efetuados, com o propósito de consultar as despesas e, posteriormente, contactar o infantário.
Especificação do(s) requisito(s) de Sistema	
2.1	O sistema disponibilizará a lista de serviços já aceites, juntamente com os detalhes do mesmo.

- **Verificar lista de utilizadores do sistema**

Definição do requisito de Utilizador	
3.	O utilizador deverá poder ver e gerir a lista de utilizadores do sistema.
Especificação do(s) requisito(s) de Sistema	
3.1	O sistema disponibilizará a lista de utilizadores, juntamente com detalhes dos mesmos.

- **Registar profissional**

Definição do requisito de Utilizador	
1.	O utilizador deve poder registar profissionais no sistema.
Especificação do(s) requisito(s) de Sistema	
1.1	O sistema deve solicitar email, username, nome, password, contacto e turno.
1.2	O sistema não deve permitir o registo de utentes com um username/email já registado.
1.3	O sistema deve armazenar os dados.

- **Verificar lista geral de trabalhos**

Definição do requisito de Utilizador
4. O utilizador deverá poder verificar e gerir a lista de trabalhos geral - já efetuados, agendados e propostos.
Especificação do(s) requisito(s) de Sistema
4.1 O sistema disponibilizará a lista de serviços, juntamente com detalhes dos mesmos.

- **Cliente, Profissional e Administrador**

- **Autenticação no sistema**

Definição do requisito de Utilizador
2. O utilizador deve conseguir autenticar-se no sistema.
Especificação do(s) requisito(s) de Sistema
2.1 O sistema deve solicitar email/username e password para se autenticar;
2.2 O sistema deve verificar a validade dos dados inseridos e associar os dados a um perfil de utilizador, de modo a reconhecê-lo.

## 2.2. Requisitos de sistema não-funcionais

Em relação à parte técnica e às características internas do programa, enunciam-se os RNF que se subdividem em organizacionais, externos e de produto.

### 2.2.1 Organizacionais

- O utilizador deve autenticar-se com o seu username e a password respetiva;
- A linguagem de programação a utilizar será C# em cooperação com a framework ASP.NET;
- O SGBD a usar deve ser o SQL Server;
- A aplicação será desenvolvida para web;



### **2.2.2 Externos**

- O sistema deve recorrer a uma API de mapas para representar a localização do trabalho ou até mesmo para representar a sua própria localização;
- O sistema não deve apresentar aos utilizadores dados privados acerca dos restantes utilizadores, sendo que, cada utilizador só pode ver os seus dados e aqueles que são públicos;

### **2.2.3 Produto**

- O sistema deve estar disponível 24h por dia. No pior dos casos, espera-se uma disponibilidade média superior a 99%;
- O sistema deve ser de fácil uso;
- O sistema deve ser produzido de modo a ser executado em todos os browsers;
- O sistema deve suportar o registo de 1000 utilizadores, no espaço de 1 ano;
- O nível de satisfação dos utilizadores com a aplicação deverá ser elevado.

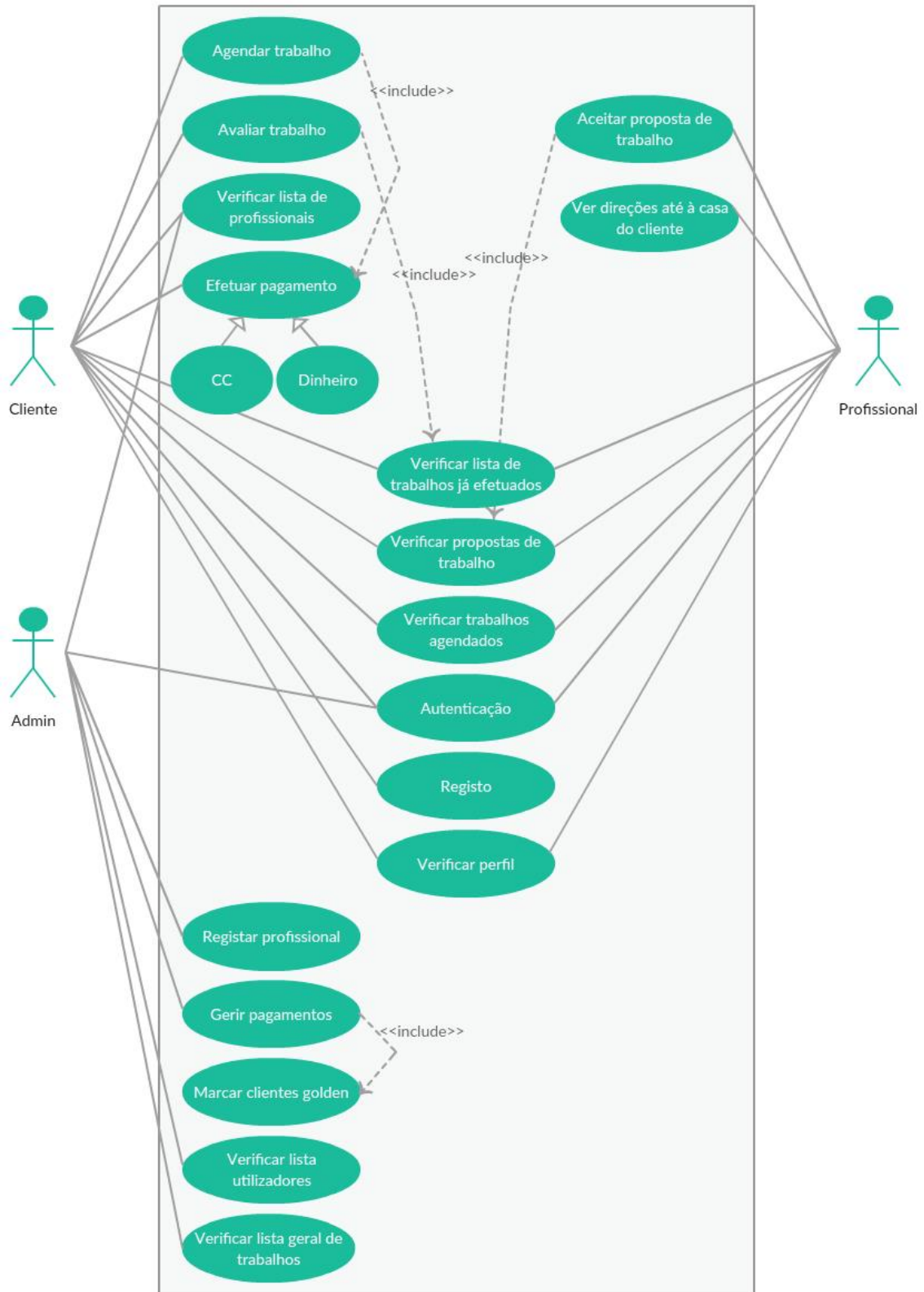
### 3. Modelação UML

A UML - “Unified Modeling Language” - é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Para o caso, esta é empregada na visualização, especificação, construção e documentação de artefatos que façam uso de sistemas complexos de software. O grupo escolheu elaborar, como se pode observar ao longo desta secção, um diagrama de use cases, juntamente com as suas especificações, diagrama de sequência de sistema, de atividade, máquinas de estado e, por fim, de classes, por achar que são diagramas de alto nível e de fácil entendimento por parte de um cliente.

#### 3.1. Diagrama de Use Cases

Uma vez validados os requisitos anteriores, estes fornecerão a base para a criação do diagrama de *use cases*, que servirá como auxílio à comunicação entre o cliente e o grupo. Assim, de maneira a delimitar bem as fronteiras e funcionalidades do sistema do ponto de vista do cliente, e para saber, posteriormente, o que teria de ser implementado, foram identificados os atores, *use cases* e associações do presente sistema de babysitting. Pode ver-se, de seguida, a existência de três atores – Profissional, Cliente e Administrador, que exercem tarefas e têm funções distintas.

# Babysitting System



### 3.1.1 Especificação do Use Case “Aceitar proposta de trabalho”

Na imagem seguinte é possível observar, em formato tabular, a especificação do *use case* “Aceitar proposta de trabalho”, em que o ator é um profissional do infantário. Este profissional, após receber uma proposta de trabalho de babysitting, deverá aceitá-la ou não, por forma a deixar saber um cliente se está disponível para a realização de um serviço ou não.

Name	Aceitar proposta de trabalho		
Brief Description	Permite a um profissional aceitar a realização de um futuro serviço		
Preconditions	Profissional autenticado pretende aceitar um trabalho		
Post-conditions	Trabalho aceite		
Flow of Events	Actor Input		System Response
	1	<<include>> Verificar propostas de trabalho	
	2	Selecionar proposta de trabalho para efetuar	
	3		Verifica estado da proposta
	4	Aceita proposta de trabalho	
	5		Regista confirmação de proposta
	6		Indica que trabalho foi aceite e que poderá ser consultado na lista de trabalhos agendados
Exceção 1 [trabalho indisponível] (passo 3)	1		Informa cliente que a proposta de trabalho não está disponível ou foi cancelada
Alternativa 1 [não aceita proposta] (passo 4)	1		Informa que proposta não será aceite e que o cliente será notificado

Figura 5 - Especificação do Use Case "Aceitar proposta de trabalho"

### 3.1.2 Especificação do Use case “Avaliar trabalho”

Apresenta-se, de seguida, a especificação tabular do *use case* “Avaliar trabalho”, efetuado por um cliente. Após um trabalho ter sido realizado, um cliente pode manifestar o seu desagrado ou satisfação através de uma avaliação numérica decimal (0-5★) a um profissional, de modo a constituir um parâmetro de escolha e dar mais confiança a próximos clientes.

Name	Avaliar trabalho		
Brief Description	Permite a um cliente avaliar um serviço prestado anteriormente		
Preconditions	Cliente autenticado pretende avaliar um trabalho		
Post-conditions	Trabalho avaliado		
Flow of Events	Actor Input		System Response
	1	<<include>> Verificar lista de trabalhos já efetuados	
	2	Selecionar trabalho para avaliar	
	3	Inserir avaliação numérica (0-5)	
	4		Verifica avaliação
	5		Regista avaliação
	6		Indica que avaliação foi registada com sucesso
Exceção 1 [avaliação já efetuada] (passo 4)	1		Informa cliente que já avaliou o serviço anteriormente

Figura 6 - Especificação do Use Case "Avaliar trabalho"

### 3.1.3 Especificação do Use Case “Registo”

Observa-se, em baixo, a especificação do *use case* “Registo”, em formato tabular. Um registo só poderá ser efetuado por um cliente, uma vez que um profissional é registado pelo administrador, e o último não precisa de registo. Desta forma, um cliente deverá fornecer ao sistema um email, username para ser identificado, nome, password e contacto telefónico, de maneira a ser contactado, por exemplo, por um profissional.

<b>Name</b>	Registo	
<b>Brief Description</b>	Permite a um profissional/admin registarem-se no sistema	
<b>Preconditions</b>	Ator pretende registar-se no sistema	
<b>Post-conditions</b>	Ator registado	
<b>Flow of Events</b>	<b>Actor Input</b>	
	1	Inserir credenciais: email, username, nome, password e contacto
	2	Confirma se foram fornecidos todos os dados
	3	Confirma se utilizador já existe
	4	Regista utilizador
	5	Indica sucesso na criação de utilizador
<b>Exceção 1</b> [sem dados] (passo 2)	1	Informa ator que não foram fornecidos dados suficientes
<b>Exceção 2</b> [utilizador já existente] (passo 3)	1	Informa que o utilizador em questão já existe

Figura 7 - Especificação do Use Case "Registo"

### 3.1.4 Especificação do Use Case “Agendar trabalho”

Na imagem seguinte é possível observar a especificação do use case “Agendar trabalho”, à qual um cliente que pretenda agendar um serviço terá de obedecer. Num primeiro instante, terá de fornecer a data e a hora desejadas para que o babysitter se desloque à morada, também fornecida. Poderá também optar por seleccionar serviços extra, como acompanhamento ao estudo ou atividades lúdicas no exterior, e por adicionar informação nas observações, por forma a melhor informar o profissional sobre o serviço que o cliente requer. Numa segunda fase, o ator deverá escolher um profissional, podendo-se basear no ranking, e, por fim, será lhe apresentado o montante e forma desejada de pagamento. Com estes passos, um cliente e um profissional ficam com um serviço agendado, que poderá ser aceite ou não pelo segundo.

<b>Name</b>	Agendar Trabalho	
<b>Brief Description</b>	Permite a um cliente agendar um novo trabalho/serviço	
<b>Preconditions</b>	Ator autenticado pretende agendar um trabalho	
<b>Post-conditions</b>	Ator tem trabalho agendado	
<b>Flow of Events</b>	<b>Actor Input</b>	
	1	Seleciona data desejada para o agendamento
	2	Insere rua e localidade
	3	Seleciona hora de início e de fim do serviço pretendido
	4	Seleciona serviços complementares
	5	Preenche campo de observações
	6	
	7	
	8	Escolhe um profissional
	9	
	10	
	11	
	12	Analisa valor calculado do serviço e concorda com o montante apresentado
	13	<<include>> Efetuar Pagamento
	14	Finaliza o agendamento
	15	
	16	
<b>Exceção 1</b> [sem data] (passo 6)	1	
<b>Exceção 2</b> [sem morada] (passo 6)	1	
<b>Exceção 3</b> [sem horas] (passo 6)	1	
<b>Exceção 4</b> [sem profissional] (passo 9)	1	

<b>Exceção 4</b> [sem profissional] (passo 9)	1		Informa cliente que não escolheu um profissional
<b>Alternativa 1</b> [não concorda com valor] (passo 12)	1	Não concorda com valor apresentado	
	2	Cancela agendamento	
	3		Indica que serviço não será agendado

Figura 8 - Especificação do Use Case "Agendar trabalho"



## **3.2. Diagramas de sequência**

Por forma a melhor descrever o comportamento do sistema, principalmente em relação ao utilizador, desenvolveram-se alguns diagramas de sequência. Estes, que aprofundam os Use Cases especificados, descrevem as funcionalidades do sistema a encontrar a cada passo.

Este tipo de diagrama, pela sua tipologia de interação Utilizador-Sistema, permite a aproximação e perceção das necessidades relativas à fase de desenvolvimento. É assim tornado claro para a equipa de desenvolvimento os componentes e funcionalidades a desenvolver.

### **3.2.1 Exemplo de Diagrama de Sequência - Aceitar proposta de trabalho**

De forma a explicitar o Use Case fornecido anteriormente, dado pelo nome de “Aceitar proposta de trabalho”, é apresentado, de seguida, o seu diagrama de sequência.

Neste diagrama é possível verificar que o ator corresponde ao profissional, que usará a aplicação. Tal como explicitado, para um funcionário aceitar determinada proposta de trabalho, é exigida ao sistema uma obtenção de todas as propostas de trabalho à espera de aceitação, para aquele funcionário. Após esta obtenção, e caso haja propostas, deverá ser selecionada a proposta a aceitar, sendo que sistema verificará se esta ainda está disponível. Não sendo possível aceitar este trabalho (por eventual cancelamento), o funcionário é informado desta situação e termina a interação. O funcionário pode também escolher não aceitar a proposta de trabalho, sendo que esta informação surgirá informada ao cliente. Caso tudo corra bem, o sistema guardará a aceitação do trabalho e informará o funcionário que já o pode consultar junto dos seus trabalhos agendados.

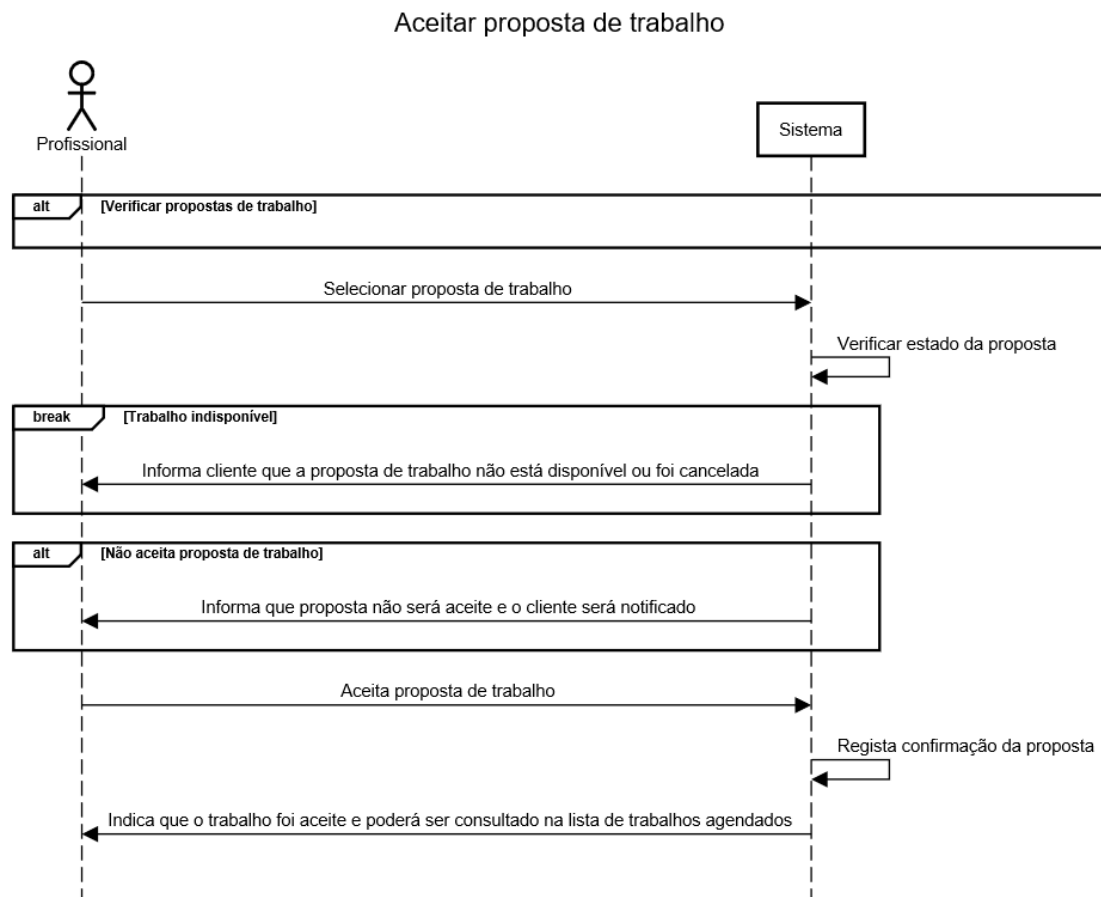


Figura 9 - Diagrama de Sequência "Aceitar Proposta de Trabalho"

### 3.2.2 Exemplo de Diagrama de Sequência - Avaliar trabalho

Em sequência com o Use Case fornecido anteriormente, dado pelo nome de "Avaliar trabalho", é apresentado, de seguida, o seu diagrama de sequência.

Neste diagrama é possível verificar que o ator corresponde ao cliente, que usará a aplicação. Tal como explicitado, para um cliente avaliar determinado trabalho, é exigida ao sistema uma obtenção de todos os trabalhos já efetuados, por ordem daquele cliente. Após esta obtenção, e caso haja trabalhos já efetuados, deverá ser selecionado aquele a avaliar, sendo que sistema verificará se este já terá ou não sido avaliado. Não sendo possível proceder à avaliação (por já ter sido feita), o cliente é informado desta situação e termina a interação. Caso tudo corra bem, o ator poderá inserir a avaliação (0-5), sendo que o sistema registá-la-á, informando o sucesso do procedimento.

## Avaliar trabalho

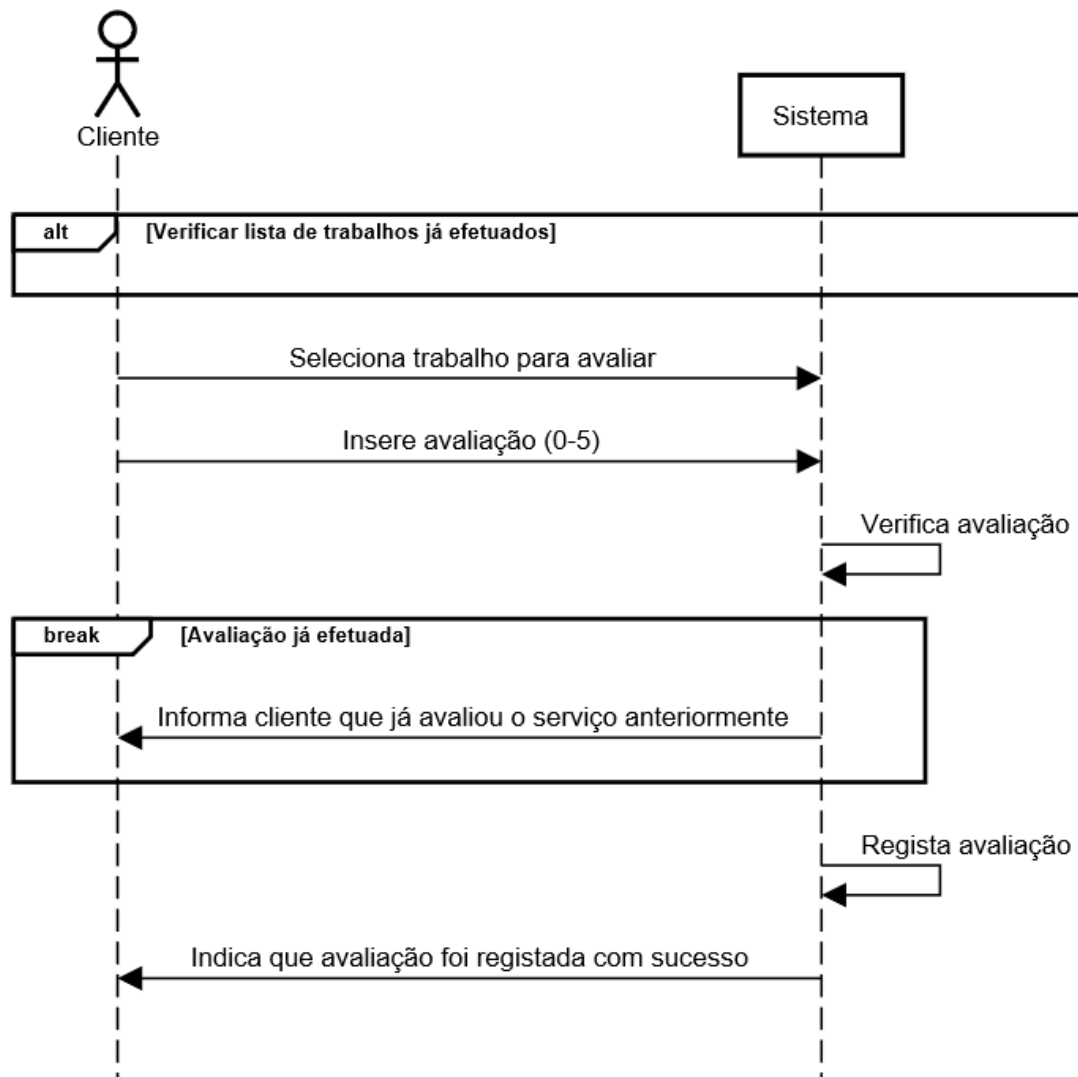


Figura 10 - Diagrama de Sequência "Avaliar Trabalho"

### 3.2.3 Exemplo de Diagrama de Sequência - Registo

Por fim, e de acordo com o Use Case fornecido anteriormente, dado pelo nome de “Registo”, é apresentado, de seguida, o seu diagrama de sequência.

Neste diagrama é possível verificar que o ator corresponde ao cliente ou funcionário, que usará a aplicação. Tal como explicitado, para um ator fazer o seu registo, deve inserir o username, nome, email, contacto e respetiva password. Após confirmados os dados inseridos, podem surgir duas situações. Na primeira, os dados inseridos são insuficientes, sendo que a interação termina. Na seguinte, será verificado se o utilizador já existe. Caso exista, o ator será notificado disto e abortado o processo de registo. Caso contrário, tendo corrido tudo bem, o utilizador será registado pelo sistema, que informará o sucesso do registo.

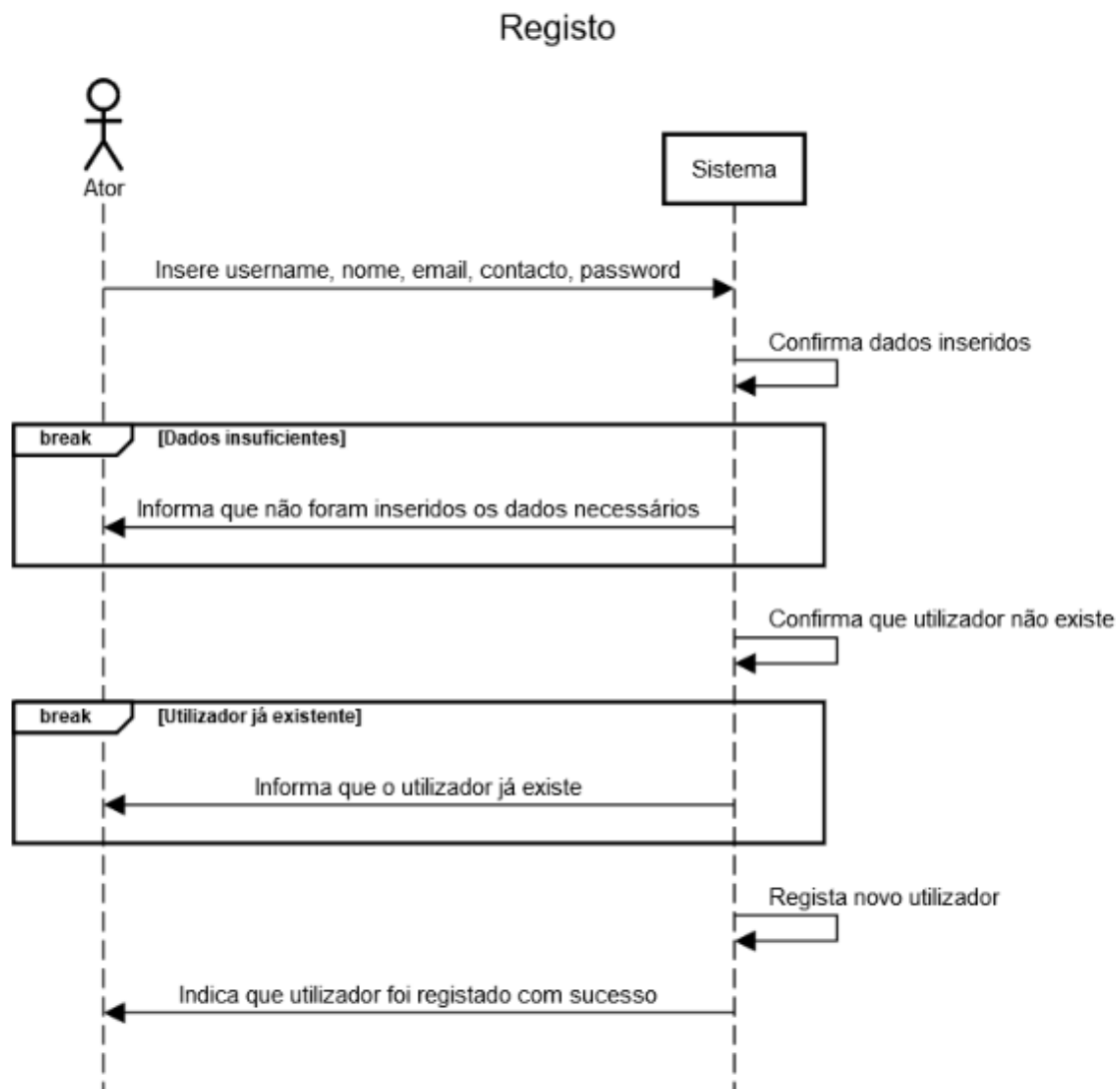


Figura 11 - Diagrama de Sequência "Registo"

### **3.2.4 Exemplo de Diagrama de Sequência – Agendar Trabalho**

Por fim, e de acordo com o Use Case fornecido anteriormente, dado pelo nome de “Agendar Trabalho”, é apresentado, de seguida, o seu diagrama de sequência.

Neste diagrama é possível verificar que o ator corresponde ao cliente, que usará a aplicação. Tal como explicitado, para agendar um serviço o cliente deverá seleccionar data para o agendamento, bem como inserir a rua e localidade. Seleccionará também a hora de início e fim, bem como possíveis serviços complementares, caso queira fazer uma atividade no exterior ou de estudo. Adicionalmente, poderá adicionar observações que queira, não sendo obrigatório fazê-lo. Após confirmados os dados inseridos, podem surgir quatro situações de erro, que resultam no término da interação.

Neste momento, caso tudo tenha corrido bem, será apresentada a lista de profissionais existente e o utilizador poderá escolher um profissional, sendo essa escolha verificada pelo programa. Novamente, caso não tenha sido escolhido o profissional, a interação é abortada. Se tudo até ao momento foi validado, e após verificado o estatuto do cliente, será apresentado o custo do serviço ao cliente, que o poderá aceitar ou não, efetuando o pagamento e confirmando o agendamento ou, caso não aceite aquele valor, cancelar o agendamento.

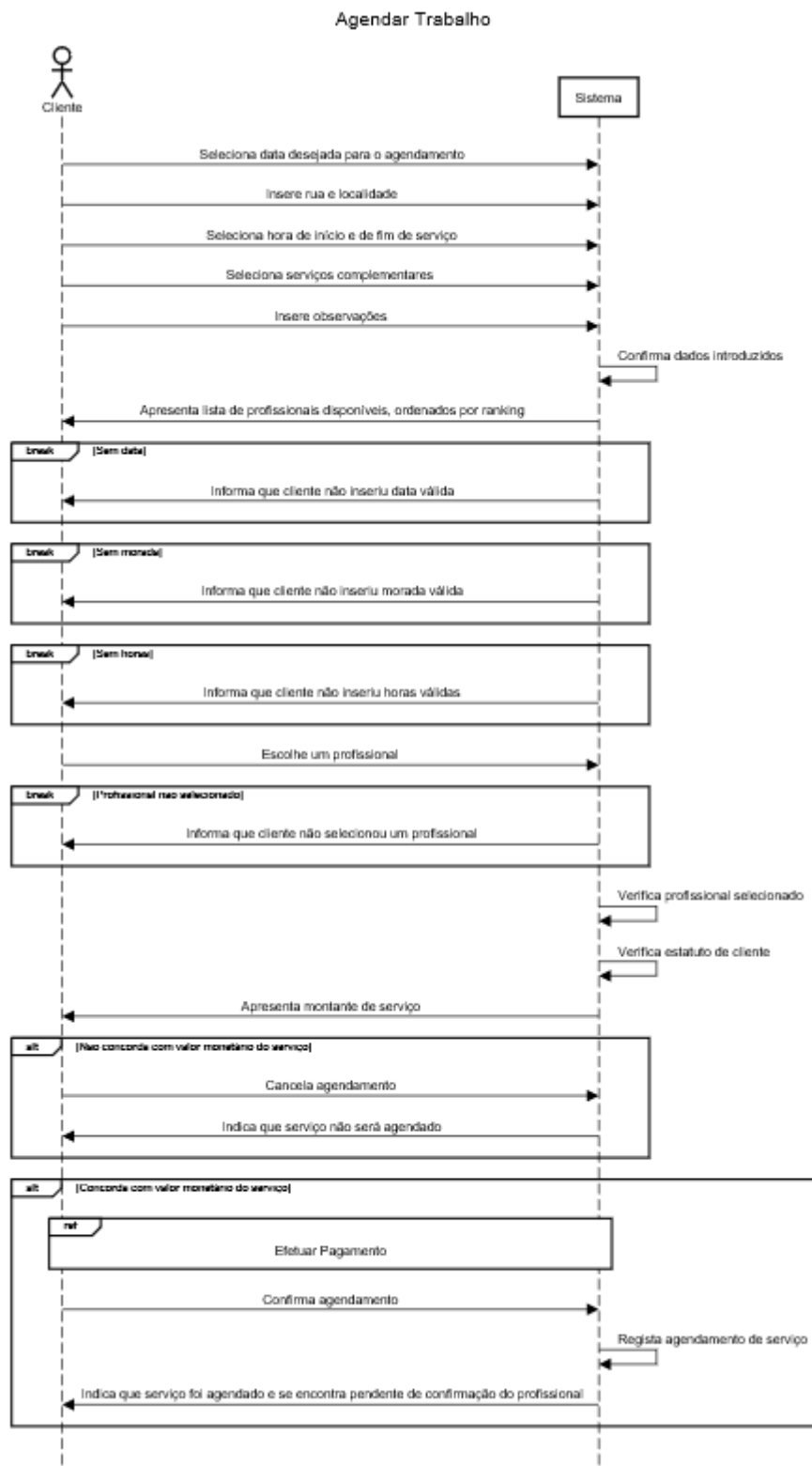


Figura 12 - Diagrama de Sequência "Agendar Trabalho"

### 3.3. Diagramas de Máquinas de Estado

Os diagramas de máquinas de estado modelam todos os estados possíveis que o sistema atravessa em resposta aos eventos que podem ocorrer e que são, de qualquer maneira, originados pelo utilizador. Deste modo, desenvolveram-se estes diagramas no sentido de representar, da melhor forma possível, aquela que virá a ser a interface da aplicação.

#### 3.3.1 GuguDadah

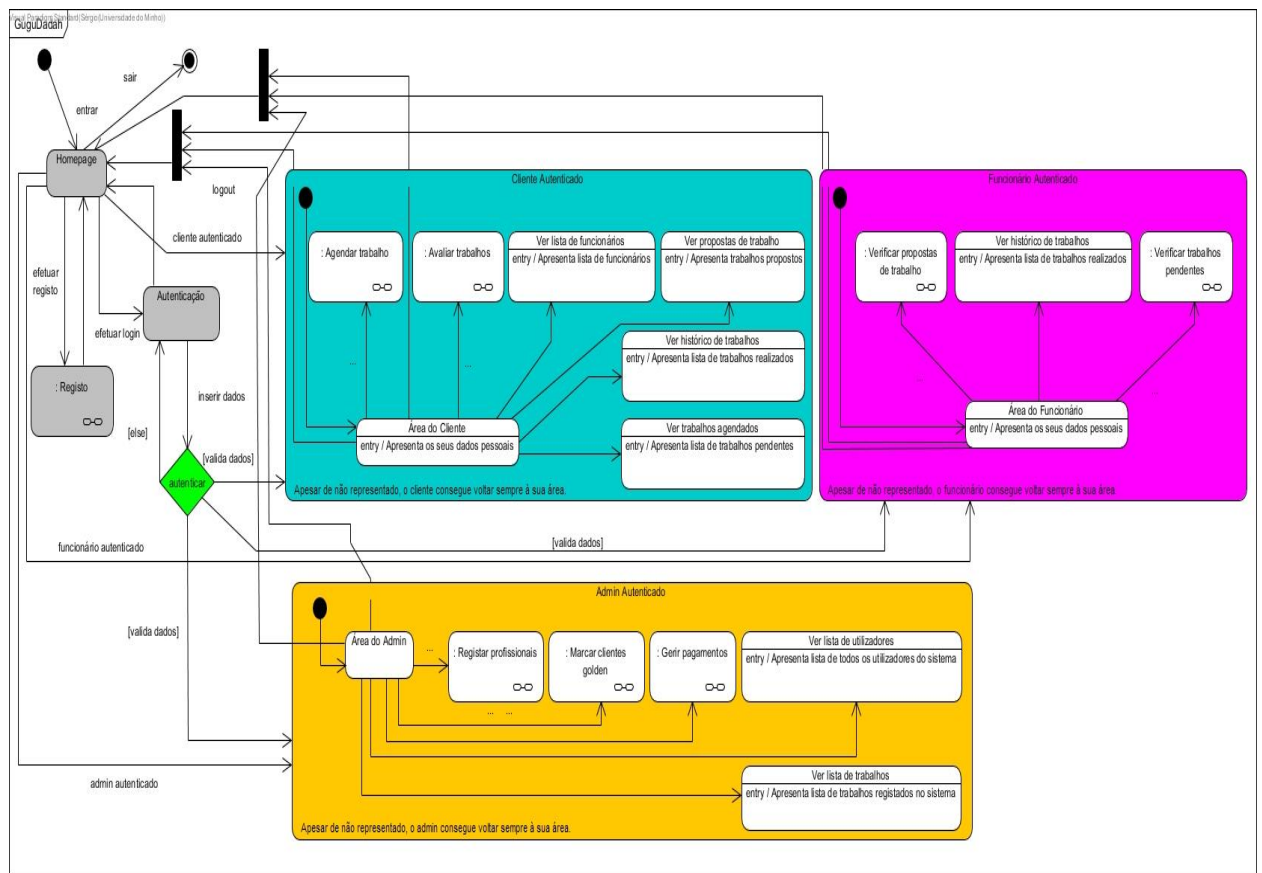


Figura 13 - Diagrama geral de máquinas de estado, com as respectivas sub-máquinas

### 3.3.2 Agendar Trabalho

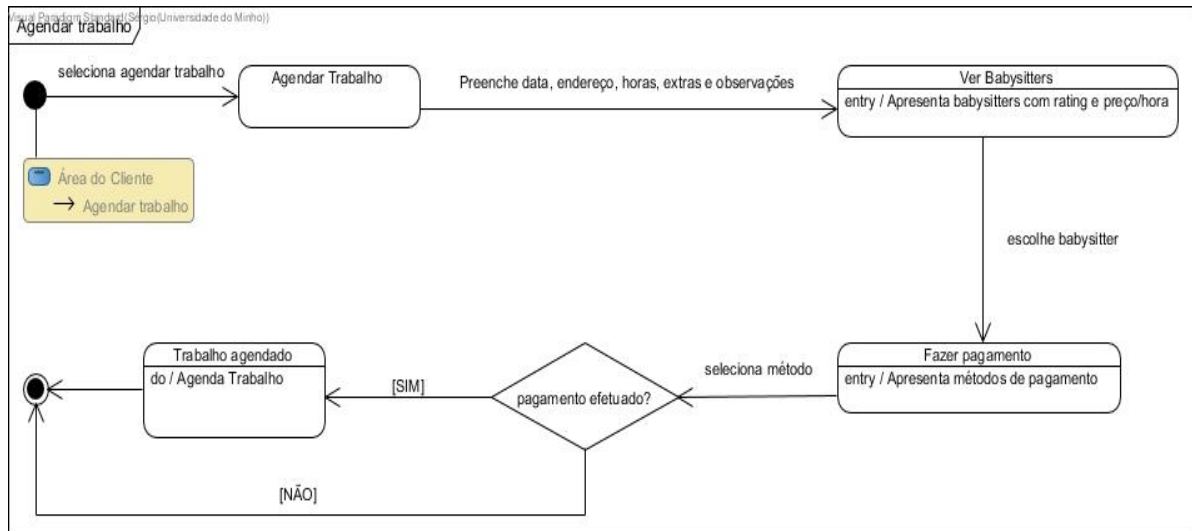


Figura 14 - Diagrama de máquinas de estado - Agendar Trabalho

### 3.3.3 Avaliar Trabalho

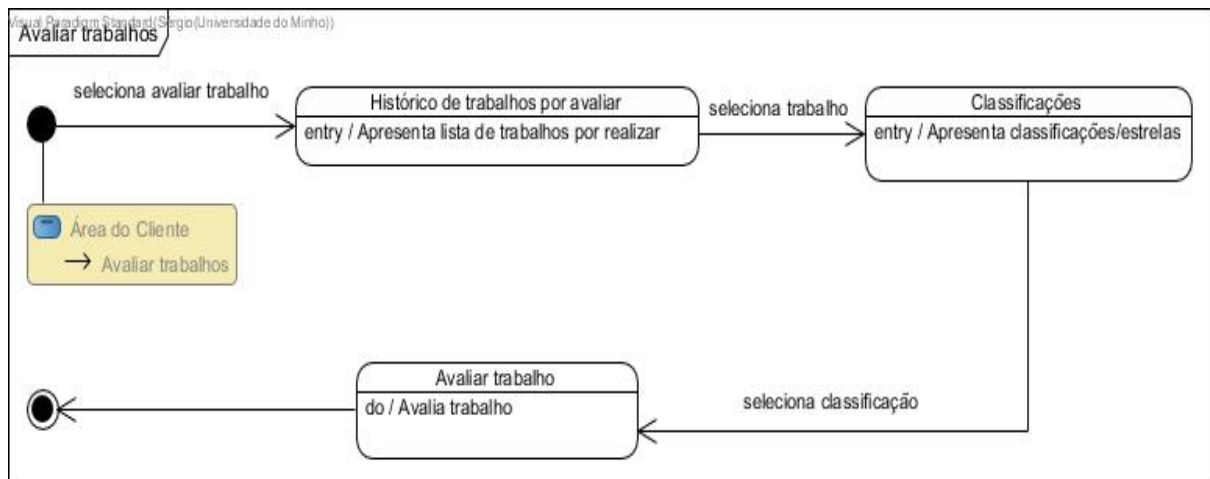


Figura 15 - Diagrama de máquinas de estado - Avaliar trabalho



### 3.3.4 Registo de Clientes

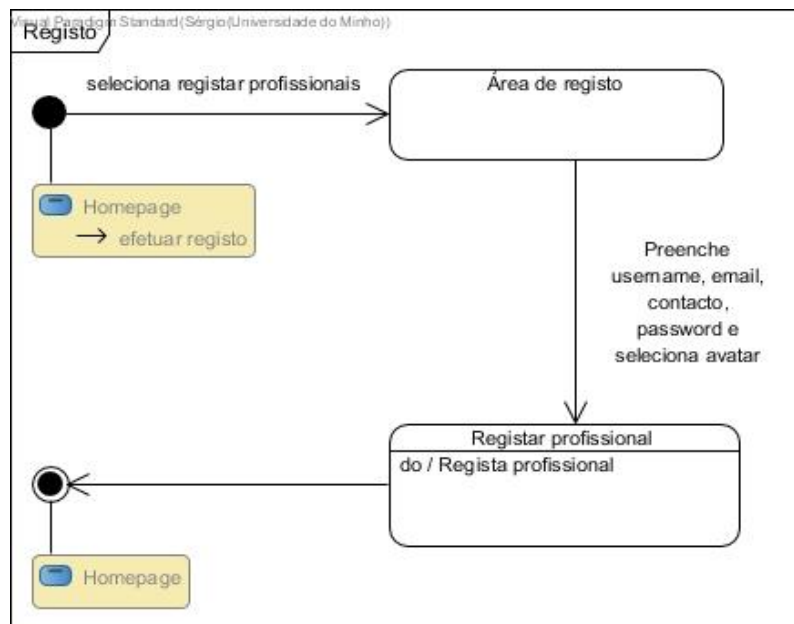


Figura 16 - Diagrama de máquinas de estado - Registo de clientes

### 3.3.5 Registo de Profissionais

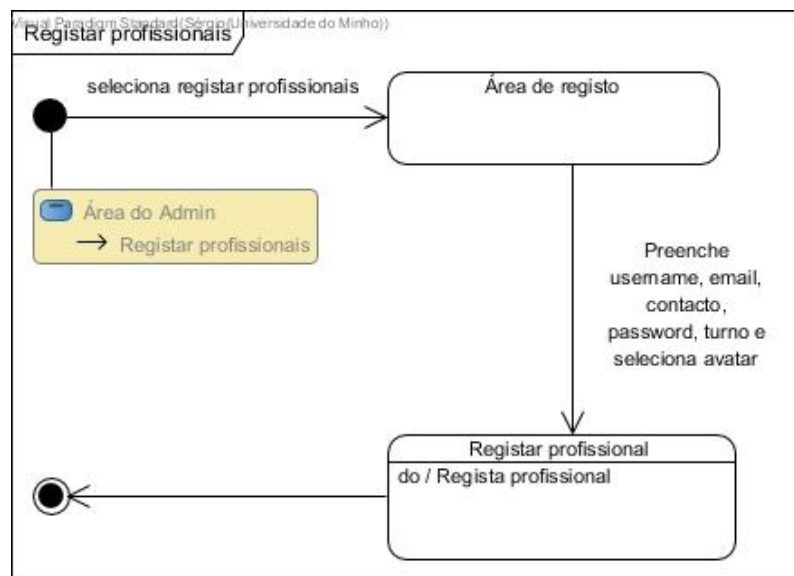


Figura 17 - Diagrama de máquinas de estado - Registo de profissionais

### 3.3.6 Gerir pagamentos

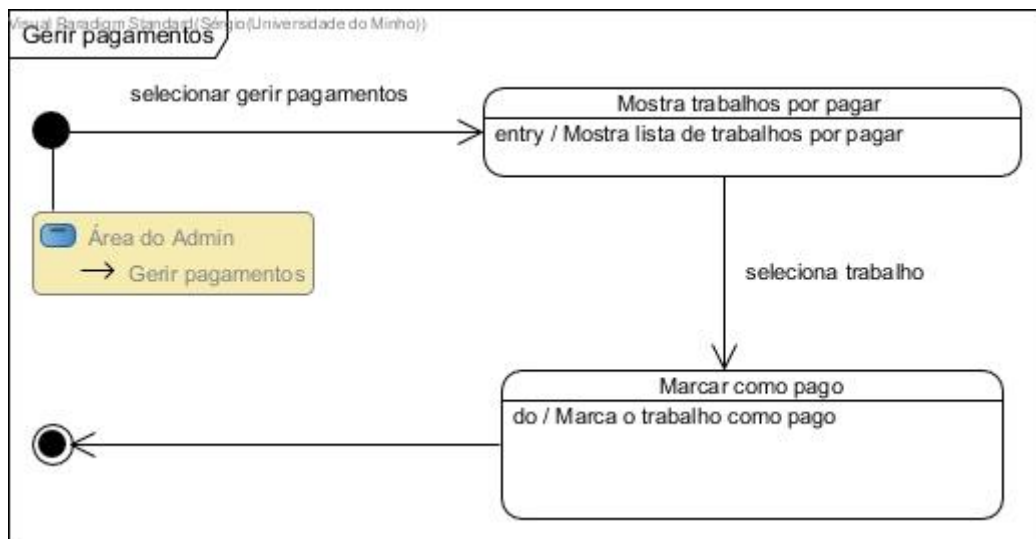


Figura 18 - Diagrama de máquinas de estado - Gerir pagamentos

### 3.3.7 Marcar Clientes Golden

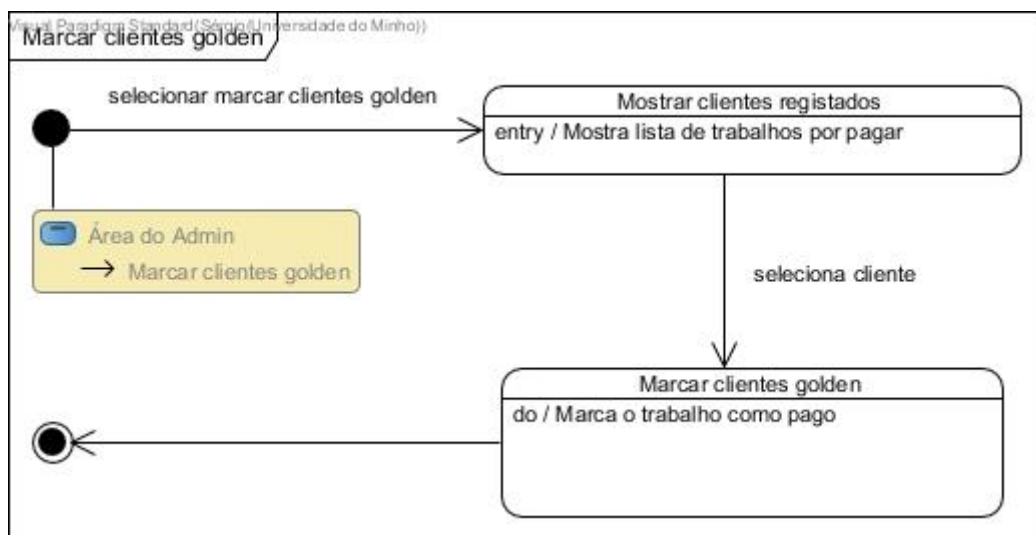


Figura 19 - Diagrama de máquinas de estado - Marcar Clientes Golden

### 3.3.8 Verificar Propostas de Trabalho

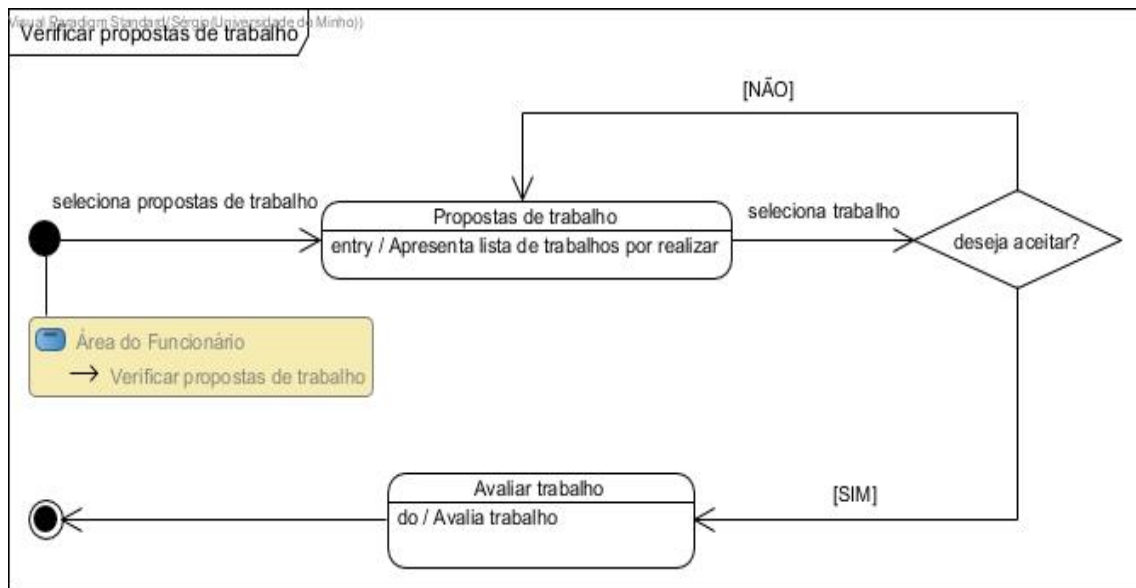


Figura 20 - Diagrama de máquinas de estado – Verificar Propostas de Trabalho

### 3.3.9 Verificar Trabalhos Pendentes

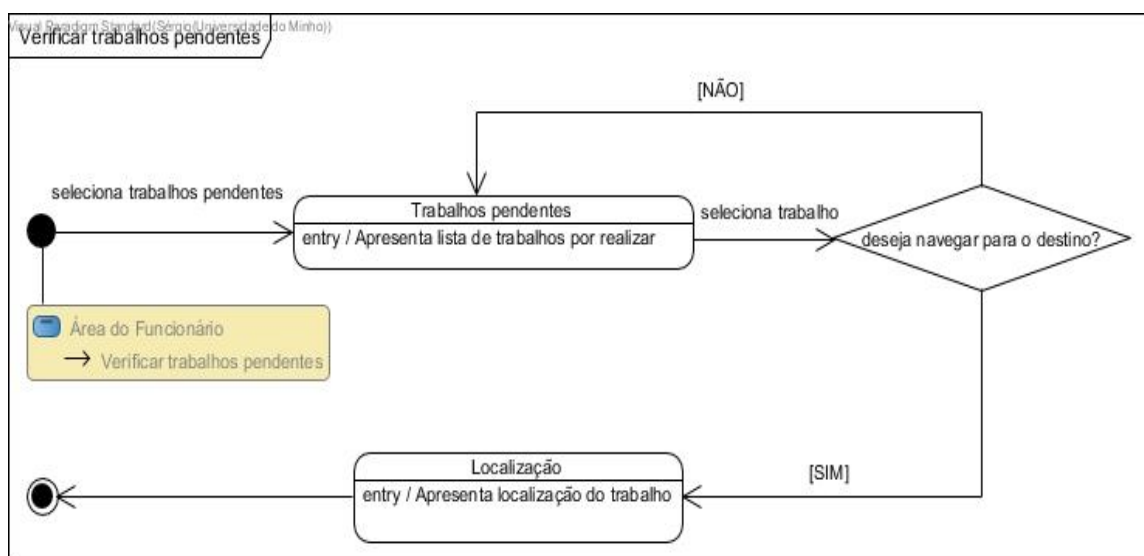


Figura 21 - Diagrama de máquinas de estado - Verificar Trabalhos Pendentes

### 3.4. Diagrama de atividades

De seguida é ilustrado o diagrama de atividades, em que é descrito a evolução da aplicação do serviço. Começando no agendamento do cliente, inserem-se os dados requeridos da ação. Aqui o fluxo toma uma alternativa, que é quando o cliente insere dados incorretos ou insuficientes, como data errada, ou não ter inserido a localização, situações essas que são informadas pelo sistema. Depois, o sistema fornece a lista de profissionais disponíveis ao cliente naquele momento, sendo que o cliente escolhe um.

Posteriormente o sistema verifica se o cliente é Golden ou não, para fazer um desconto na estimativa de preço. Neste ponto, o serviço continua caso o cliente concorde. Se não, volta para o ponto de escolha de profissionais. Quando o funcionário é escolhido, o sistema indica ao mesmo, que foi destacado para realizar a tarefa em causa. Aqui o curso da atividade, pode tomar três direções. Durante 10 minutos, ou o babysitter aceita a proposta, ou recusa, ou ainda o tempo é excedido. Nestas duas últimas o programa é interrompido, e concluído de uma maneira ortodoxa.

Caso aceite, o sistema informa paralelamente, tanto o funcionário como o cliente, que a proposta foi aceite, e os seus detalhes. Existe a possibilidade de o responsável pela atividade não se deslocar ao local, independentemente do motivo, se assim o acontecer, o mesmo é penalizado, e o serviço é cancelado. Caso o funcionário realize o trabalho requerido, o cliente efetua o pagamento (ao infantário, não ao babysitter), e o sistema atualiza o estado da tarefa. Por fim, o cliente avalia o serviço prestado quando desejar, e depois o sistema atualiza o rating do funcionário.



### 3.5. Diagrama de Classes

Este diagrama representa, em alto nível, as classes do sistema bem como os seus atributos e métodos. O diagrama foi construído com auxílio dos requisitos e dos mockups apresentados.

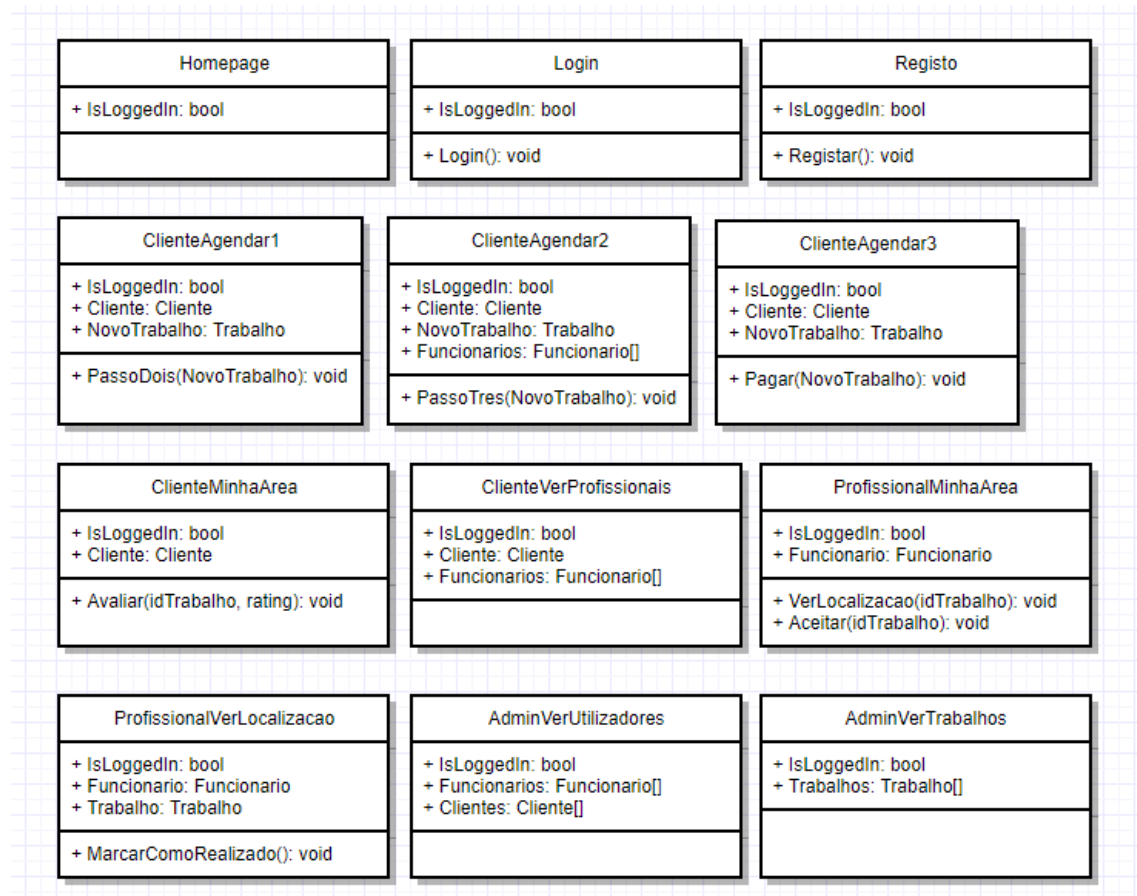


Figura 23 - Diagrama de classes

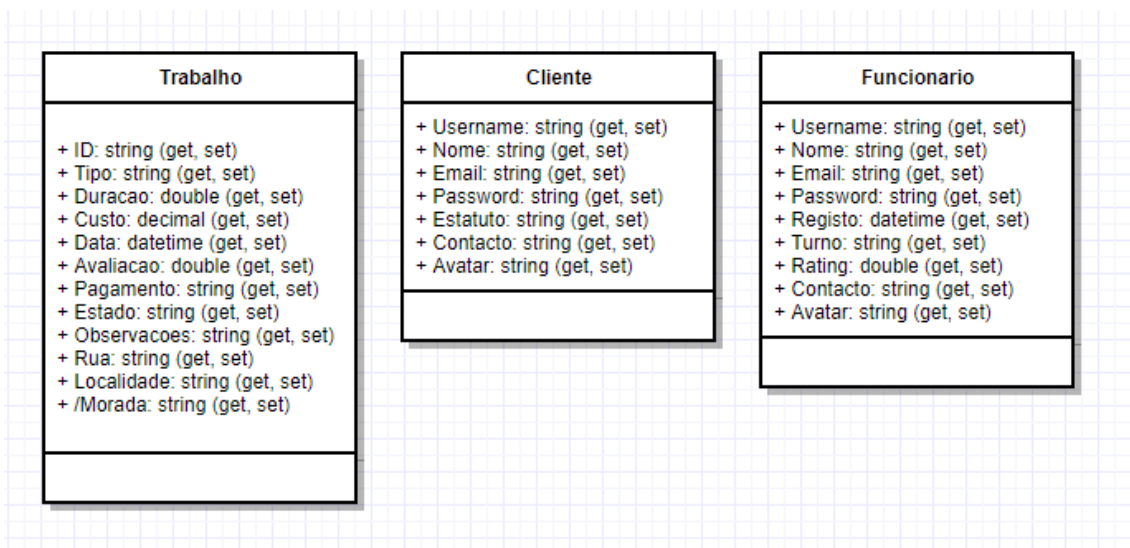


Figura 24 - Diagrama de classes - DAOs

## 4. Base de Dados

### 4.1. Modelo Conceptual

#### 4.1.1 Identificar os tipos de identidades

Após a leitura e interpretação dos requisitos, identificaram-se vários objetos chave. De entre estes, surgiram as entidades: cliente, trabalho e funcionário.

Nome da Entidade	Descrição	Sinónimos	Ocorrências
Cliente	Cliente que pretende usar a aplicação e encontrar alguém que cuide do seu filho	Utilizador, Responsável	Um cliente pode selecionar um cuidador para, em determinadas horas, cuidar do filho
Trabalho	Trabalho prestado a um cliente por um dado funcionário, que regista toda a informação útil sobre o mesmo	Serviço	Um cliente pede um trabalho, a determinado funcionário
Funcionário	Funcionário que presta serviços de cuidado, a pedido do cliente, mediante sua disponibilidade	Cuidador, Babysitter	Um funcionário pode aceitar ou não um serviço que lhe é proposto

#### 4.1.2 Identificar tipos de relacionamento

Uma vez definidas as entidades do sistema, torna-se imperativo estabelecer as relações entre as mesmas. Novamente, por leitura e interpretação dos requisitos, identificaram-se os relacionamentos e respetiva multiplicidade, apresentados na tabela seguinte:



Nome da Entidade	Multiplicidade	Relacionamento	Multiplicidade	Nome da Entidade
Cliente	1	Requisita	N	Trabalho
Trabalho	N	Tem	1	Funcionário

### 4.1.3 Identificar e associar atributos com os tipos de entidades e relacionamentos

Tendo cada uma das entidades um conjunto de informações importantes a reter, as quais foram especificadas pelo cliente, surge a seguinte tabela. Esta evidencia todos os atributos retidos em cada uma das entidades.

### 4.1.4 Atributos simples/compostos

Após a leitura dos requisitos, chegou-se à conclusão que não existe a necessidade de ter quaisquer atributos compostos.

### 4.1.5 Atributos derivados

Após o levantamento de requisitos e identificação dos atributos, foi possível verificar que o atributo Custo é derivado, para a entidade Trabalho. O seu cálculo é feito através de uma fórmula que conjuga Duração e Tipo (da entidade Trabalho), com Turno (da entidade Funcionário) e Estatuto (da entidade Cliente).

### 4.1.6 Atributos multivalor

Da mesma análise referida nos pontos anteriores, é possível concluir que não existem atributos multivalor. É de notar que, caso fosse requisito haver vários contactos para o Cliente, surgiria um atributo multivalor Contacto.

### 4.1.7 Associação entre atributos e entidades

Apresenta-se, de seguida, a relação de atributos e respetivas entidades.

Nome da entidade	Atributo	Descrição	Tipo	Nulo	Multivalor	Derivado	Composto
<b>Cliente</b>	Username	É o username, que o identifica	Caracteres	Não	Não	Não	Não
	Nome	Nome do cliente	Caracteres	Não	Não	Não	Não
	Email	Email com que o cliente se registou	Caracteres	Não	Não	Não	Não
	Password	Password atual da conta do cliente	Caracteres	Não	Não	Não	Não
	Estatuto	Condição que pode permitir acesso a descontos	Char	Não	Não	Não	Não
	Contacto	Telemóvel que o cliente registou	Valor inteiro positivo	Não	Não	Não	Não
	Avatar	Imagem representativa do cliente	Binários	Sim	Não	Não	Não
<b>Trabalho</b>	Id	Identificador daquele trabalho	Valor inteiro positivo	Não	Não	Não	Não
	Tipo	Tipo de trabalho que vai ser desempenhado	Char	Não	Não	Não	Não
	Duração	Duração do trabalho em causa (minutos)	Valor inteiro positivo	Não	Não	Não	Não
	Custo	Custo final do trabalho realizado	Valor decimal positivo	Não	Não	Sim	Não
	Data	Data e Hora do	DATETIME	Não	Não	Não	Não

		início do trabalho					
	Avaliação	Avaliação do trabalho, caso seja avaliado	Valor decimal positivo	Sim	Não	Não	Não
	Pagamento	Tipo de pagamento efetuado (CC ou dinheiro)	Char	Não	Não	Não	Não
	Estado	Estado atual do pedido de serviço	Char	Não	Não	Não	Não
	Observações	Observações efetuadas ao serviço	Caracteres	Sim	Não	Não	Não
	Morada	Morada de prestação de serviço	Caracteres	Não	Não	Não	Não
<b>Funcionário</b>	Username	É o username, que o identifica	Caracteres	Não	Não	Não	Não
	Nome	Nome do funcionário	Caracteres	Não	Não	Não	Não
	Email	Email com que o funcionário está registado	Caracteres	Não	Não	Não	Não
	Password	Password atual da conta do funcionário	Caracteres	Não	Não	Não	Não
	Turno	Turno em que funcionário se encontra disponível	Char	Não	Não	Não	Não
	Rating	Rating atribuído ao funcionário	Valor decimal positivo	Sim	Não	Não	Não
	Contacto	Telemóvel que o funcionário está registado	Valor inteiro positivo	Não	Não	Não	Não
	Avatar	Imagem representativa	Binários	Sim	Não	Não	Não

		do funcionário					
	Desde	Data de registo do funcionário	Data	Não	Não	Não	Não
	Apresentação	Texto de apresentação do funcionário	Texto	Sim	Não	Não	Não

### 4.1.8 Determinar domínio dos atributos

Neste passo serão descritos os domínios dos atributos das várias entidades. Tornar-se-á evidente o tipo e os valores que os atributos tomarão, tal como o porquê de tal decisão.

- Cliente
  - Username: String, que poderá conter todo o tipo de caracteres. Com tamanho limitado a 10, por questões de eficiência.
  - Nome: String, que conterà o nome do cliente.
  - Email: String, que conterà o email de registo do cliente.
  - Password: String, que conterà a password do cliente. Como a aplicação não necessita de elevada segurança, esta ficará guardada na mesma Base de Dados que todas as outras informações.
  - Estatuto: Char, que identifica o cliente como sendo Gold ou não, dependendo se é utilizador do infantário.
  - Contacto: Int, que guarda o número do cliente.
  - Avatar: VARBINARY, que guardará a imagem do cliente. É guardado diretamente na BD pois a aplicação é de pequena dimensão e diminui a carga de implementação. De facto, não há necessidade de ter um host de imagens e de fazer backups ao mesmo, ao invés de fazer apenas a toda a BD.
- Trabalho
  - Id: Int, identifica o trabalho requerido. Auto incremental.
  - Tipo: Char, que identificará o(s) serviço(s) complementar(es) (caso haja(m)) selecionados pelo cliente.
  - Duração: Int, em minutos que conterà a duração da prestação de serviço.
  - Custo: Float, que é derivado e representa o custo total.
  - Data: DATETIME, representa a hora e dia de início da atividade.

- Avaliação: Float, que guarda, caso seja avaliado, a pontuação dada ao trabalho.
- Pagamento: Char, que representa pagamento efetuado por Cartão de Crédito ou a Dinheiro. Caso seja a dinheiro, o responsável por serviço é encarregado de fazer a cobrança.
- Estado: Char, que identifica se o serviço está agendado, pendente ou terminado.
- Observações: Texto, que poderá existir ou não, de acordo com os comentários que o cliente desejar colocar.
- Morada: String, o local onde o trabalho é proposto ser efetuado. Não é dividido em Rua + Localidade pois a BD é pequena e o Parser do Bing Maps fará a devida compreensão. Implica menos necessidade de manutenção de tabelas a nível de implementação.
- Funcionário
  - Username: String, que poderá conter todo o tipo de caracteres. Com tamanho limitado a 10, por questões de eficiência.
  - Nome: String, que conterá o nome do funcionário.
  - Email: String, que conterá o email de registo do funcionário.
  - Password: String, que conterá a password do funcionário. Como a aplicação não necessita de elevada segurança, esta ficará guardada na mesma Base de Dados que todas as outras informações.
  - Turno: Char, que identifica o turno (manhã, tarde ou noite) em que o funcionário trabalha.
  - Rating: Float, que identifica o rating atual daquele funcionário.
  - Contacto: Int, que guarda o número do funcionário.
  - Avatar: VARBINARY, que guardará a imagem do funcionário. É guardado diretamente na BD pois a aplicação é de pequena dimensão e diminui a carga de implementação. De facto, não há necessidade de ter um host de imagens e de fazer backups ao mesmo, ao invés de fazer apenas a toda a BD.
  - Desde: Datetime, que guarda a data de registo daquele funcionário.
  - Apresentação: Text, que guardará o texto de apresentação daquele funcionário. Pode ser nulo.

#### **4.1.9 Determinar chaves primárias, candidatas e alternativas**

De forma a identificar unicamente cada uma das ocorrências das distintas entidades, foi necessário determinar os atributos chave.

Durante o processo de estudo, surgiram, para cada uma das entidades as seguintes chaves candidatas:

- Cliente: Username, Email
- Trabalho: Id
- Funcionário: Username, Email

Tendo sido seleccionadas para chaves primárias:

- Cliente: Username
- Trabalho: Id
- Funcionário: Username

Optou-se pelo uso do username ao invés do email uma vez que é uma chave de menor comprimento.

#### 4.1.10 Desenho do diagrama ER

Apresentamos, de seguida, o desenho de diagrama E-R (Entidade-Relacionamento) de forma a representar conceptualmente as relações entre as entidades da base de dados.

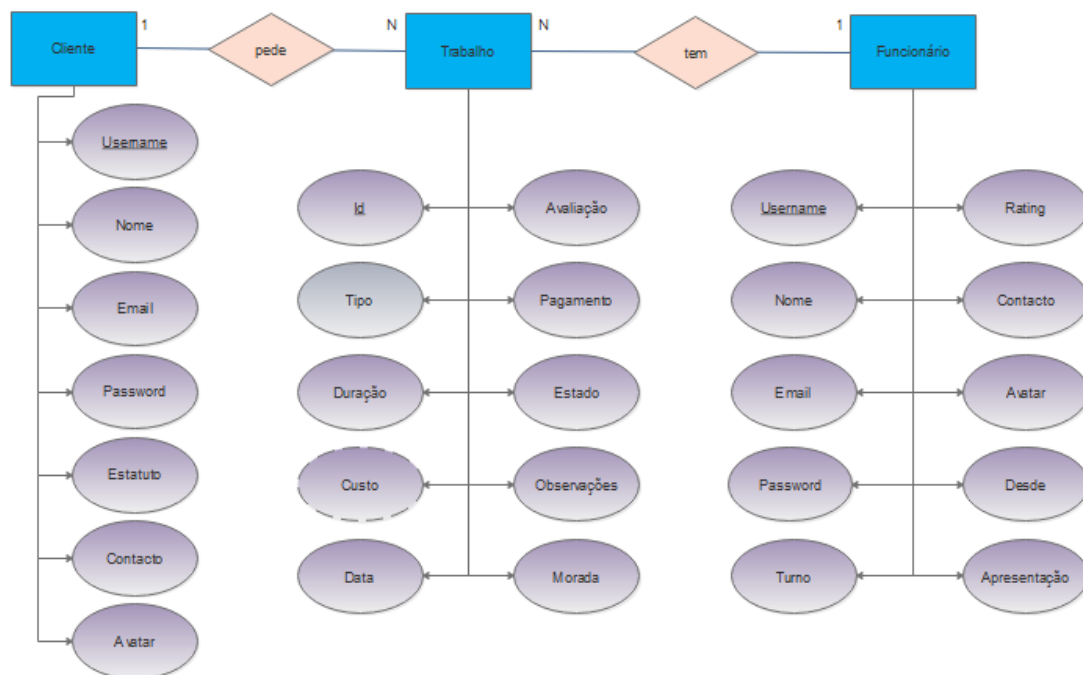


Figura 25 - Desenho do Diagrama ER

#### **4.1.11 Revisão do modelo de dados com o utilizador**

Após a realização do modelo conceptual, este foi revisto pelo utilizador. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

## 4.2. Modelo Lógico

### 4.2.1 Construção e validação do modelo de dados lógico

Nesta parte reproduzir-se-á o modelo lógico com base no modelo concetual anteriormente estabelecido. Por isso, derivam-se os relacionamentos no modelo lógico criando tabelas ou relações a fim de se representar as entidades, os atributos e os relacionamentos que foram identificados.

#### 4.2.1.1 Entidades fortes

Na modelação concetual realizada estão presentes três entidades: Cliente, Trabalho e Funcionário. Como tal, por serem independentes umas das outras, são então também entidades fortes. Por isso, para cada uma delas, foi criada uma tabela no modelo lógico, com os seus atributos.

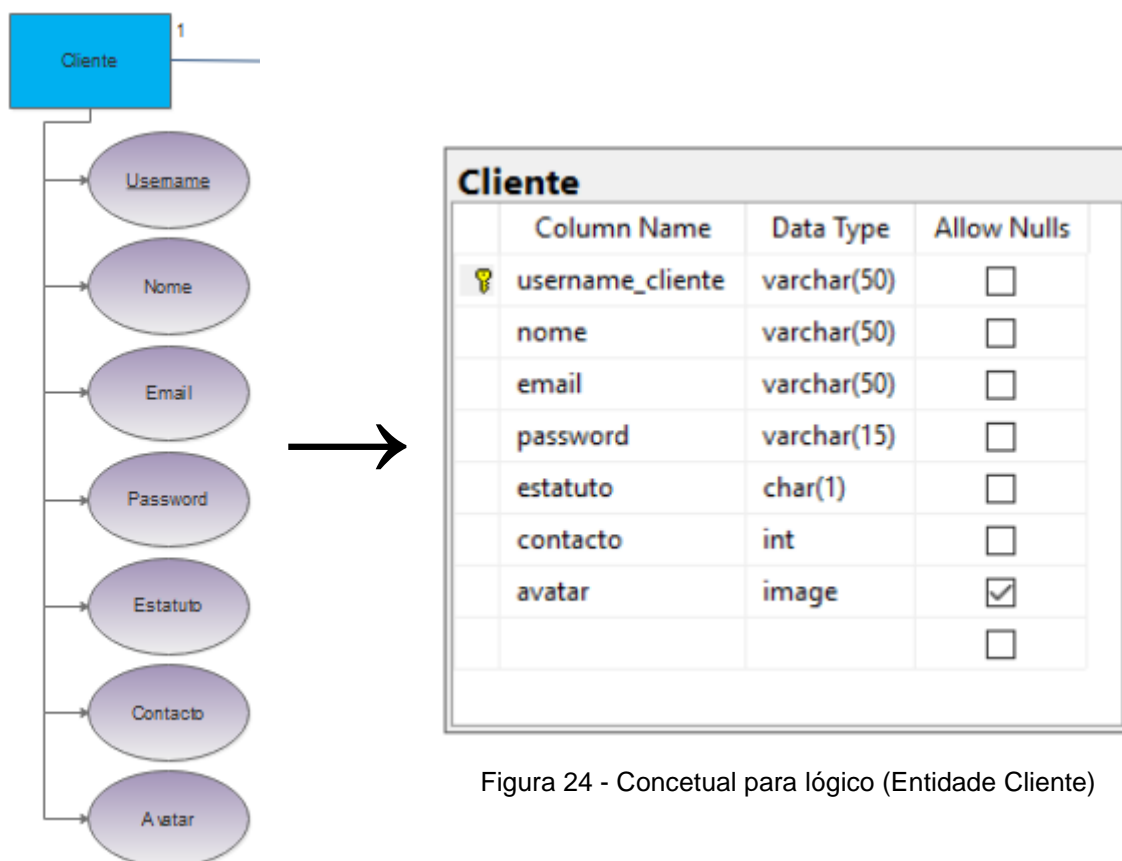


Figura 24 - Concetual para lógico (Entidade Cliente)



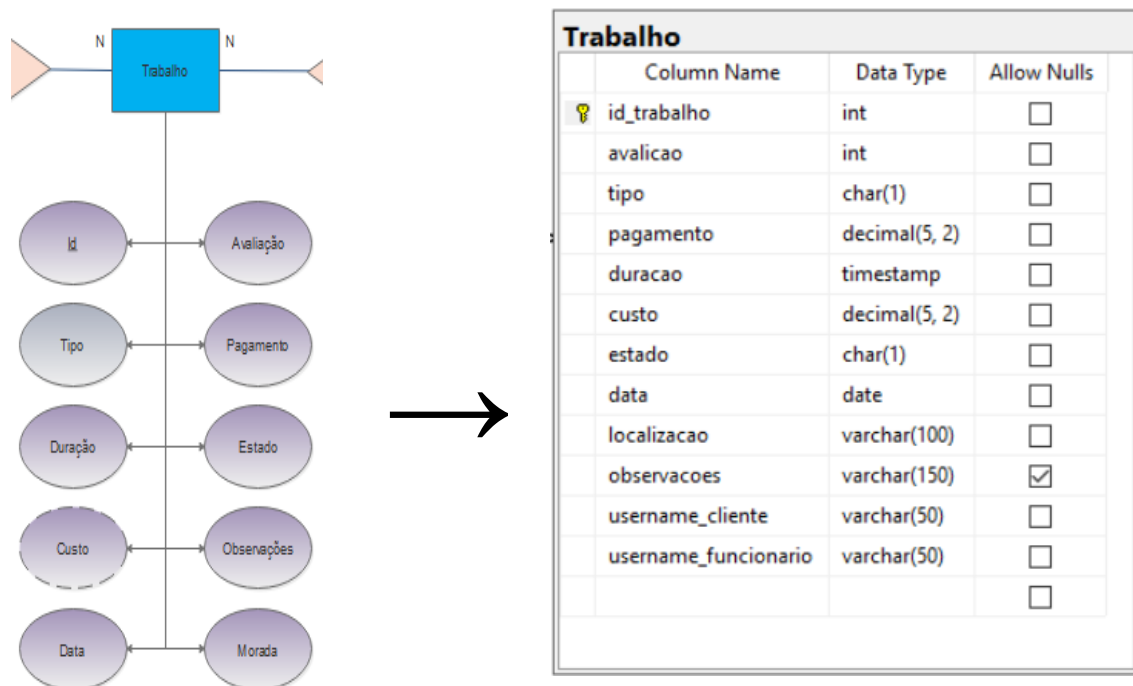


Figura 25 - Concetual para lógico (Entidade Trabalho)

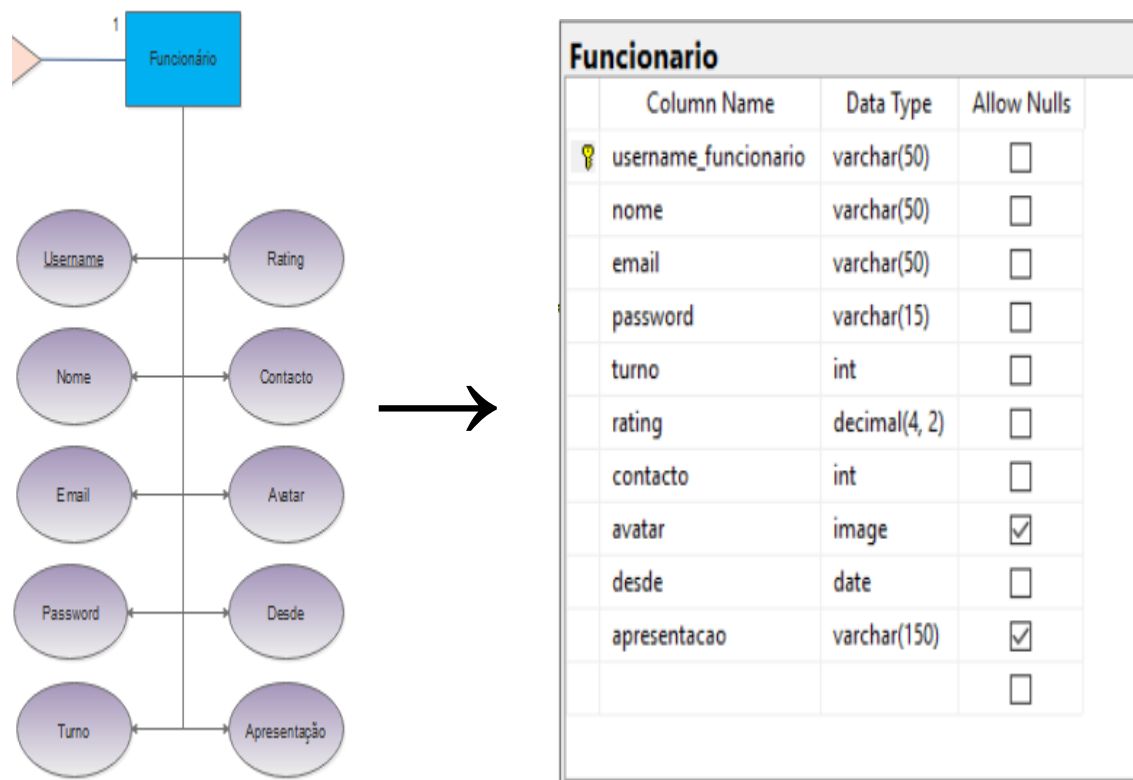


Figura 26 - Concetual para lógico (Entidade Funcionario)

### 4.2.1.2 Relacionamentos binários 1:N

No modelo conceitual desenhado, estão representadas duas relações binárias 1:N. Que são as relações entre Cliente e Trabalho, e entre Trabalho e Funcionário. Em cada caso é inserido um atributo no lado N da relação, que serve como chave estrangeira, em ambos os casos essa é a tabela Trabalho.

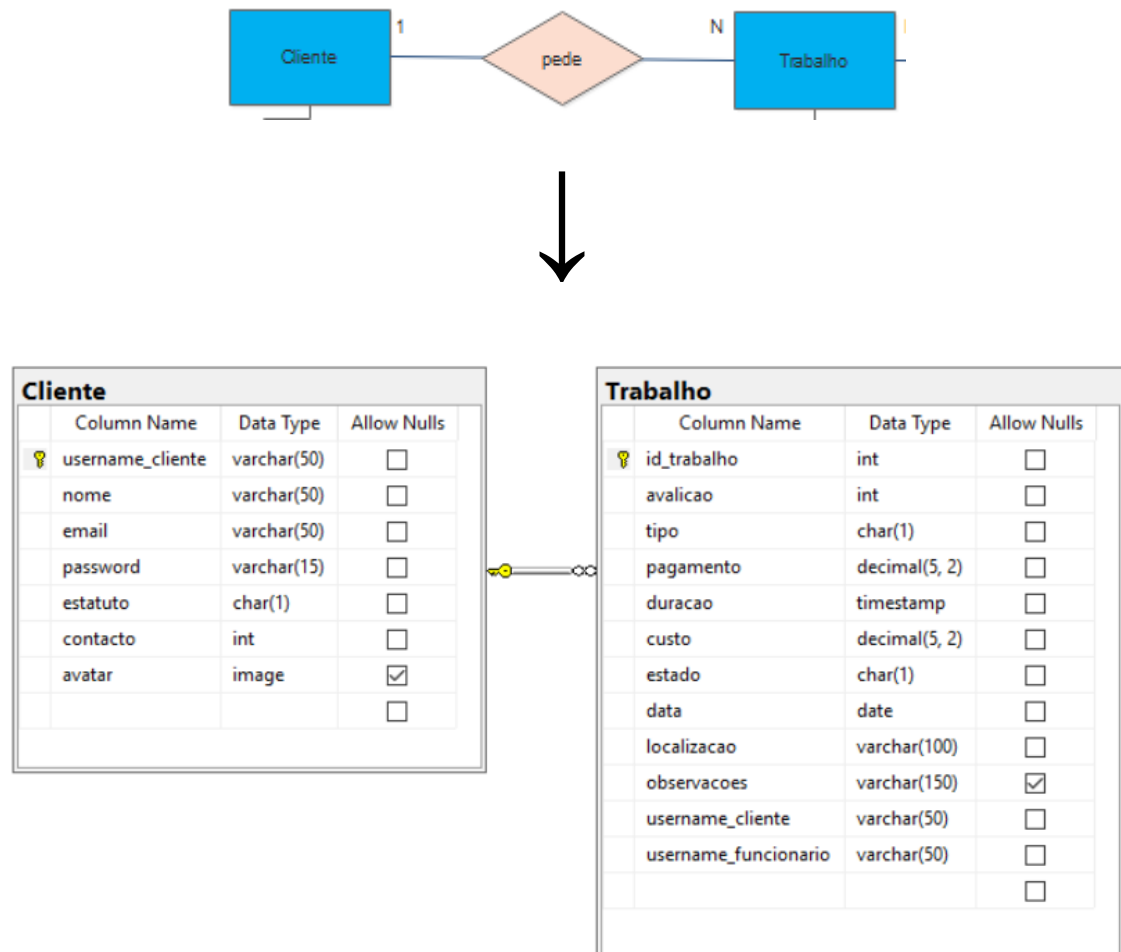


Figura 27 - Conceitual para lógico (relacionamento Cliente Trabalho)

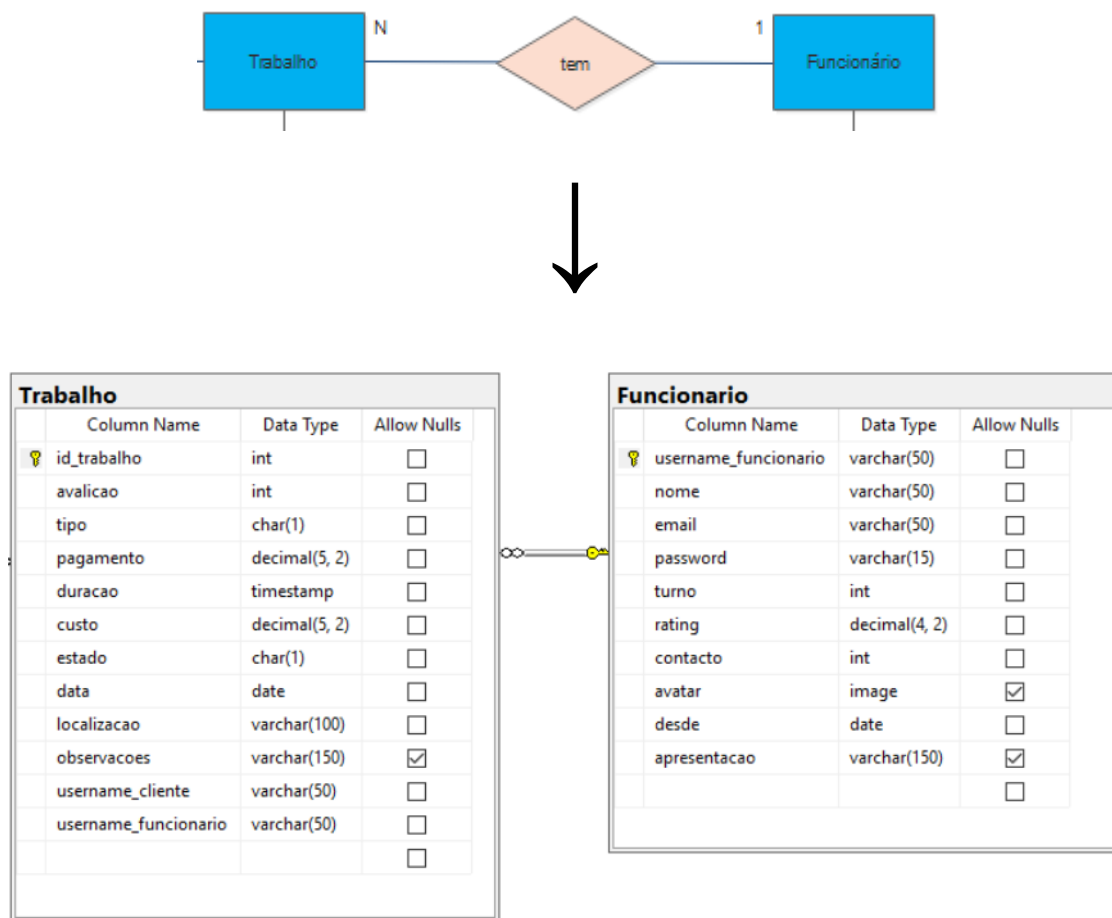


Figura 2826 - Concetual para lógico (Relacionamento Trabalho Funcionario)

## 4.2.2 Desenho do modelo lógico

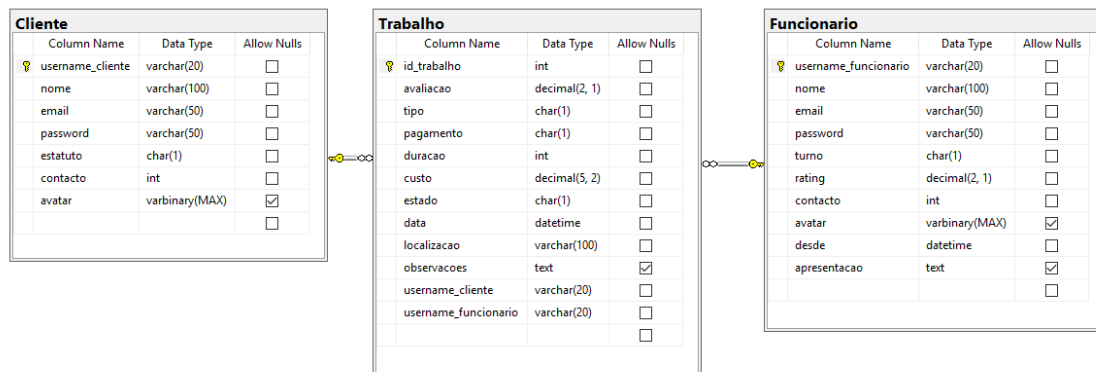


Figura 29 - Modelo lógico

## 4.2.3 Validação do modelo através da normalização

Neste momento, e com vista a eliminar redundâncias, urgindo aumentar o desempenho e aumentar a integridade dos dados, procede-se à normalização dos mesmos, até à terceira forma normal, inclusive.

### 1. 1ª Forma Normal

De modo a uma tabela estar de acordo com a 1ªFN, é necessário que a interceção de cada linha e coluna possua um, e apenas um valor.

Ora, de acordo com uma análise ao modelo atualmente contruído, é possível contatar que não existem entidades com atributos repetidos e multi-valorados, ou seja, têm um e um só valor. Assim todas as tabelas estão na 1ªFN.

### 2. 2ª Forma Normal

Como anteriormente foi provado, o modelo encontra-se de acordo com a 1ªFN, que é o primeiro requisito para estar de acordo com a 2ªFN. A outra condição necessária é que todos os atributos que não são chaves primárias, sejam completamente dependentes da chave primária. Algo que, através da observação das tabelas existentes, é também possível confirmar. Existem, no entanto, atributos que poderiam por si mesmos identificar uma tabela, como é o caso dos emails, no entanto foram anteriormente considerados como chaves candidatas, o que as torna também dependentes das primárias escolhidas.

### 3. 3ª Forma Normal

Finalmente, para uma tabela se encontrar na 3ªFN, é necessário que cumpra também a 2ªFN. E que também não possua alguma dependência transitiva, ou seja, atributos não chave, que dependam de outros atributos não chave. Algo que não acontece em nenhuma das tabelas existentes no modelo lógico construído.

## **4.3 Análise do crescimento futuro**

Embora a atual base de dados tenha sido criada para satisfazer as necessidades anteriormente mencionadas, é bastante flexível para trabalhar com alterações que sejam pretendidas. É possível reparar que é fácil tornar o sistema acessível para outras cadeias de infantários, ou até mesmo para profissionais da área que queiram trabalhar como agente livres de contrato.

## **4.4 Revisão do modelo lógico com o utilizador**

Após a realização do modelo lógico, este foi revisto pelo utilizador com o intuito de assegurar que este o considera uma representação fidedigna do que é pretendido. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

## 5. Mockups

Numa última fase, e como resultado de todo o planeamento, fizeram-se os Mockups. São imagens ilustrativas de como a página web deve ser organizada, sendo que o seu aspeto é básico, mostrando todas as funcionalidades expectáveis.

Estes permitirão ao cliente ter uma ideia do que vai obter, podendo apresentar críticas com vista à melhoria do sistema. Adicionalmente, serão um bom suporte aquando do desenvolvimento.

As primeiras quatro imagens são relativas à homepage. Nesta, pode ser feito o registo ou login. Uma vez feito o registo, o cliente (neste caso), terá acesso à sua Área Pessoal e poderá Agendar Serviços, entre outras funcionalidades acessórias. As figuras número cinco e seis nesta lista, demonstram, entre as funcionalidades adicionais, a consulta de todos os serviços e babysitters. Já nas figuras de sete a onze, demonstra-se um agendamento de serviço, onde se escolhe o babysitter e a forma de pagamento. Esta secção termina com a página do profissional, onde ele pode ver a localização do serviço, e as últimas três imagens representam a área do administrador, que pode fazer gestão de tudo o que envolve o serviço.

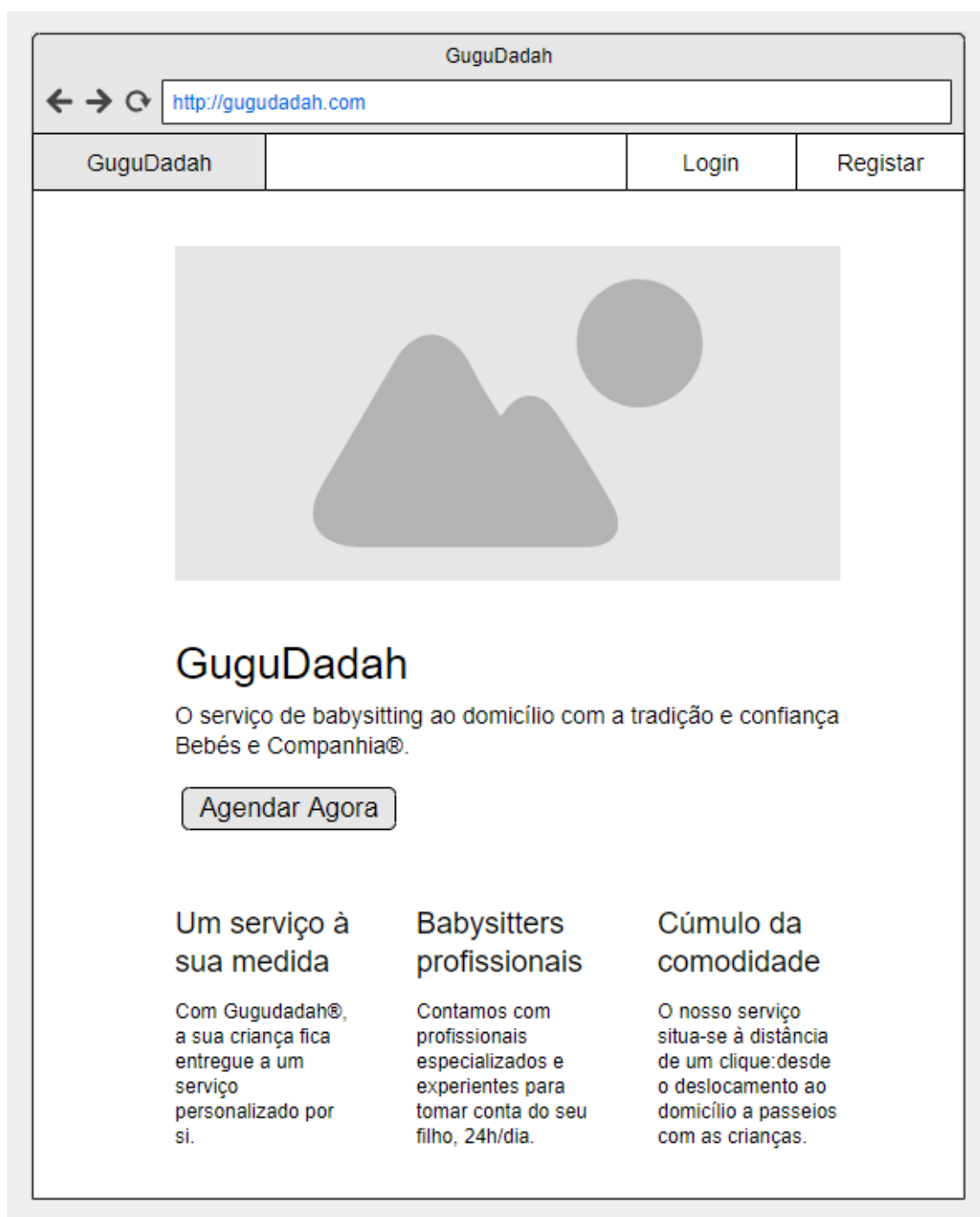


Figura 27 - Homepage

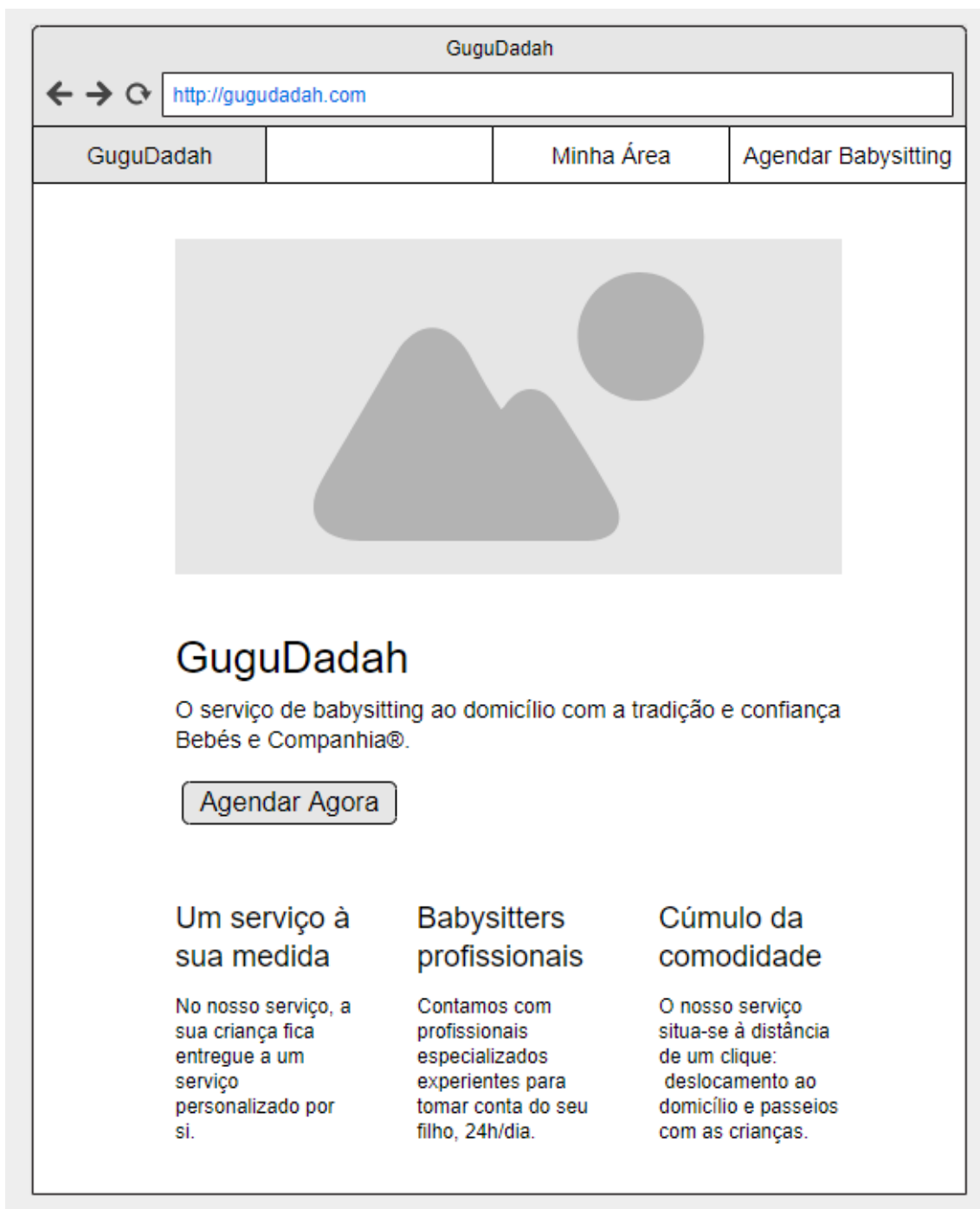


Figura 28 - Homepage com sessão iniciada



GuguDadah - Login

← → ↻ <http://gugudadah.com/login>

## Login

Email ou Username
Password

Login

© GuguDadah 2018

Figura 29 – Login

GuguDadah - Registrar

← → ↻

<http://gugudadah.com/register>

# Registrar

Username
Nome
Email
Contacto
Password
Confirmar Password

Registrar

© GuguDadah 2018

Figura 30 – Registo

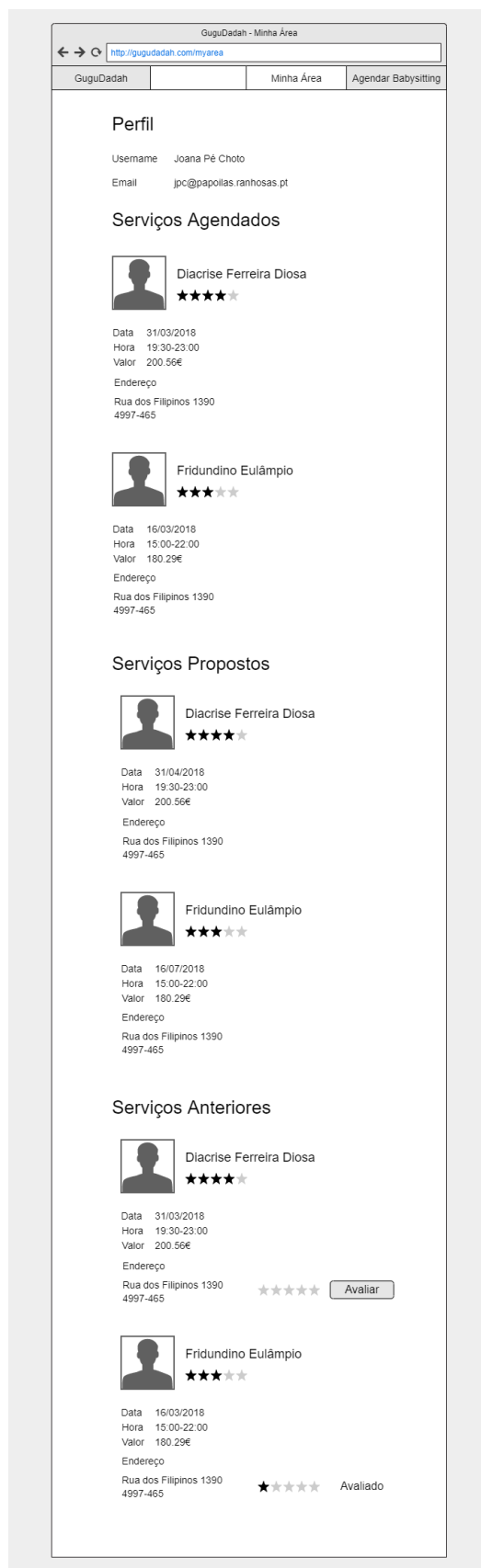


Figura 31 - Cliente - minha área

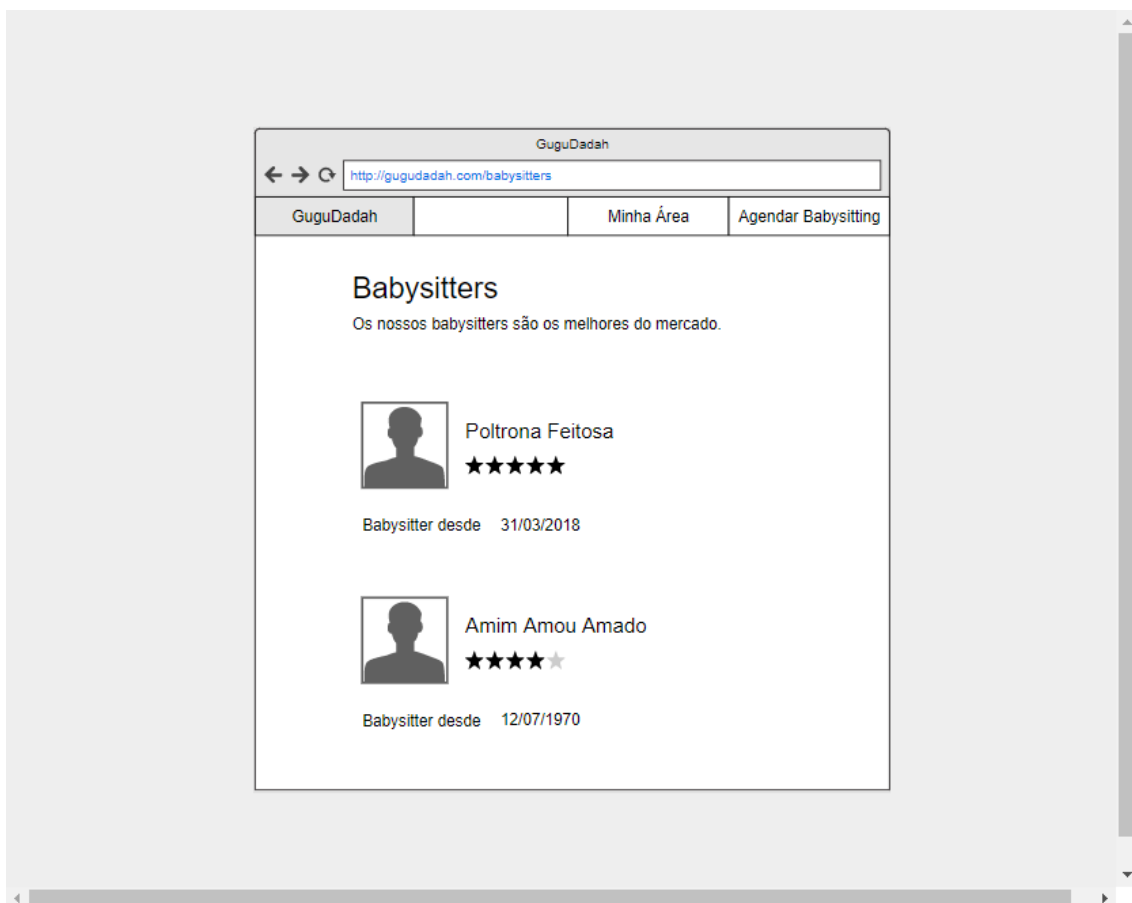


Figura 32 - Cliente – Babysitters

GuguDadah - Agendar - Passo 1

← → ↻

http://gugudadah.com/schedule/stepone

GuguDadah

Minha Área

Agendar Babysitting

## Agendar Babysitting

Data

4/22/2012



Rua

Rua dos arbustos 30

Localidade

Mirandela

Hora de início

09:00

▼

Hora de fim

13:00

▼

### Serviços extras

☒ Acompanhamento ao estudo

☒ Atividade lúdica no exterior

### Observações

O meu filho é muito carinhoso.

Continuar

Figura 33 - Cliente - Agendar passo 1

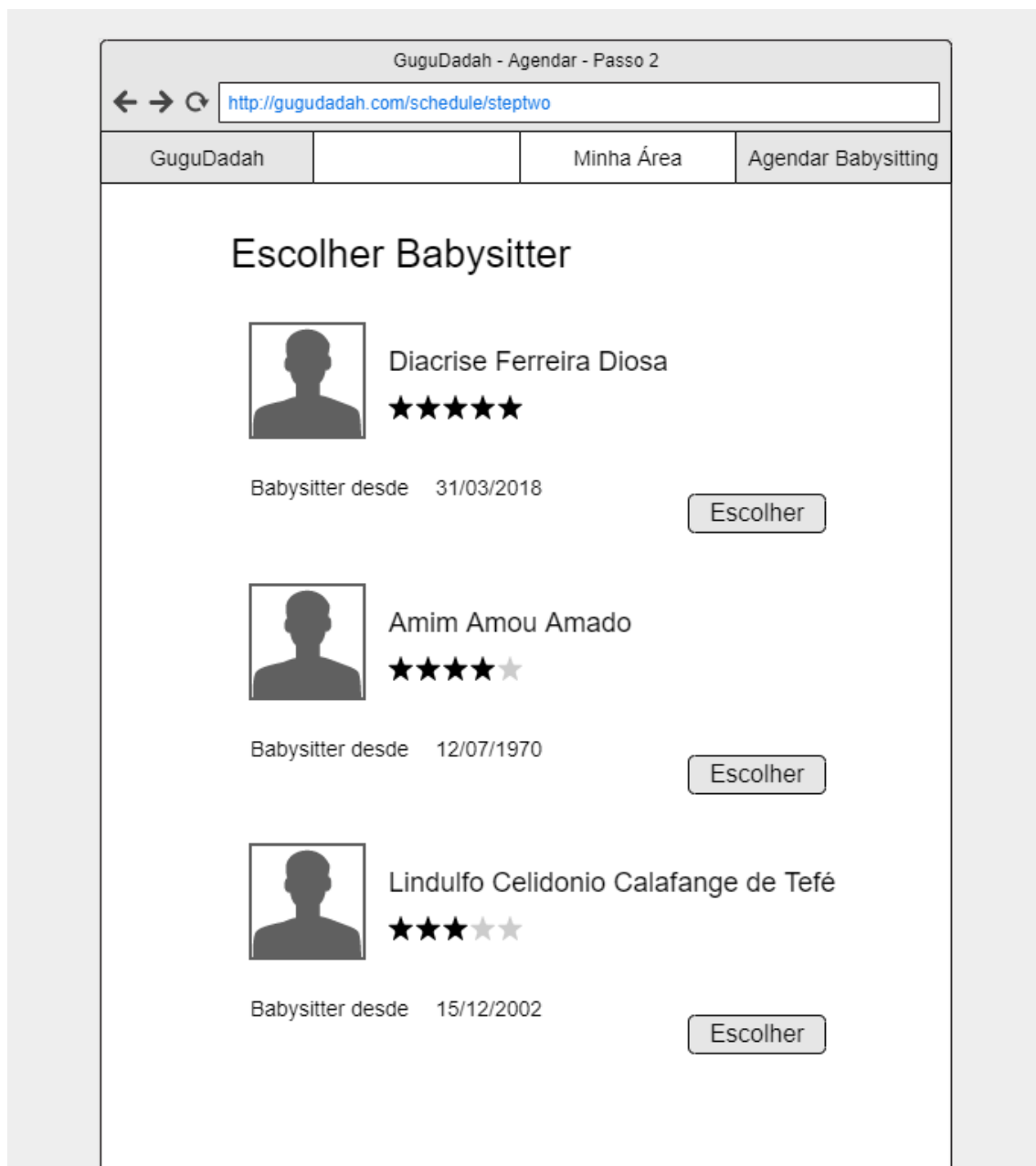


Figura 34 - Cliente - Agendar passo 2

The screenshot displays a web browser window titled "GuguDadah - Agendar - Passo 3". The address bar shows the URL "http://gugudadah.com/schedule/stepthree". Below the browser window is a navigation bar with three links: "GuguDadah", "Minha Área", and "Agendar Babysitting". The main content area is titled "Pagamento" and displays the following information:

- Valor 230€
- Modalidade de Pagamento
  - ☐ Multibanco
  - ☐ PayPal
  - ☒ MBWAY

At the bottom right of the payment section are two buttons: "Cancelar" and "Finalizar".

Figura 35 - - Cliente - Agendar passo 3

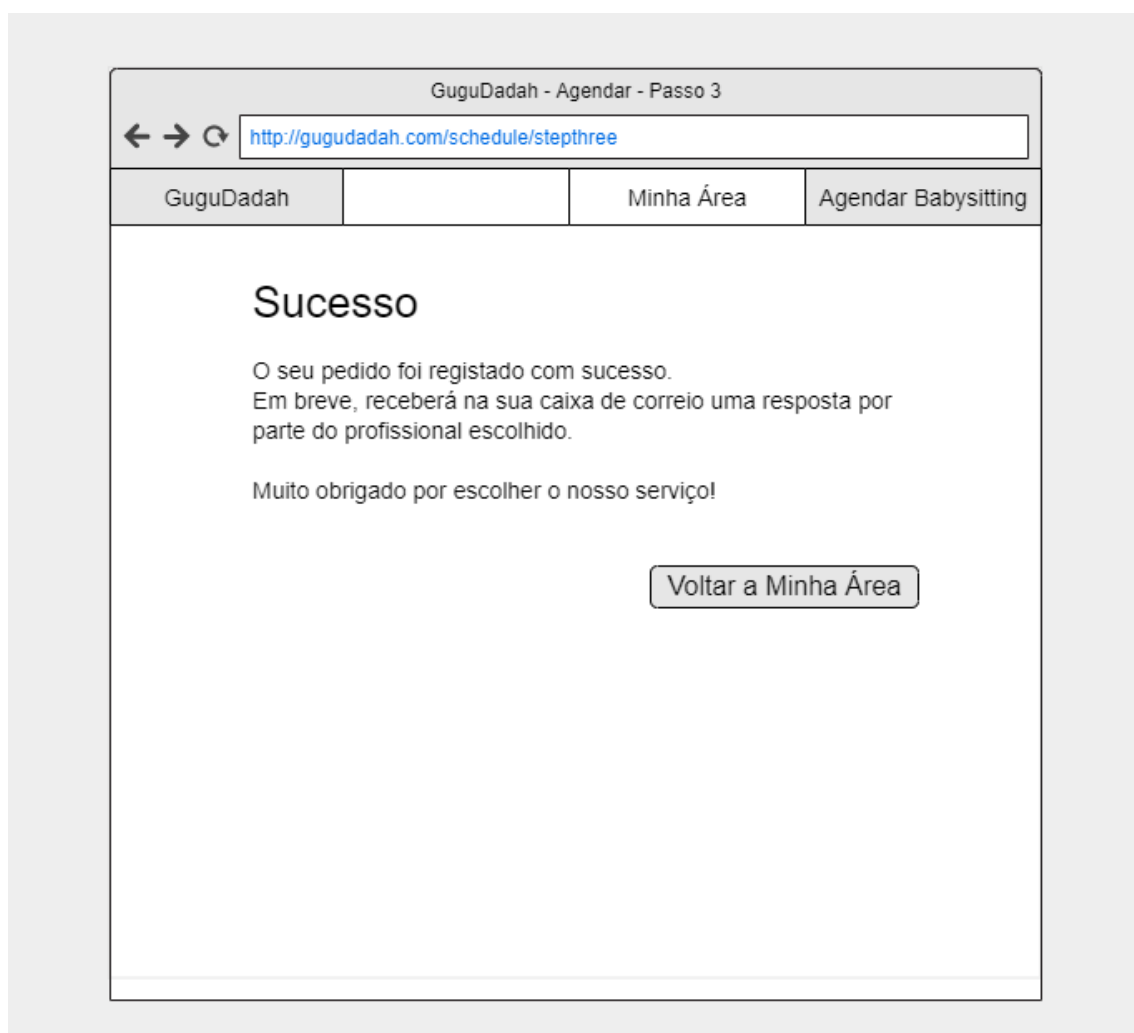


Figura 36 - Cliente - Agendar passo 4



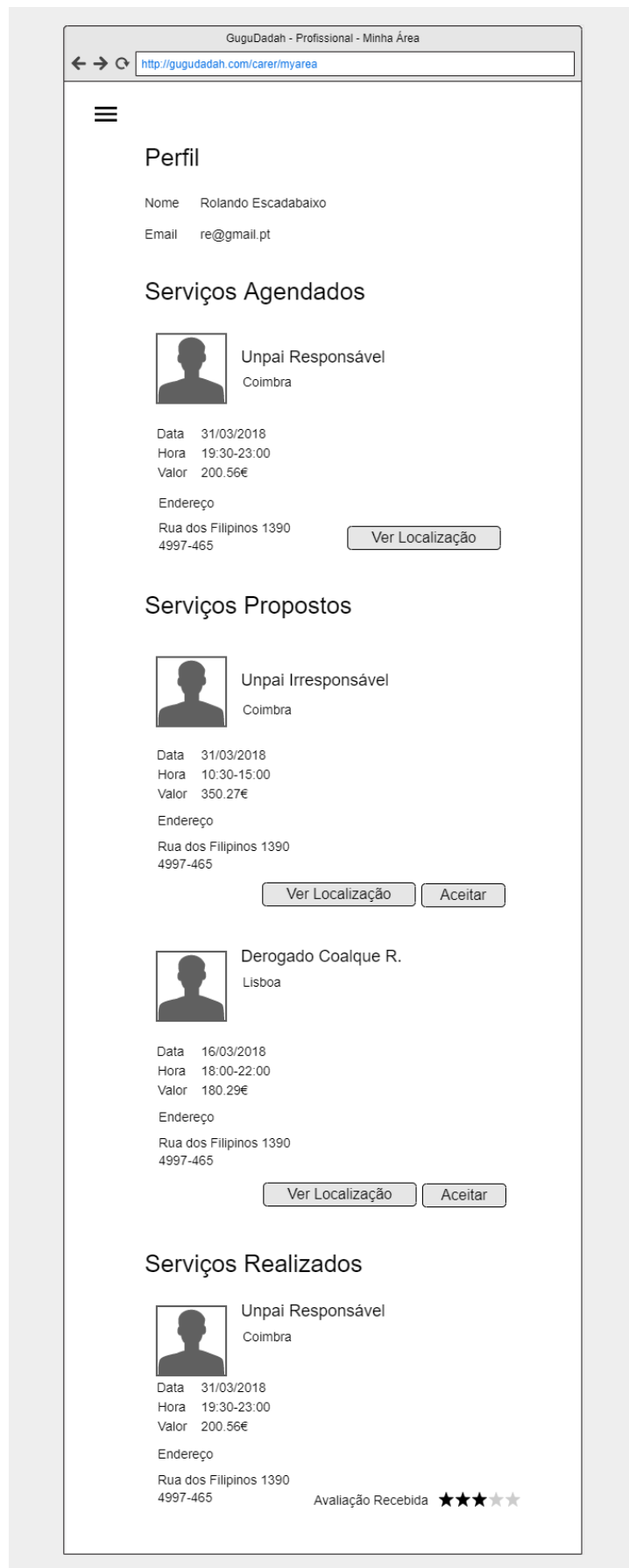


Figura 37 - Profissional - Minha área

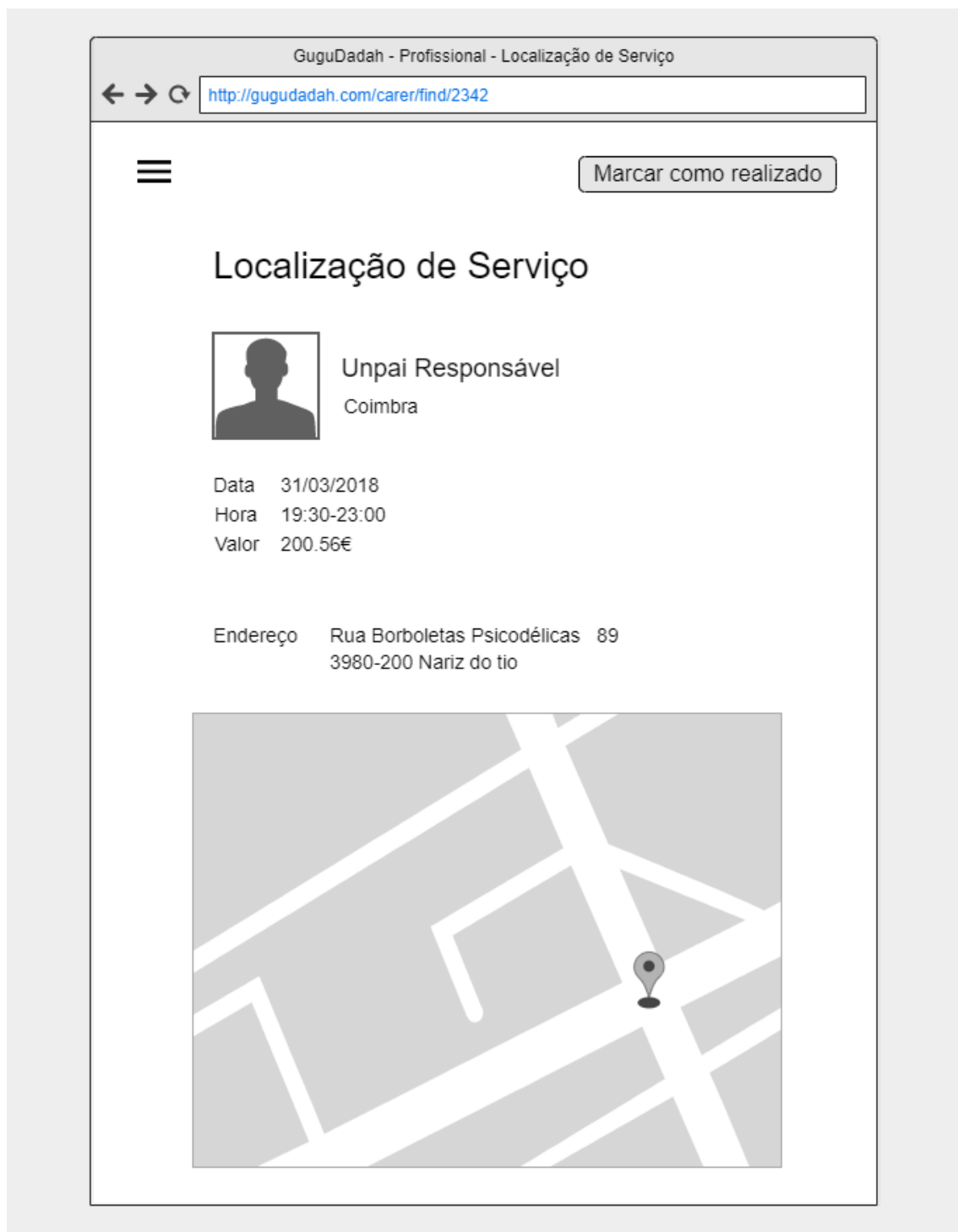


Figura 38 - Profissional - Ver localização

The image shows a web browser window with the title "GuguDadah - Registrar profesional". The address bar contains the URL "http://gugudadah.com/registerbabysitter". The main content area has the heading "Registrar Profesional" and a form with the following fields: "Username", "Email", "Contacto Telefónico", "Password", "Confirmar Password", "Turno", and "Avatar". Below these fields is a "Registrar" button. At the bottom of the form, it says "Copyright 2018 GuguDadah".

GuguDadah - Registrar profesional

← → ↻ <http://gugudadah.com/registerbabysitter>

## Registrar Profesional

Username

Email

Contacto Telefónico

Password

Confirmar Password

Turno

Avatar

Registrar

Copyright 2018 GuguDadah

Figura 39 - Admin - Registrar profesional

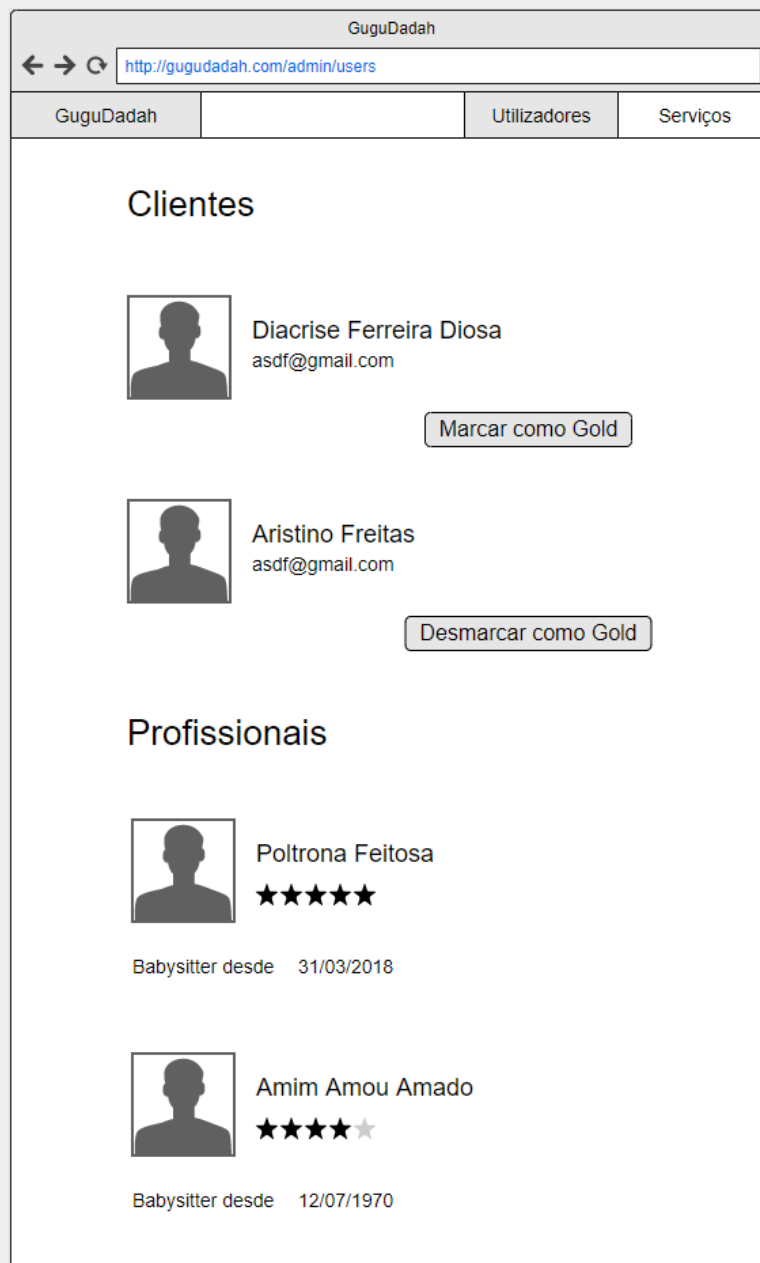



Figura 40 - Admin - utilizadores do Sistema

GuguDadah

<http://gugudadah.com/admin/jobs>

GuguDadah
Utilizadores
Serviços


### Serviços Agendados



Unpai Responsável  
Coimbra


Data 31/03/2018  
Hora 19:30-23:00  
Valor 200.56€  
Babysitter Toucinho Delgado  
Endereço  
Rua dos Filipinos 1390  
4997-465

### Serviços Propostos



Unpai Irresponsável  
Coimbra


Data 31/03/2018  
Hora 10:30-15:00  
Valor 350.27€  
Babysitter Toucinho Delgado  
Endereço  
Rua dos Filipinos 1390  
4997-465



Derogado Coalque R.  
Lisboa

Data 16/03/2018  
Hora 18:00-22:00  
Valor 180.29€  
Babysitter Toucinho Delgado  
Endereço  
Rua dos Filipinos 1390  
4997-465

### Serviços Realizados



Unpai Responsável  
Coimbra

Data 31/03/2018  
Hora 19:30-23:00  
Valor 200.56€  
Babysitter Toucinho Delgado  
Endereço  
Rua dos Filipinos 1390  
4997-465

Avaliação Recebida ★★★★★

Figura 41 - Admin - Lista geral de trabalhos

## 5 Conclusões e Trabalho Futuro

A ideia de criar um serviço ao domicílio que funcionasse 24/365, deixando ao critério do grupo a escolha do tema e todo o trabalho envolvido na criação e fundamentação do mesmo, foi, sem dúvida, desafiante. Foi possível constatar que todo o esforço envolvido aquando do surgimento e definição de uma ideia exige tanto ou mais trabalho quanto a pô-la em prática. Na primeira fase, apenas foi necessário elaborar um modelo geral da ideia a desenvolver, esclarecer alguns pontos e medidas de sucesso e viabilidade, e analisar os recursos necessários associados ao serviço. Ainda assim, sem uma boa fundamentação e sem o estabelecimento de metas, todos os passos futuros acabariam por levantar demasiadas dúvidas e poderiam acabar por divergir, tanto da ideia inicial, como entre os elementos do grupo.

Já na segunda fase, foi requerido que fossem especificados os requisitos de utilizador e de sistema, elaborados diversos diagramas, construída uma base de dados e, por fim, desenhados os *mockups* para a aplicação. O grande desafio consistiu na descrição completa e detalhada de todos os requisitos, uma vez que estes são a base de todo o sistema. O grupo viu-se, com o passar do tempo, a refazer e a aperfeiçoar este tópico. Uma vez estes concluídos, foram elaborados diagramas, sendo que a equipa optou pelos de mais alto nível, para uma melhor compreensão da parte do utilizador. Tanto a BD como os *mockups* fluíram naturalmente, uma vez que já havia uma boa base para o projeto.

Assim sendo, a equipa pretende, finalmente e numa última fase, iniciar a implementação da aplicação em si, juntamente com a plataforma móvel. O grupo espera conseguir finalizar as próximas etapas e tirar proveito das mesmas, levando consigo conhecimentos úteis para a vida profissional futura.

## Lista de Siglas e Acrónimos

<b>BD</b>	Base de Dados
<b>RNF</b>	Requisitos não funcionais
<b>UML</b>	Unified Modeling Language