



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2017/2018

“Cleanica” – Sistema de prestação de serviços de limpeza ao domicílio

Carlos Pedrosa – a77320
David Sousa - a78938
Daniel Vieira – a73974
Manuel Sousa – a78869
Rodrigo Ferreira - a74216

Março, 2018

Data de Recepção	
Responsável	
Avaliação	
Observações	

“Cleanica” – Sistema de prestação de serviços de limpeza ao domicílio

Carlos Pedrosa – a77320
 David Sousa - a78938
 Daniel Vieira – a73974
 Manuel Sousa – a78869
 Rodrigo Ferreira - a74216

Março, 2018

Resumo

Numa primeira fase, começamos por definir qual o objetivo deste trabalho e qual o tema que nos propusemos a desenvolver. Neste sentido, depois de feita a contextualização do projeto, justificamos o enquadramento da nossa ideia. Através da análise do mercado ficamos a perceber quem seriam os principais utilizadores, quais os seus interesses, e quais as propostas já existentes no meio informático. Para além disso, tivemos ainda em consideração o esforço tido por cada um dos elementos do grupo para que, deste modo, numa fase final seja possível estimar o valor monetário do projeto. Por fim, o presente trabalho contém ainda uma maquete onde se mostra um esquema que a aplicação final seguirá. Deste modo, procuramos percorrer todas as etapas que consideramos necessárias para que o trabalho seja o mais conciso, metódico e bem fundamentado possível.

Tendo sido concluída a fase de contextualização, debruça-se agora sobre toda a modelação necessária para a posterior implementação do projeto.

Assim, primeiramente levantamos os requisitos que resultaram do contacto com a empresa contratante, Cleanica. Seguidamente passamos pela especificação UML, criando os diagramas que consideramos necessários. Numa fase final, a modelação recaiu na criação de uma base de dados que servirá de alicerce para toda a aplicação. Deste modo, procuramos percorrer todas as etapas que consideramos necessárias para que o trabalho seja o mais conciso, metódico e bem fundamentado possível.

Área de Aplicação: Desenvolvimento de software para sistemas reais.

Palavras-Chave: *Software*, limpeza, modelação, base de dados relacional, modelo concetual, modelo lógico e SQL Server.

Índice

1. Fundamentação	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objetivos	2
1.4. Estrutura do Relatório	2
1.5. Definição da identidade do sistema a desenvolver	3
1.6. Justificação, viabilidade e utilidade do sistema	4
1.7. Identificação dos recursos necessários	5
1.7.1. Recursos de hardware	5
1.7.2. Recursos de rede	5
1.7.3. Recursos humanos	5
1.7.4. Recursos materiais	6
1.8. Modelo do sistema a implementar - Maqueta	6
1.9. Definição de medidas de sucesso	7
1.10. Plano de desenvolvimento	7
2. Especificação	8
2.1. Levantamento de Requisitos	8
2.1.1. Método de levantamento e análise de requisitos adotado	8
2.1.2. Requisitos levantados e respetiva análise	8
2.2. Modelação do Software	14
2.2. Modelo de domínio	14
2.3. Use Cases	15
2.4. Especificação dos Use Cases	17
➤ Administrador	17
2.4.1. Registrar novo funcionário	17
2.4.2. Remover funcionário	18
2.5. Diagramas de Sequência de Sistema	31
➤ Administrador	31
2.5.1. Registrar novo funcionário	31
2.5.2. Alterar dados de serviço	32
➤ Cliente	33

2.5.3.	Preencher formulário de satisfação	33
2.5.4.	Requisitar serviços	35
➤	Funcionário	36
2.5.5.	Confirmar realização de serviço	36
2.5.6.	Consultar calendário de serviços	37
2.6.	Diagrama de Classes	38
2.7.	Diagrama de Atividade	39
2.7.1.	Diagrama de atividade Administrador	40
2.7.2.	Diagrama de atividade Cliente	41
2.7.3.	Diagrama de atividade Funcionário	42
2.8.	Mockup da interface	43
2.8.1.	Administrador	44
2.8.2.	Cliente	46
2.8.3.	Funcionário	49
2.9.	Base de dados	50
2.9.1.	Identificação e caracterização das entidades	50
2.9.2.	Identificação e caracterização dos relacionamentos	51
2.9.3.	Identificação e caracterização das associações dos atributos com as entidades e relacionamentos	51
2.10.	Modelo lógico	55
2.11.	Modelo físico	57
3.	Implementação	58
3.1.	Apresentação e caracterização geral da arquitetura do sistema	58
3.2.	Descrição dos diversos módulos	62
3.3.	Plano de desenvolvimento e distribuição do trabalho	63
3.4.	Ferramentas utilizadas e implementação das várias funcionalidades	64
3.5.	Validação do software desenvolvido	65
4.	Conclusões e perspectivas de trabalho futuro	66
 Anexos		
I.	Anexo 1 – Entrevista a “Cleanica”	70
II.	Anexo 2 – Tabela de preços	72
III.	Diagramas de Sequência	73
IV.	Modelo Conceitual	85
V.	Modelo físico no SQL Server	86
VI.	Arquitetura do sistema	89

Índice de Figuras

Figura 1 - Logótipo da empresa	3
Figura 2 - Estudo obtido através da Consumer Barometer with Google	4
Figura 3 - Diagrama de Gantt	7
Figura 4 - Modelo de domínio	14
Figura 5 - Diagrama de Use Cases gerais	15
Figura 6 - Subdiagrama administrador	16
Figura 7 - Especificação do Use Case Registrar funcionário	17
Figura 8 - Especificação do Use Case Remover funcionário	18
Figura 9 - Especificação do Use Case Alterar dados de serviço	19
Figura 10 - Especificação do Use Case Consultar formulários de satisfação	19
Figura 11 - Especificação do Use Case Efetuar registo	20
Figura 12 - Especificação do Use Case Consultar serviços	20
Figura 13 - Especificação do Use Case Preencher formulários de satisfação	21
Figura 14 - Especificação do Use Case Consultar histórico	21
Figura 15 - Especificação do Use Case Requisitar serviço de lavagem de automóvel	22
Figura 16 - Especificação do Use Case Requisitar serviço de jardinagem	23
Figura 17 - Especificação do Use Case Requisitar serviço de jardinagem	24
Figura 18 - Especificação do Use Case Requisitar serviço de limpeza de interiores	25
Figura 19 - Especificação do Use Case Efetuar Pagamento	26
Figura 20 - Especificação do Use Case Escolher funcionário(s) para o serviço	26
Figura 21 - Especificação do Use Case Gerar fatura	27
Figura 22 - Especificação do Use Case Alterar dados	27
Figura 23 - Especificação do Use Case Consultar calendário de serviços	28
Figura 24 - Especificação do Use Case Consultar histórico de encargos	29
Figura 25 - Especificação do Use Case Confirmar realização do serviço	29
Figura 26 - Especificação do Use Case Pedir direções	30
Figura 27 - Especificação do Use Case Alterar dados funcionário	30
Figura 28 - Diagrama de sequência: registar novo funcionário	31
Figura 29 - Especificação do diagrama de sequência: registar funcionário	32

Figura 30 - Diagrama de sequência: alterar dados de serviço	32
Figura 31 - Especificação do diagrama: alterar dados de serviço	33
Figura 32 - Diagrama de sequência: preencher formulário de satisfação	33
Figura 33 - Especificação do diagrama de sequência: preencher formulário de satisfação	34
Figura 34 - Diagrama de sequência: requisitar serviço	35
Figura 35 - Especificação do diagrama de sequências: requisitar serviço	36
Figura 36 - Diagrama de sequência: confirmar realização de serviço	36
Figura 37 - Especificação do diagrama de sequência: confirmar serviço	37
Figura 38 - Diagrama de sequência: consultar calendário de serviços	37
Figura 39 - Especificação do diagrama de sequência: consultar calendário de serviços	37
Figura 40 – Diagrama de classes	38
Figura 41 - Diagrama de Atividade Administrador	40
Figura 42 - Diagrama de Atividade Cliente	41
Figura 43 - Diagrama de Atividade Funcionário	42
Figura 44 - Mockup Login	43
Figura 45 - Mockup Menu principal Administrador	44
Figura 46 - Mockup Alterar serviço	44
Figura 47 - Mockup Formulários de satisfação	45
Figura 48 - Mockup Novo Funcionário	45
Figura 49 - Mockup Remover funcionário	46
Figura 50 - Mockup Menu principal Cliente	46
Figura 51 - Mockup Consultar serviço	47
Figura 52 - Mockup Requisitar serviço lavagem automóvel	47
Figura 53 - Mockup Requisitar serviço, interiores, exteriores e piscina	48
Figura 54 - Mockup Alterar dados (cliente)	48
Figura 55 - Mockup Menu principal funcionário	49
Figura 56 - Modelo lógico	55
Figura 57 - Implementação do menu de Login	58
Figura 58 - Implementação do menu Alterar Dados de Serviço	59
Figura 59 - Implementação do menu Remover funcionário	59
Figura 60 - Implementação do menu de alterar dados	60
Figura 61 - Implementação do menu requisitar serviço	60
Figura 62 - Implementação do menu Home	61
Figura 63 - Divisão do trabalho segundo Model-View-Controller	62
Figura 64 - Planeamento da terceira etapa	63
Figura 65 - Planeamento da primeira e segunda etapa	63
Figura 66 - Esquematização da requisição de um serviço	65

Figura 67 - Diagrama de sequência (Administrador): consultar formulário de satisfação	73
Figura 68 - Especificação do diagrama de sequência (Administrador): consultar formulário de satisfação	73
Figura 69 - Diagrama de sequência (Administrador): remover funcionário	74
Figura 70 - Especificação do diagrama de sequência (Administrador): remover funcionário	74
Figura 71 - Diagrama de sequência (cliente): alterar dados	75
Figura 72 - Especificação do diagrama de sequência (cliente): alterar dados	75
Figura 73 - Diagrama de sequência (cliente): consultar histórico	75
Figura 74 - Especificação do diagrama de sequência (cliente): consultar histórico	76
Figura 75 - Diagrama de sequência (cliente): consultar serviço	76
Figura 76 - Especificação do diagrama de sequência (cliente): consultar serviço	76
Figura 77 - Diagrama de sequência: efetuar Login	77
Figura 78 - Especificação do diagrama de sequência: efetuar Login	77
Figura 79 - Diagrama de sequência: efetuar pagamento	78
Figura 80 - Especificação do diagrama de sequência: efetuar pagamento	79
Figura 81 - Diagrama de sequência: escolher funcionário(s)	79
Figura 82 - Especificação do diagrama de sequência: escolher funcionário(s)	80
Figura 83 - Diagrama de sequência (funcionário): alterar dados	80
Figura 84 - Especificação do diagrama de sequência (funcionário): consultar serviço	81
Figura 85 - Diagrama de sequência (funcionário): consultar histórico de encargos	81
Figura 86 - Especificação do diagrama de sequência (funcionário): consultar histórico de encargos	82
Figura 87 - Diagrama de sequência (funcionário): pedir direções	82
Figura 88 - Especificação do diagrama de sequência (funcionário): pedir direções	83
Figura 89 - Diagrama de sequência (funcionário): gerar fatura	83
Figura 90 - Especificação do diagrama de sequência (funcionário): gerar fatura	84
Figura 91 - Modelo Concetual	85
Figura 92 - Implementação do menu inicial (Administrador)	89
Figura 93 - Implementação do menu registrar novo funcionário (Administrador)	89
Figura 94 - Implementação do menu Formulários de Satisfação (Administrador)	90
Figura 95 - Implementação do menu consultar histórico (Cliente)	90
Figura 96 - Implementação do menu serviços disponíveis (Cliente)	91

Figura 97 - Implementação do menu tarefas pendentes (Cliente)	91
Figura 98 - Implementação do menu preencher formulário de realização (funcionário)	92
Figura 99 - Implementação do menu alterar dados (funcionário)	92
Figura 100 - Implementação do menu Histórico de Serviços (funcionário)	93

Índice de Tabelas

Tabela 1 - Identificação das entidades	50
Tabela 2 - Caracterização dos relacionamentos	51
Tabela 3 - Identificação e associação dos atributos com as entidades	54
Tabela 4 - Tabela de preços disponibilizada pela empresa	72

1. Fundamentação

No âmbito da unidade curricular de Laboratórios de Informática IV foi-nos proposto o desenvolvimento de uma aplicação que permita dar resposta a serviços ao domicílio. Neste seguimento, o tema escolhido passa pela implementação de um serviço de limpeza que atuará no lar de quem o requisitar.

Assim, pretendemos demonstrar de seguida uma divisão por capítulos que farão alusão às diversas partes que o trabalho contemplará. Deste modo, num primeiro momento a fundamentação do trabalho vigorará, sendo que pretendemos contextualizar e justificar o modelo em termos de negócio. Pretendemos ainda especificar, numa fase posterior, todos os esforços levados a cabo para a realização do trabalho. Por fim, o objetivo passa por demonstrar a implementação que resultará do cumprimento das fases já enunciadas.

1.1. Contextualização

A “Cleanica” é uma empresa embrionária relacionada com serviços de limpeza ao domicílio. De facto, trata-se de uma empresa criada por um grupo de jovens que viram na limpeza uma ótima oportunidade de negócio. Apesar de ser um negócio que se encontra num estado inicial, a companhia já conta com inúmeros clientes que todos os dias telefonam para a marcação de um ou mais serviços de limpeza que esta contempla.

No entanto, preocupada com a concorrência neste ramo e, visto que, apenas consegue dar resposta aos pedidos através do contacto telefónico, a empresa decidiu mais uma vez arriscar e investir na criação de um sistema diferente neste tipo de serviços em Portugal. O sistema, consistiria na implementação de uma aplicação e/ou um serviço Web que passaria por facilitar a interação entre cliente-profissional. No entanto, não sendo uma ideia fácil de implementar, a companhia manifestou interesse em estabelecer uma parceria com o nosso grupo que se foca no desenvolvimento de sistemas de software. Assim, estando disposta a novas ideias, não colocaria qualquer entrave nas propostas apresentadas pelo grupo.

1.2. Apresentação do Caso de Estudo

O projeto consiste na implementação de uma aplicação que se designará por “Cleanica”. Esta aplicação suportará o ciclo completo de prestação de serviços de limpeza ao domicílio pelo que será capaz de dar resposta aos mais variados requisitos. Nesse sentido, o utilizador poderá solicitar um dos serviços de limpeza disponível, no horário que achar mais conveniente e por fim, poderá ainda pagar de acordo com o serviço prestado.

1.3. Motivação e Objetivos

Num mundo em que cada vez mais as pessoas têm menos tempo, é nas pequenas tarefas semanais que podemos acumular algumas horas. O crescente uso das novas tecnologias, serviram como motivação para a tomada de decisão da “Cleanica”. Assim sendo, esta propôs a criação de uma aplicação que permite facilitar a interação da empresa com as pessoas que anteriormente já utilizavam os serviços desta, evitando assim, fazer os serviços de limpeza que hoje em dia, não querem, ou têm tempo para os fazer. É também um desafio novo na área da engenharia de software, pois vamos usar ferramentas que não nos são familiares, o que permitirá enriquecer o nosso portfólio com mais uma panóplia de tecnologias.

Este ambicioso projeto vai permitir aos donos de casa solicitar os serviços de um ou vários profissionais, através do preenchimento de um formulário web, com pré-pagamento e com total confidencialidade e segurança, algo que hoje em dia, nem sempre é possível. Estes profissionais, garantidos pela “Cleanica”, realizarão serviços que são essenciais, porém são também dispendiosos em termos temporais. Com a terceirização destas tarefas, o cliente pode gastar o seu tempo de forma mais proveitosa, quer seja em trabalho, em lazer ou até mesmo em descanso.

1.4. Estrutura do Relatório

Depois de explicitar o contexto do projeto, bem como apresentar quais as motivações e quais os objetivos do mesmo segue-se agora a definição do sistema a desenvolver, a sua viabilidade e os recursos necessários para levar a cabo a ideia apresentada. Para além disso, é nos próximos capítulos que pretendemos demonstrar a maquete desenvolvida, as medidas de sucesso e o plano de desenvolvimento que tem por base o tempo dispendido na realização do trabalho.

1.5. Definição da identidade do sistema a desenvolver

Nome: “Cleanica”

Categoria: Serviços ao domicílio

Descrição: O funcionamento da aplicação consiste em realizar uma procura de serviços de limpezas ao domicílio, serviços estes que serão garantidos pela empresa contratante. Esta ficará assim encarregue de inserir na aplicação cada um dos seus funcionários. No início de cada dia de trabalho, o sistema enviará para cada funcionário uma notificação com as tarefas que este terá de realizar. Do lado do cliente, este também terá de estar registado para que efetue login na aplicação. Depois disso, poderá escolher os serviços de que pretende usufruir, definir a data e hora pretendida, indicar o número de metros quadrados da área a ser intervencionada bem como algumas informações que considere relevantes para a realização do serviço. Efetivamente, assim que forem inseridas todas as informações necessárias pelo cliente, o sistema verifica o colaborador disponível que se encontra mais próximo do local do serviço. No final, cada cliente poderá pagar através da aplicação e posteriormente poderá ainda classificar quem “contratou” com base no trabalho que fora executado na sua casa.

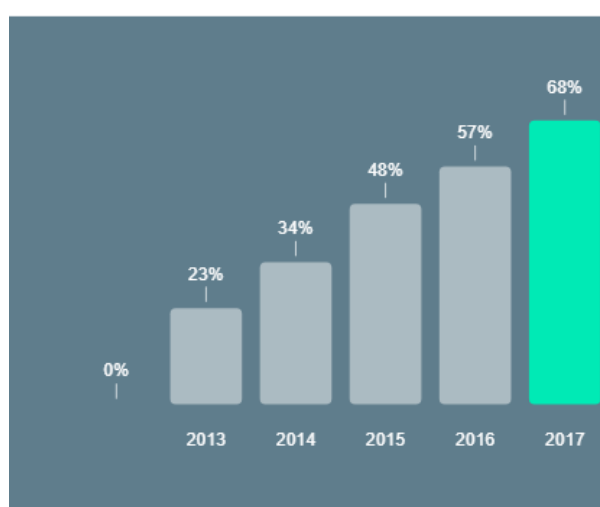
Características: Histórico de utilização, serviços de localização, avaliação de trabalho.



Figura 1 - Logótipo da empresa

1.6. Justificação, viabilidade e utilidade do sistema

Cada vez mais as pessoas recorrem a empregados de limpeza para que estes se ocupem das mais variadas tarefas domésticas¹. No entanto, os serviços pontuais são mais recorrentes e funcionam de forma a permitir ao cliente economizar tempo que considere precioso. Paralelamente, um estudo conduzido pela Google, revela que a utilização de *smartphones* e de tecnologias web tem vindo a crescer. Nesse sentido, torna-se viável a implementação de uma aplicação que conjuga estes dois tipos de serviços. Só seria necessário responder a um pequeno formulário e em questões de minutos cada utilizador iria encontrar a solução que preenchesse os seus requisitos.



Percentage of people who access the internet at least as often via smartphone as computer.

Figura 2 - Estudo obtido através da *Consumer Barometer with Google*

Deste modo, o sistema em questão permitirá uma maior facilidade de contratar colaboradores pertencentes a uma empresa, para efetuar serviços de limpeza ao domicílio. Isto é possível devido ao facto de se tratar de uma empresa dinâmica, que admite trabalhadores de vários locais de acordo com a área que considere alcançável, bastando para isso ter efetuado o registo e ser aceite pela empresa, não necessitando da sua presença física em nenhum local em concreto. Deste modo, a empresa reduz os custos fixos (como aluguer de edifícios e equipamentos, salários de administração, segurança e vigilância, entre outros). Deixa de ser necessário uma pesquisa de

¹ Segundo "Por uma sociologia do trabalho doméstico: contribuição para um projecto interdisciplinar", contemplado ainda na bibliografia.

peessoas com competências, por parte dos clientes para os realizar, visto que a qualidade dos colaboradores é assegurada pela empresa. Após a realização dos serviços, o cliente poderá avaliar o trabalhador e, através disto, a empresa poderá ajustar-se de forma a garantir a qualidade máxima dos seus serviços. Em relação a estes a empresa proporcionará também a vantagem de poder contar com trabalhadores em serviço nos mais variados horários (consoante a sua disponibilidade), não existindo uma hora limite para o seu funcionamento. Por forma a facilitar o pagamento a companhia assegura este, permitindo que seja feito através da aplicação (cartão de crédito, por exemplo).

1.7. Identificação dos recursos necessários

1.7.1. Recursos de hardware

Sendo uma aplicação web/móvel como recursos hardware apenas são necessários aparelhos que permitam uma ligação internet. Não são requeridos modelos muito sofisticados, pois será uma aplicação simples, acessível aos mais variados utilizadores. No entanto, para que se reduzam os custos de transporte, é necessário saber qual o trabalhador que se encontra mais próximo do local sujeito à intervenção, assim como o trajeto mais curto até ao mesmo. Para isto, é necessário um serviço de GPS.

1.7.2. Recursos de rede

Como a aplicação funcionará com base em vários recursos obtidos através da internet é essencial que os clientes possuam *gadgets* que têm acesso a este tipo de serviços.

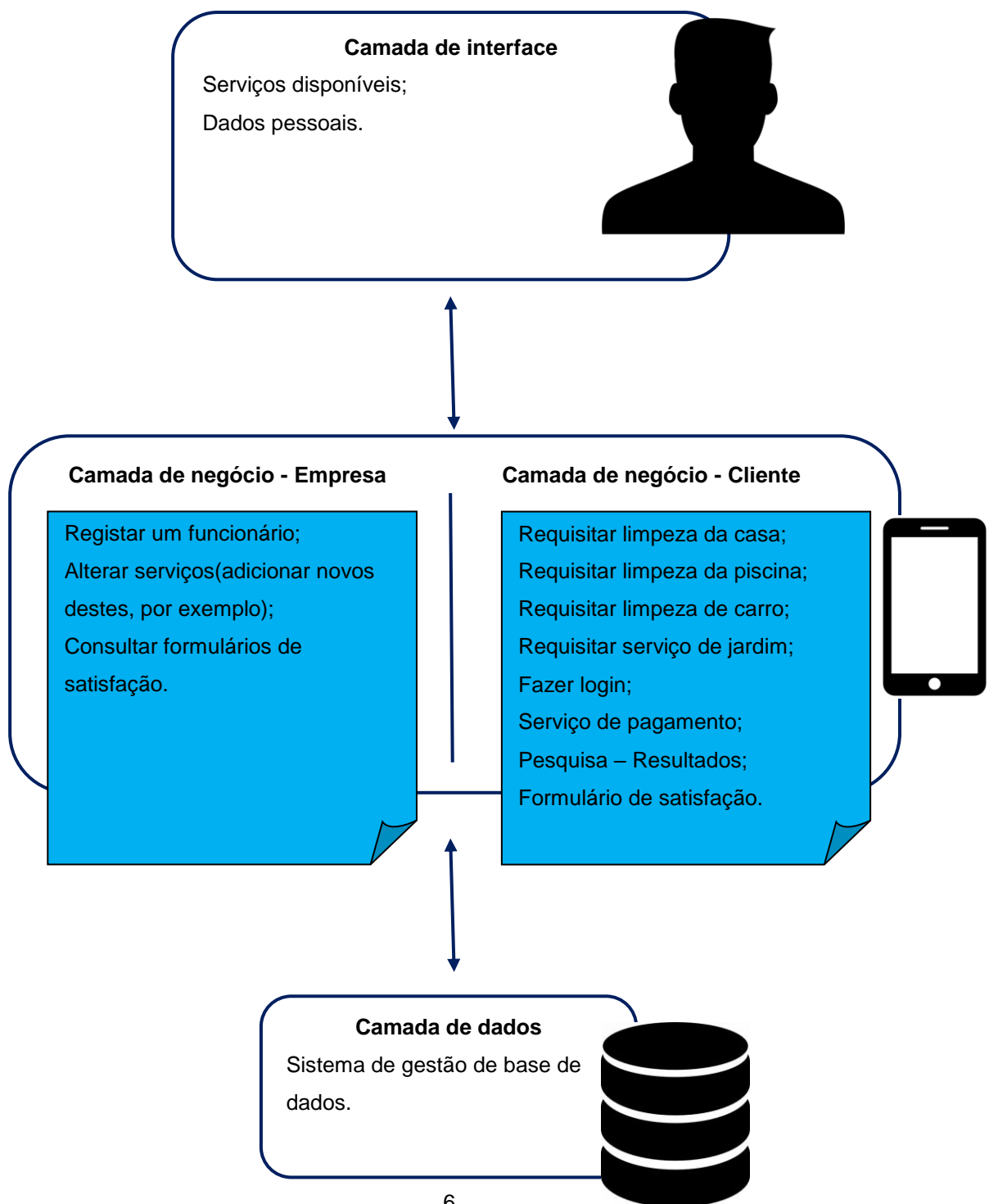
1.7.3. Recursos humanos

Os recursos humanos serão garantidos pela empresa contratada. Esta terá de registar os funcionários que aí trabalham para que estes possam, posteriormente, estar ao serviço da empresa. Sempre que for contratado um novo empregado de limpeza, é da inteira responsabilidade de “Cleanica” adicioná-lo à aplicação.

1.7.4. Recursos materiais

Como recursos materiais podemos enumerar os produtos de limpeza necessários para os mais variados serviços. Estes serão ajustados de acordo com a opção tomada pelos clientes, solicitá-los ou não.

1.8. Modelo do sistema a implementar - Maqueta



1.9. Definição de medidas de sucesso

Antes de qualquer tipo de modelação e/ou projeção da ideia em vigor, o principal objetivo do grupo é o de finalizar o produto que foi requerido, bem como o bom funcionamento do mesmo. De facto, uma primeira abordagem para a principal medida de sucesso seria a de lançar o produto final consoante o prazo estabelecido e com o mínimo de falhas e irregularidades possível.

A satisfação da empresa em relação ao produto final será também uma medida de sucesso fulcral para que este seja exposto no mercado. Visto que todo este processo foi planeado e bem estruturado, a equipa pretende um produto final homogéneo àquilo que foi previamente modelado, e que funcione da maneira prevista.

Posto isto, é necessário que os utilizadores (que são considerados como público alvo) se identifiquem com os diversos componentes e funcionalidades da aplicação. A adesão por parte dos mesmos significará que todo o estudo feito sobre o mercado do ramo de limpeza tal como a análise aos utilizadores deste tipo de tecnologias fosse o expectável.

1.10. Plano de desenvolvimento

Por forma a fazer um planeamento do trabalho que será efetuado ao longo da elaboração do projeto, decidimos criar um diagrama de Gantt com o auxílio da ferramenta da Microsoft, "Project". Através do diagrama é possível ver o tempo que será dedicado a cada uma das tarefas para que o trabalho seja concluído no prazo estipulado.

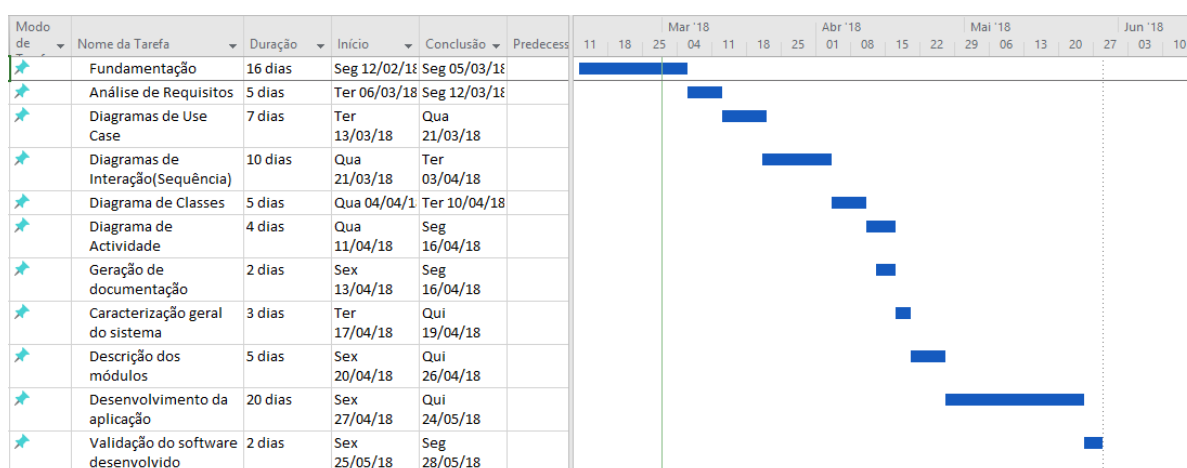


Figura 3 - Diagrama de Gantt

2. Especificação

2.1. Levantamento de Requisitos

2.1.1. Método de levantamento e análise de requisitos adotado

No sentido a identificar os requisitos necessários para a elaboração do projeto, decidimos reunir com o responsável da empresa “Cleanica”. Desta forma, pretendíamos perceber quais eram os objetivos deste relativamente ao software que iria ser implementado. Optamos, para o efeito, por efetuar uma entrevista. No decorrer desta foi possível levantar requisitos que permitiriam combater as eventuais inconsistências que pudessem surgir.

2.1.2. Requisitos levantados e respetiva análise

2.1.2.1 Requisitos funcionais

➤ **Administrador:**

- **RF1:** *O administrador poderá fazer o login na plataforma.*

- 1.1. O administrador insere os dados de login, sendo estes, o email e password;
- 1.2. O sistema valida os dados deste, de acordo com o registo apresentado, previamente, na base de dados;
- 1.3. O sistema apresenta um menu especializado que permita realizar a gestão dos recursos da empresa.

- **RF2:** *O administrador poderá registar novos funcionários na plataforma.*

- 2.1. O sistema apresentará uma secção dedicada ao registo de funcionários onde serão indicados todos os dados importantes. Estes serão: Nome,

Data de Nascimento, Nº Cartão de cidadão, IBAN, telemóvel e por último a especialidade deste;

2.2. O sistema deverá emitir uma confirmação de registo.

- **RF3:** *O administrador deverá poder alterar serviços.*

3.1. O sistema deverá apresentar um menu com a opção: “Alterar serviços”;

3.2. O sistema deverá permitir alterar o preço e a descrição do serviço;

3.3. O sistema deverá confirmar a alteração do serviço e atualizar a base de dados.

- **RF4:** *O administrador poderá consultar os formulários de satisfação.*

4.1. O sistema deverá apresentar uma área relativa à consulta de formulários de satisfação;

4.2. O sistema apresentará todos os formulários relativos ao funcionário indicado ordenados por data ou nome.

➤ **Cliente:**

- **RF1:** *O cliente poderá registar-se na plataforma.*

1.1. O sistema deverá apresentar um formulário ao utilizador com os campos necessários para registo;

1.2. O sistema deverá pedir apenas: nome, email, password, telemóvel e morada;

1.3. Sendo concluído o registo, o sistema deverá guardar todas as informações referentes a este numa base de dados.

- **RF2:** *O cliente poderá efetuar login na plataforma.*

2.1. O sistema deverá apresentar campos para o utilizador introduzir as suas credencias;

2.2. O sistema deverá verificar se as credenciais inseridas se encontram válidas;

2.3. Sendo que se trata de um cliente, o sistema deverá apresentar o menu que se encontre em conformidade com este tipo de utilizador. Neste menu, deverão constar vários separadores, cada um de acordo com as funcionalidades que o cliente tem acesso.

- **RF3:** *O cliente poderá fazer uma consulta de serviços.*

- 3.1. O sistema deverá apresentar um separador onde se encontrem todos os serviços fornecidos pela empresa;
- 3.2. Relativo a cada serviço devem constar informações sobre os mesmos. Entre essas informações devem constar: nome do serviço, preços e uma pequena descrição

- **RF4:** *O cliente poderá aceder a um histórico dos serviços requisitados.*

- 4.1. O sistema deverá apresentar um separador com todas as informações referentes aos serviços que um cliente requisitou;
- 4.2. Os dados acedidos por cada cliente devem apenas dizer respeito a ele próprio;
- 4.3. O histórico apresentado conterá informação referente ao nome, preço, data e hora dos serviços que foram efetuados.

- **RF5:** *O cliente poderá alterar os seus dados pessoais.*

- 5.1. O sistema deverá apresentar uma área onde seja possível efetuar a alteração dos dados pessoais;
- 5.2. O sistema deverá permitir a alteração da informação relativa à morada, ao nome, telemóvel e à password;
- 5.3. O sistema deverá registar na base de dados todas as informações alteradas.

- **RF6:** *O cliente poderá selecionar um dos serviços de limpeza apresentados.*

- 6.1. O sistema deverá apresentar um menu com os serviços que poderão ser requisitados;
- 6.2. O sistema deverá permitir a seleção de apenas um serviço de cada vez;
- 6.3. O sistema deverá emitir um formulário assim que o serviço seja selecionado;
- 6.4. O sistema deverá apresentar, no formulário referido, a data e hora que se encontram disponíveis;
- 6.5. Caso o serviço selecionado se trate da limpeza de interiores, de exteriores, ou de limpeza de piscinas o formulário deverá ainda conter campos relativos a: área a ser intervencionada(m²) e se é necessário ou não o fornecimento de produtos de limpeza;

- 6.6. Caso o serviço selecionado se trate da limpeza de automóveis o formulário deverá variar quanto à forma como o tamanho é apresentado, substituindo, a área a ser intervencionada por tamanho do carro (três ou cinco portas);
- 6.7. Para além das informações indicadas no formulário o sistema deverá permitir ao cliente especificar algumas considerações relativas à limpeza que considere importante;
- 6.8. O sistema deverá registar o serviço solicitado no histórico do utilizador e consequentemente, na base de dados.

- ***RF7: O cliente deverá poder pagar aquando o requisito do pedido.***

- 7.1. O sistema deverá apresentar métodos de pagamento depois de ter sido concluída a solicitação de um serviço;
- 7.2. O sistema deverá validar os dados do pagamento;
- 7.3. O sistema deverá registar os dados relativos a este pagamento(preço) no histórico do cliente;
- 7.4. O sistema deverá emitir uma fatura, onde se comprova o pagamento, que deverá ser enviada para o email especificado no registo.

- ***RF8: O cliente deverá poder avaliar o serviço prestado.***

- 8.1. O sistema deverá emitir um formulário, após a data de marcação do serviço, em que se peça ao cliente a sua avaliação relativamente ao serviço que fora executado;
- 8.2. Este formulário deverá constar no separador referente a “tarefas pendentes”;
- 8.3. No formulário deverão estar algumas perguntas e respetivas opções relativa ao funcionário que realizou a limpeza. As opções apresentadas basear-se-ão no grau de satisfação do cliente.

➤ **Funcionário:**

- ***RF1: O funcionário poderá efetuar login na plataforma.***

- 1.1. O sistema deverá apresentar campos para o utilizador introduzir as suas credencias;
- 1.2. O sistema deverá verificar se as credencias inseridas se encontram válidas;
- 1.3. Sendo que se trata de um funcionário, o sistema deverá apresentar o menu que se encontre em conformidade deste tipo de utilizador. Neste menu,

deverão constar vários separadores, cada um de acordo com as funcionalidades que o cliente tem acesso.

- **RF2:** *O funcionário poderá consultar um calendário relativo aos trabalhos que terá de efetuar.*

- 2.1. O sistema apresentará um calendário que poderá ser consultado pelo funcionário;
- 2.2. Neste calendário, constarão as notificações acerca dos trabalhos a efetuar;
- 2.3. O sistema deverá marcar no calendário todos os serviços já realizados;
- 2.4. O sistema fará referência a todos os trabalhos que não foram ainda efetuados.

- **RF3:** *O funcionário poderá confirmar os serviços já efetuados.*

- 3.1. O sistema deverá apresentar uma área onde seja possível confirmar os serviços já realizados;
- 3.2. O sistema deverá emitir um formulário sobre o serviço e considerações a ter em conta;
- 3.3. O sistema deverá registar na base de dados todas as informações alteradas.

- **RF4:** *O funcionário poderá pedir indicações da morada onde irá trabalhar.*

- 4.1. O sistema deverá apresentar permitir o pedido de indicações;
- 4.2. O sistema deverá apresentar um mapa onde se apresenta o caminho mais curto para a morada indicada.

- **RF5:** *O funcionário poderá consultar um histórico com o seu caderno de encargos.*

- 5.1. O sistema apresentará um separador dedicado à consulta do histórico;
- 5.2. O sistema permitirá a consulta por datas ou por cliente.

- **RF6:** *O funcionário poderá alterar os seus dados pessoais.*

- 6.1. O sistema deverá apresentar uma área onde seja possível efetuar a alteração dos dados pessoais;
- 6.2. O sistema deverá permitir a alteração da informação relativa à morada, ao IBAN, ao nome, telemóvel e à password;

- 6.3. O sistema deverá registar na base de dados todas as informações alteradas.

2.2.1.2. Requisitos não funcionais

- **RNF1:** *O sistema deverá ser executado num aparelho com conexão à rede.*

Por forma a aceder aos diferentes recursos o sistema deverá possuir ligação à rede.

- **RNF2:** *O aparelho deverá possuir sistema de GPS.*

O cálculo da localização é crucial no funcionamento do sistema. Assim, para que um dispositivo possa utilizar esta funcionalidade, este deverá possuir GPS.

- **RNF3:** *O sistema deverá ser desenvolvido para a plataforma Windows.*

O dono de “Cleanica” requereu que a Microsoft fosse uma constante no desenvolvimento da aplicação desenvolvida. Sendo assim, um dos requisitos prende-se com a criação de um projeto que funcione no sistema alvo da empresa, o Windows.

- **RNF4:** *O sistema deverá proteger os dados dos diferentes utilizadores.*

A aplicação requer o acesso a serviços que necessitam de ser seguros, como é o caso dos serviços de pagamentos. Neste sentido, de forma a garantir a fiabilidade do sistema, é necessário que os dados dos utilizadores sejam encriptados.

- **RNF5:** *O sistema deverá ser capaz de atualizar a base de dados em tempo real.*

Muitos dos recursos apresentados na aplicação funcionam de forma dinâmica. É necessário, deste modo, que a base de dados seja atualizada em tempo real.

- **RNF6:** *O sistema deverá estar disponível 24 horas/365 dias por ano.*

Os serviços podem ser utilizados em qualquer dia da semana a qualquer hora. No entanto, os dias que poderão ser requisitados é que irão variar.

- **RNF7:** *O sistema deverá ser capaz de funcionar no browser Microsoft Edge.*

Sendo uma aplicação web, o *browser* de eleição será o Microsoft Edge.

2.2. Modelação do Software

Por forma a implementar de modo conciso e organizado todo o sistema a que nos propusemos, passamos pela modelação que consideremos necessária. Assim, todos os diagramas que a seguir se apresentam foram desenvolvidos de acordo com a linguagem *UML*, com o auxílio da ferramenta *Visual Paradigm*. Deste modo, incluímos grande parte dos diagramas desenvolvidos em anexo pelo que nos capítulos seguintes apresentaremos apenas alguns exemplares para que o presente documento se torne mais perceptível.

2.2. Modelo de domínio

O modelo de domínio permite representar o problema sobre o qual nos debruçamos. Assim, este permite capturar as entidades dos problemas e quais os relacionamentos entre elas. Efetivamente, o modelo de domínio ajuda a raciocinar sobre o projeto que nos propusemos a desenvolver. Nesse sentido, consideramos que este se trata de um modelo fulcral pois serve como um ponto de partida para toda a modelação.

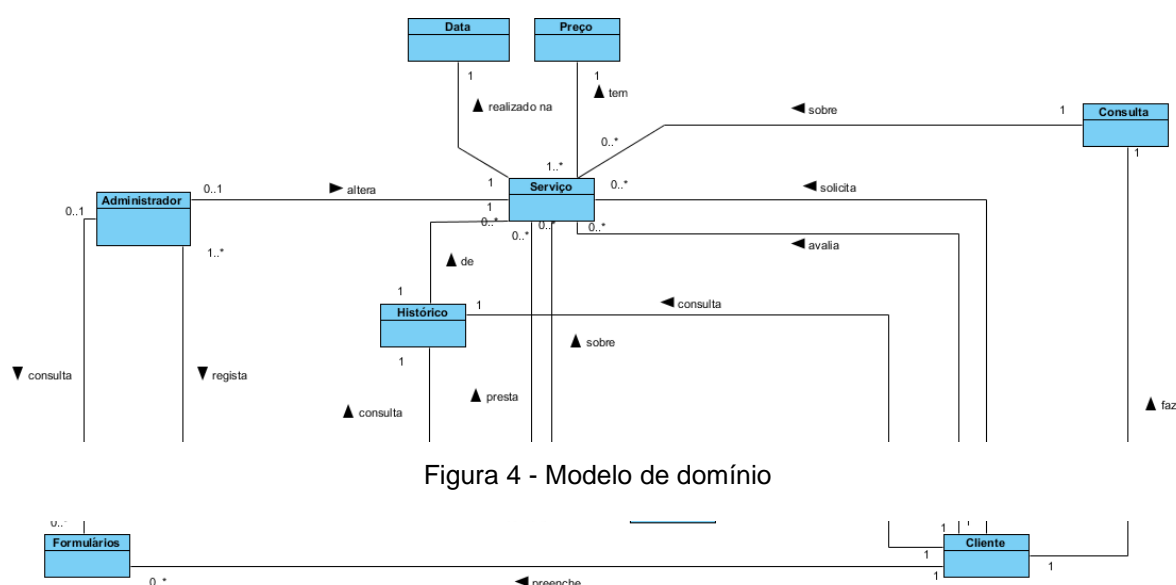


Figura 4 - Modelo de domínio

2.3. Use Cases

Os Use Cases são uma forma sistemática de capturar requisitos funcionais, fornecendo assim uma notação diagramática que permite modelar o contexto geral do sistema. Deste modo, uma vez que o levantamento de requisitos já tinha sido efetuado o desenvolvimento dos Use Cases surgiu de forma natural.

Foram vários os passos que demos, de modo a, tornar estes modelos os mais objetivos e precisos possível. Nesse sentido, procuramos especificar todas as possíveis utilizações do Sistema, pelo que, todo o processo foi desenvolvido de forma a ir ao encontro do *software* idealizado.

Logo à partida identificamos três atores do Sistema. Assim, o cliente e o funcionário foram os utilizadores que, logo à partida, consideramos suscetíveis de serem criados. Para além disso, segundo a contextualização efetuada, entendemos que o dono da empresa possui um papel fulcral na gestão dos recursos desta. De facto, este será o terceiro ator a ser criado. Para promover uma melhor organização e, conseqüentemente, um melhor entendimento, subdividimos os diagramas.

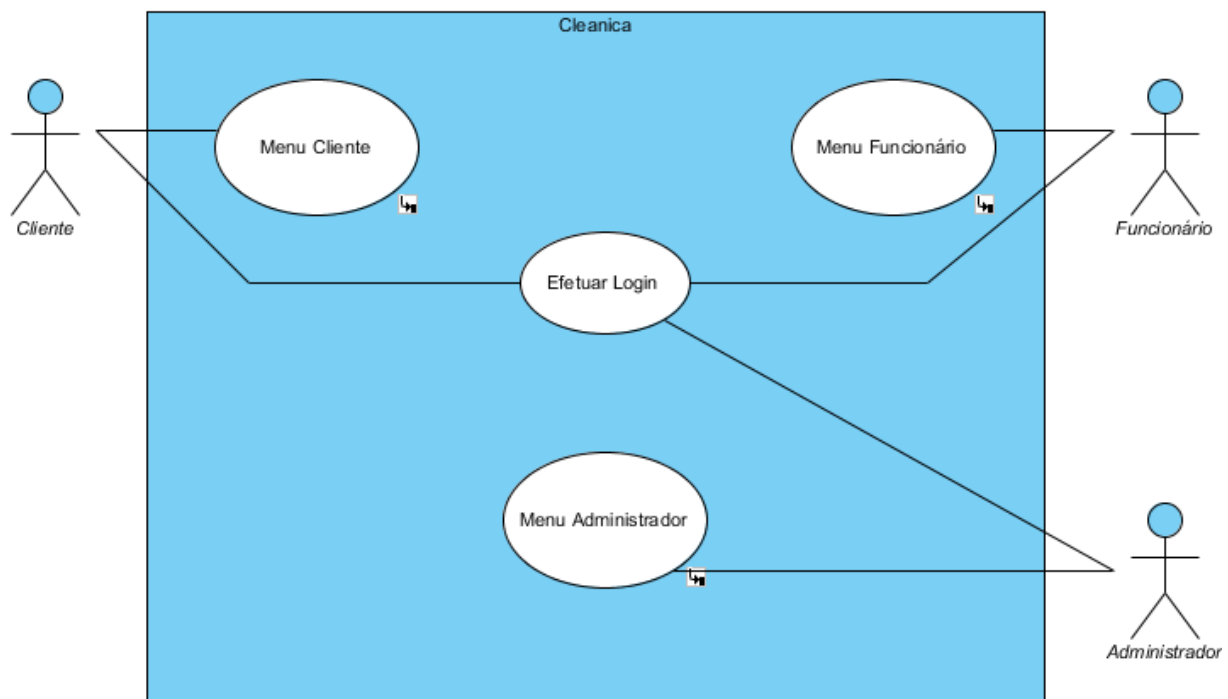


Figura 5 - Diagrama de Use Cases gerais

Assim, na figura 1, identificamos os três atores referidos, cada um associado ao seu menu. Cada menu é apresentado nos diagramas seguintes, pelo que estes fazem referência às funcionalidades que cada um dos utilizadores possuirá.

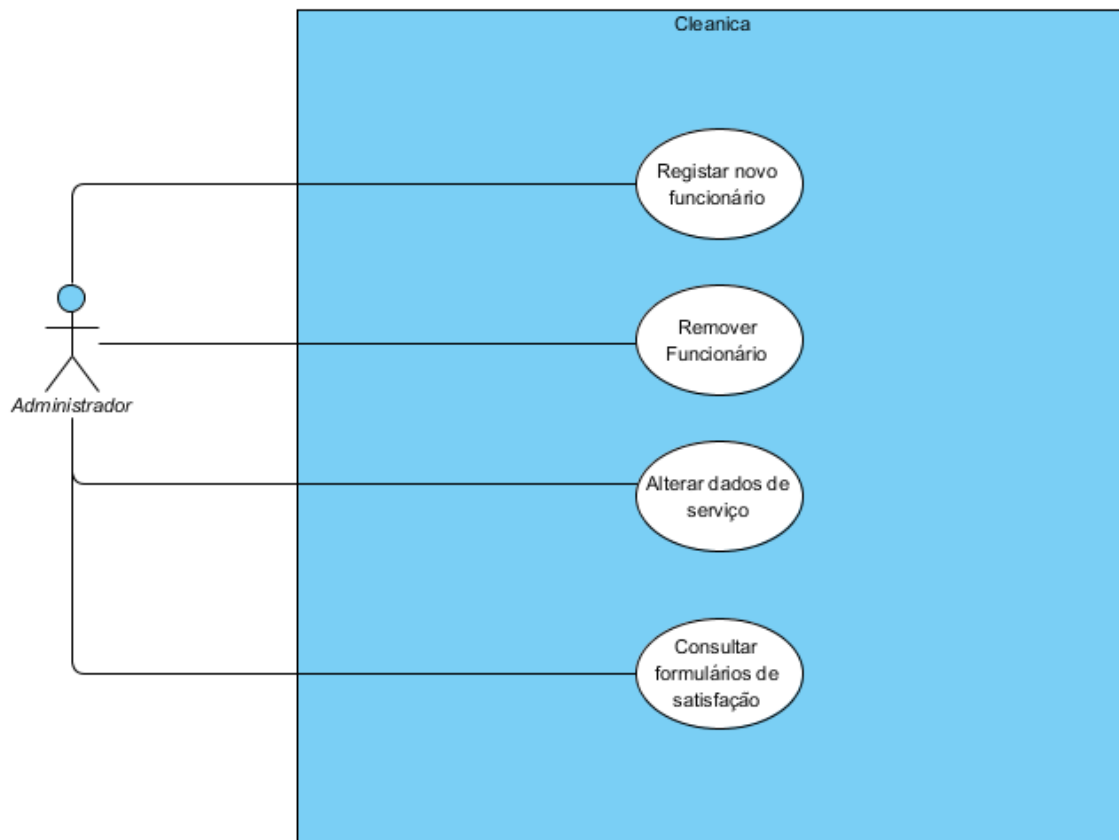


Figura 6 - Subdiagrama administrador

O Administrador é um utilizador prioritário do sistema, que será adicionado pelos engenheiros de software. Este possui privilégios que dizem respeito à forma como Cleanica controla os seus recursos. Os funcionários são adicionados pela empresa, pelo que também é esta que tem o poder de os remover. Para além disso, os formulários podem ser consultados por esta entidade. Por conseguinte, o administrador pode ainda alterar informações que dizem respeito aos serviços que este possui registados.

2.4. Especificação dos Use Cases

➤ Administrador

2.4.1.Registar novo funcionário

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Administrador fornece nome, data de nascimento, número de cartão de cidadão, IBAN e especialidade	
	2		Sistema valida dados
	3		Sistema regista novo funcionário na base de dados
	4		Sistema notifica que novo funcionário foi registado com sucesso
Excepção 1 [dados inválidos] (passo 2)		Actor Input	System Response
	1		Sistema indica que algum campo fornecido está errado

Figura 7 - Especificação do Use Case Registar funcionário

Um administrador autenticado no sistema, pode adicionar um novo funcionário. Assim, como informações de registo este necessitará de fornecer o nome, data de nascimento e número de cartão de cidadão. Para além disso, no sistema deverá também ser registado o IBAN deste para eventuais transferências bancárias fruto do trabalho deste na empresa. Por último, a especialidade também deverá ser anotada, já que, cada funcionário está confinado apenas a uma área. O sistema trata de validar todos os dados, indicando eventuais erros.

2.4.2. Remover funcionário

Preconditions	Estar autenticado	
Post-conditions		
Flow of Events		Actor Input
	1	Administrador indica email do funcionário que pretende remover
	2	
	3	Administrador indica que pretende remover o funcionário
	4	
	5	
		System Response
		Sistema valida se funcionário com esse email existe na base de dados
		Sistema remove funcionário da base de dados
		Sistema indica que funcionário foi removido com sucesso
Excepção 1 [Dados inválidos] (passo 2)		Actor Input
	1	
		System Response
		Sistema indica que o email indicado não existe na base de dados
Excepção 2 [Administrador cancela operação] (passo 3)		Actor Input
	1	
		System Response
		Sistema informa que operação foi cancelada

Figura 8 - Especificação do Use Case Remover funcionário

Depois de autenticado, um administrador pode ainda remover um funcionário. Assim, o sistema trata de remover da base de dados o funcionário que o utilizador indicar. No fim, indica que a remoção foi efetuada com sucesso.

2.4.3. Alterar dados de serviço

Preconditions	Estar autenticado		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Administrador seleciona serviço que pretende alterar	
	2		Sistema recolhe informação sobre o serviço na base de dados
	3	Administrador altera preço e/ou descrição do serviço pretendido	
	4		Sistema valida informações introduzidas pelo Administrador
	5		Sistema atualiza informações na base de dados
	6		Sistema indica que as informações do serviço foram alteradas com sucesso
Exceção 1 [Dados inválidos] (passo 4)		Actor Input	System Response
	1		Sistema indica que dados introduzidos são inválidos

Figura 9 - Especificação do Use Case Alterar dados de serviço

Este use case surge para especificar o facto de o administrador poder alterar dados relativos a um serviço. Este pode querer atualizar os preços deste ou simplesmente alterar a descrição que será apresentada no menu do cliente. Mais uma vez o sistema validará todos os dados inseridos.

2.4.4. Consultar formulários de satisfação

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Administrador seleciona formulário a ser observado	
	2		Sistema procura formulário na base de dados
	3		Sistema retorna formulário

Figura 10 - Especificação do Use Case Consultar formulários de satisfação

Por último surge o Use Case “Consultar formulários de satisfação” que permite ao administrador perceber a dedicação envolvida por parte dos funcionários no trabalho que efetuam.

Assim, este seleciona o formulário que pretende consultar, sendo que, é o sistema que trata de o retornar, depois de recolher as informações relativas a este na base de dados.

➤ Cliente

2.4.5. Efetuar registo

Preconditions		
Post-conditions	O actor fica registado no sistema	
Flow of Events		Actor Input
	1	Fornece nome, email, password, telemóvel e morada
	2	
	3	
		System Response
		Valida dados fornecidos
		Indica que o utilizador foi registado
Excepção 1 [dados inválidos] (passo 2)		Actor Input
	1	
		System Response
		Indica que dados são inválidos

Figura 11 - Especificação do Use Case Efetuar registo

Sendo uma especificação simples, apenas é importante referir que para o registo do cliente é necessário, fornecer dados relativos ao nome, email e password. Para além disso, é necessária a indicação do telemóvel para eventuais contactos. A morada, por sua vez, é útil para que seja possível identificar a localização do cliente.

2.4.6. Consultar serviços

Preconditions	Estar autenticado o sistema	
Post-conditions		
Flow of Events		Actor Input
	1	
	2	
		System Response
		Consulta a base de dados e extrai a informação relativa aos serviços disponíveis
		Envia a lista de todos os serviços encontrados

Figura 12 - Especificação do Use Case Consultar serviços

Os serviços serão apresentados aos clientes previamente autenticados, assim, este ao escolher a consulta destes poderá posteriormente requisitá-los.

2.4.7. Preencher formulários de satisfação

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Cliente preenche formulário	
	2		Sistema verifica integridade do formulário
	3		Sistema insere formulário na base de dados
	4		Sistema confirma envio de formulário
Excepção 1 [Formulário mal preenchido] (passo 5)		Actor Input	System Response
	1		Sistema informa que algum campo do formulário está incorreto

Figura 13 - Especificação do Use Case Preencher formulários de satisfação

O cliente pode preencher um formulário relativo ao serviço que fora executado na morada indicada. Este formulário será gerado pelo sistema, onde o cliente poderá atribuir uma pontuação ao funcionário. Depois de verificar a integridade do formulário, o sistema confirma o envio deste.

2.4.8. Consultar histórico

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Consulta a base de dados e extrai toda a informação dos serviços que o cliente requisitou
	2		Envia uma lista com toda a informação dos serviços requisitados pelo cliente

Figura 14 - Especificação do Use Case Consultar histórico

O cliente depois de autenticado pode consultar histórico dos serviços que requisitou. O sistema tratará do procedimento interno para a apresentação deste histórico.

2.4.9. Requisitar serviço

Sendo o Use Case “Requisitar serviço” um pouco complexo, subdividimos também este noutro diagrama. Assim, de seguida apresentaremos as especificações dos Use Cases, que se encontram na figura 4.

2.4.9.1. Requisitar serviço de lavagem de automóvel

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Sistema gera formulário com datas e horas disponíveis para o serviço, morada, dimensão, produtos e considerações importantes
	2	Utilizador preenche o formulário	
	3		Sistema pergunta se pretende continuar
	4	Utilizador indica que pretende continuar	
	5		Sistema valida formulário
	6	<<include>> Efetuar Pagamento	
	7		<<include>> Gerar Fatura
	8		<<include>> Escolher funcionário(s) para serviço
	9		Sistema informa que o serviço foi requisitado com sucesso
Excepção 1 [Utilizador cancela operação] (passo 3)		Actor Input	System Response
	1	Utilizador indica que pretende cancelar o requisito	
	2		Sistema informa que operação foi cancelada
Excepção 2 [Formulário Inválido] (passo 4)		Actor Input	System Response
	1		Sistema informa que algum parâmetro do formulário não está correto

Figura 15 - Especificação do Use Case Requisitar serviço de lavagem de automóvel

Este Use Case diz respeito, em particular, à requisição do serviço de automóvel. Nesse sentido, o cliente selecionará a data, a hora que pretende que o serviço seja realizado e ainda morada caso seja uma diferente da sua residencia atual. Para além disto, é importante indicar o tamanho do automóvel para fazer uma gestão de recursos necessários. O cliente pode ainda indicar considerações importantes para o funcionário. Posteriormente, o cliente pode efetuar pagamento, sendo que o sistema gera fatura e escolhe os funcionários para o serviço que acaba de ser requisitado.

2.4.9.2. Requisitar serviço de jardinagem

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Sistema gera formulário com datas e horas disponíveis para o serviço, morada, dimensão, produtos e considerações importantes
	2	Utilizador preenche o formulário	
	3		Sistema pergunta se pretende continuar
	4	Utilizador indica que pretende continuar	
	5		Sistema valida formulário
	6	<<include>> Efetuar Pagamento	
	7		<<include>> Gerar Fatura
	8		<<include>> Escolher funcionário(s) para serviço
	9		Sistema informa que o serviço foi requisitado com sucesso
Exceção 1 [Utilizador cancela operação] (passo 3)		Actor Input	System Response
	1	Utilizador indica que pretende cancelar o requisito	
	2		Sistema informa que operação foi cancelada
Exceção 2 [Formulário Inválido] (passo 4)		Actor Input	System Response
	1		Sistema informa que algum parâmetro do formulário não está correto

Figura 16 - Especificação do Use Case Requisitar serviço de jardinagem

Este Use case é em tudo semelhante ao anterior, no entanto, as informações que são fornecidas são distintas.

2.4.9.3. Requisitar serviço de limpeza de piscinas

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Sistema gera formulário com datas e horas disponíveis para o serviço, morada, dimensão, produtos de limpeza e considerações importantes
	2	Utilizador preenche o formulário	
	3		Sistema pergunta se pretende continuar
	4	Utilizador indica que pretende continuar	
	5		Sistema valida formulário
	6	<<include>> Efetuar Pagamento	
	7		<<include>> Gerar Fatura
	8		<<include>> Escolher funcionário(s) para serviço
	9		Sistema informa que o serviço foi requisitado com sucesso
Excepção 1 [Utilizador cancela operação] (passo 3)		Actor Input	System Response
	1	Utilizador indica que pretende cancelar o requisito	
	2		Sistema informa que operação foi cancelada
Excepção 2 [Formulário Inválido] (passo 4)		Actor Input	System Response
	1		Sistema informa que algum parâmetro do formulário não está correto

Figura 17 - Especificação do Use Case Requisitar serviço de jardinagem

Mais uma vez o Use Case do serviço de jardinagem é idêntico aos anteriores. A única diferença ocorre apenas no formulário apresentado para a requisição dos serviços. Neste formulário encontrar-se-á a dimensão da área a ser intervencionada, segundo diferentes graus, para que seja mais fácil a alocação de recursos numa fase posterior. Também se realça o facto do cliente poder optar por fornecer os produtos de limpeza o que vai de encontro à política estipulada pela Cleanica.

2.4.9.4. Requisitar serviço de limpeza de interiores

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Sistema gera formulário com datas e horas disponíveis para o serviço, morada, dimensão, fornecimento de produtos de limpeza e considerações importantes
	2	Utilizador preenche o formulário	
	3		Sistema pergunta se pretende continuar
	4	Utilizador indica que pretende continuar	
	5		Sistema valida formulário
	6	<<include>> Efetuar Pagamento	
	7		<<include>> Gerar Fatura
	8		<<include>> Escolher funcionário(s) para serviço
	9		Sistema informa que o serviço foi requisitado com sucesso
Exceção 1 [Utilizador cancela operação] (passo 3)		Actor Input	System Response
	1	Utilizador indica que pretende cancelar o requisito	
Exceção 2 [Formulário Inválido] (passo 4)		Actor Input	System Response
	1		Sistema informa que algum parâmetro do formulário não está correto

Figura 18 - Especificação do Use Case Requisitar serviço de limpeza de interiores

Este será o quarto serviço que o cliente poderá requisitar. A sua especificação é semelhante aos serviços anteriormente apresentados.

2.4.9.5. Includes: Efetuar Pagamento, escolher funcionários para serviço e gerar fatura

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Utilizador introduz nome do titular do cartão de crédito, número do cartão, data de validade e código de segurança	
	2		Sistema valida informações fornecidas pelo utilizador
	3		Sistema efetua a transação
	4		Sistema notifica que a transação foi bem sucedida
Exceção 1 [dados inválidos] (passo 2)		Actor Input	System Response
	1		Sistema informa que os dados fornecidos são inválidos
Exceção 2 [erro na transação] (passo 3)		Actor Input	System Response
	1		Sistema informa que houve um problema na transação, e que por isso, não foi possível concluir o pagamento

Figura 19 - Especificação do Use Case Efetuar Pagamento

Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Cliente indica dimensão do serviço	
	2		Sistema recolhe informações sobre funcionários aptos a realizar o serviço
Comp. Alternativo [Serviço de pequena dimensão] (passo 2)		Actor Input	System Response
	1		Sistema determina o funcionário disponível mais perto da localização do serviço
	2		Regressa a 2
Comp. Alternativo [Serviço de média dimensão] (passo 2)		Actor Input	System Response
	1		Sistema determina os dois funcionários disponíveis mais perto da localização do serviço
	2		Regressa a 2
Comp. Alternativo [Serviço de grande dimensão] (passo 2)		Actor Input	System Response
	1		Sistema determina os três funcionários disponíveis mais perto da localização do serviço
	2		Regressa a 2

Figura 20 - Especificação do Use Case Escolher funcionário(s) para o serviço

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Sistema reúne as informações relativas ao pagamento anteriormente efetuado
	2		Sistema extrai email do Utilizador
	3		Sistema pergunta qual o número de contribuinte do Utilizador
	4	Utilizador insere número de contribuinte	
	5		Sistema valida operação
	6		Sistema gera fatura
	7		Sistema envia fatura gerada para o email anteriormente extraído
	8		Sistema informa que a fatura foi enviada para o email do Utilizador

Figura 21 - Especificação do Use Case Gerar fatura

Cada um destes Use Cases foram desenvolvidos por forma a simplificar a criação dos restantes diagramas. O *Efetuar pagamento*, trata de toda as burocracias que o pagamento envolve. O *Escolher funcionário(s)* trata de alocar os recursos necessários a cada trabalho. Por último o *Gerar fatura* é o que permite gerar a fatura que será enviada como confirmação para o email.

2.4.10. Alterar dados

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Fornece nome e/ou morada e/ou nº de telemóvel e/ou password	
	2		Sistema valida dados fornecidos
	3		Sistema atualiza informações na base de dados
	4		Sistema indica que dados foram alterados com sucesso
Exceção 1 [dados inválidos] (passo 2)		Actor Input	System Response
	1		Sistema indica que dados são inválidos

Figura 22 - Especificação do Use Case Alterar dados

O Cliente sempre que entender poderá alterar os dados relativos à sua conta. Estes poderão ser nome, morada, telemóvel e/ou password.

➤ Funcionário

2.4.11. Consultar calendário de serviços

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1		Consulta a base de dados de forma a obter todos os trabalhos a serem realizados pelo funcionário em sessão
	2		Envia a lista de todos os trabalhos a serem realizados

Figura 23 - Especificação do Use Case Consultar calendário de serviços

O funcionário poderá consultar um calendário onde estão os serviços que este tem a efetuar.

2.4.12. Consultar histórico de encargos

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Utilizador seleciona opção de pesquisa (por datas ou por cliente)	
	2		Consulta base de dados e extraí informação pretendida consoante o parâmetro de pesquisa
	3		Envia o histórico da informação para o utilizador

Comp. Alternativo 1 [por datas] (passo 1)		Actor Input	System Response
	1	Utilizador seleciona a data pretendida	
	2		<i>Regressa ao passo 2</i>
Comp. Alternativo 2 [por cliente] (passo 1)		Actor Input	System Response
	1	Utilizador introduz o nome de um cliente	
	2		Sistema verifica a existência do cliente fornecido pelo utilizador
	3		<i>Regressa ao passo 2</i>
Excepção 1 [nome de cliente inválido] (passo 2.2)		Actor Input	System Response
	1		Sistema informa que o cliente fornecido pelo utilizador não existe

Figura 24 - Especificação do Use Case Consultar histórico de encargos

O funcionário poderá consultar um histórico, ordenado por datas ou clientes. Nesse sentido, o sistema deverá dar resposta a cada uma das pesquisas efetuadas, validando também os dados fornecidos pelo utilizador

2.4.13. Confirmar realização de serviço

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Funcionário seleciona serviço que pretende confirmar	
	2		Sistema gera formulário com informações sobre o serviço e algumas considerações a ter em conta
	3	Funcionário preenche o formulário recebido	
	4		Sistema insere formulário na base de dados
	5		Sistema declara serviço anteriormente selecionado como efetuado
	6		Sistema informa que processo foi concluído com sucesso

Figura 25 - Especificação do Use Case Confirmar realização do serviço

O Use Case de “Confirmar a realização do serviço” surge para o funcionário ter a possibilidade de anotar quais os serviços que já efetuou, referindo assim, considerações relativas a este.

2.4.14. Pedir direções

Preconditions			
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Funcionário introduz morada destino	
	2		Sistema mostra caminho mais rápido para a morada pretendida

Figura 26 - Especificação do Use Case Pedir direções

Sempre que o funcionário entender poderá pedir indicações sobre a morada do cliente. Assim, o sistema mostrará o caminho mais rápido para esta.

2.4.15. Alterar dados funcionário

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Fornece nome e/ou password e/ou telemóvel e/ou IBAN e/ou morada	
	2		Valida dados fornecidos
	3		Indica que dados foram alterados com sucesso
Excepção 1 [dados inválidos] (passo 2)		Actor Input	System Response
	1		Indica que dados são inválidos

Figura 27 - Especificação do Use Case Alterar dados funcionário

Finalmente, o funcionário poderá alterar dados relativos à conta que possui. Estes dados podem ser, nome, password, telemóvel, IBAN e/ou morada. O sistema depois de verificar os dados trata de guardar estas alterações.

2.5. Diagramas de Sequência de Sistema

Os diagramas de sequência de sistema(DSS) permitem começar uma análise do que o sistema vai ser. Sendo assim, representam uma visão de mais alto nível. Deste modo, consideramos que os DSS's apresentados seriam uma mais valia para a implementação da aplicação facilitando o trabalho que teremos em mãos. Estes não surgiram imediatamente na sua forma final pelo que foram sendo refinados como os exemplos seguintes o demonstram.

➤ Administrador

2.5.1.Registar novo funcionário

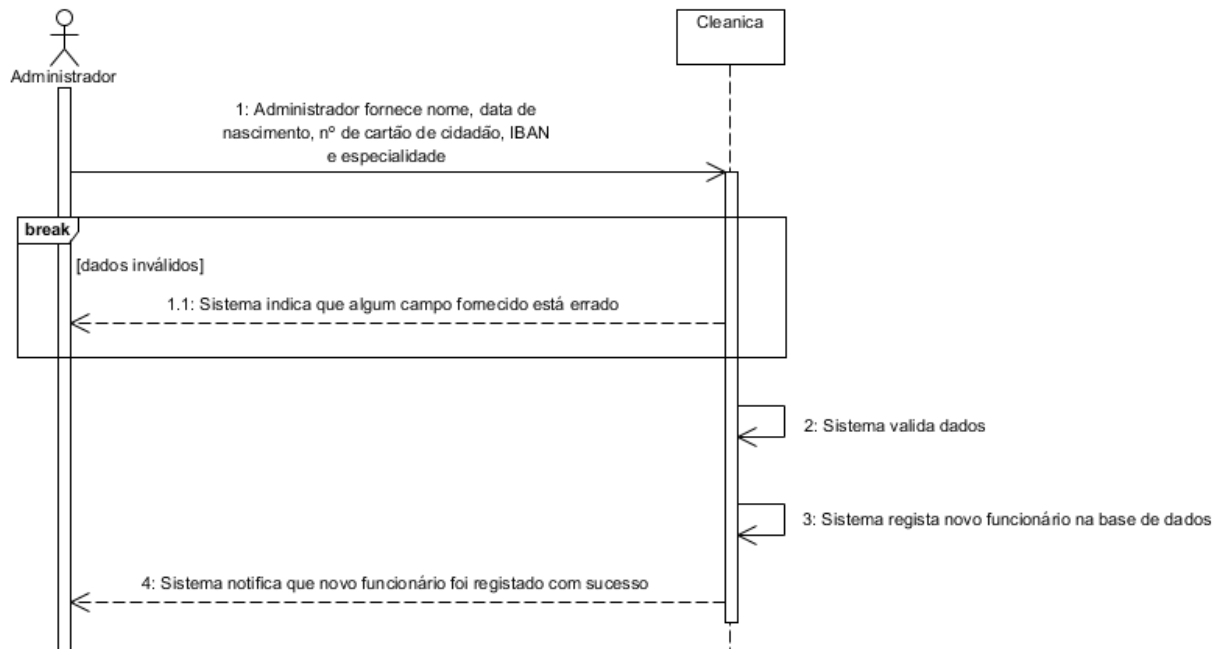


Figura 28 - Diagrama de sequência: registar novo funcionário

Seguidamente, apresentaremos o aprimoramento deste DSS. Ao longo desta secção são apresentados os respetivos diagramas de sequência e a seu respetivo refinamento.

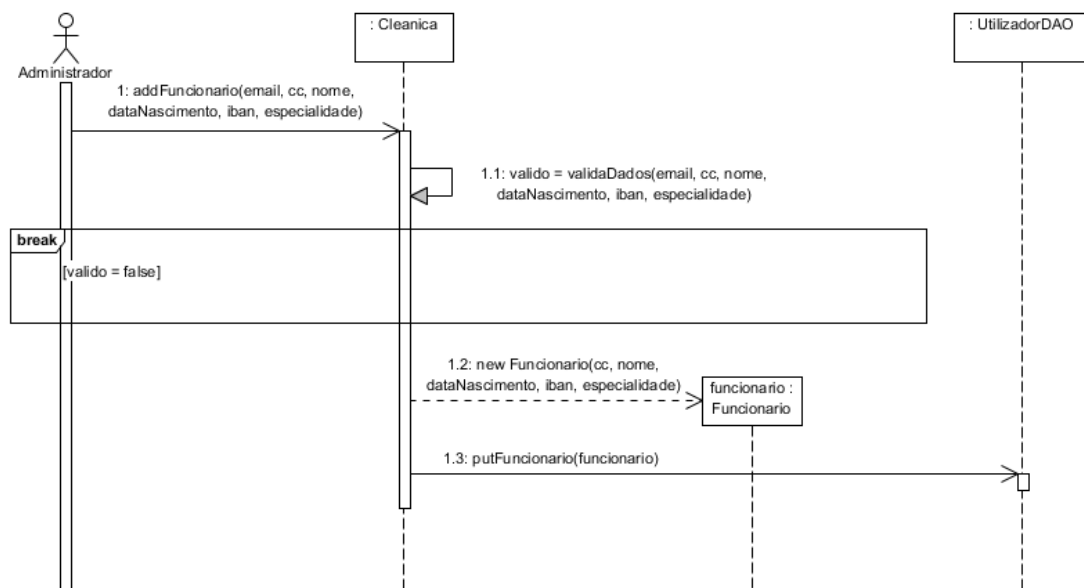


Figura 29 - Especificação do diagrama de sequência: registrar funcionario

2.5.2. Alterar dados de serviço

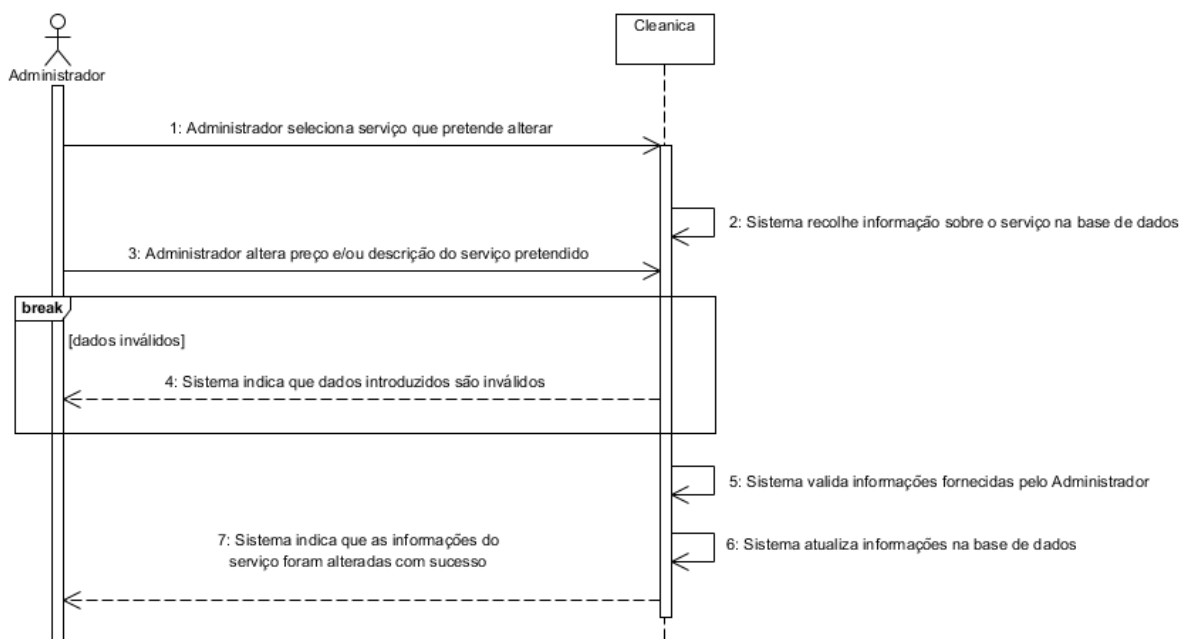


Figura 30 - Diagrama de sequência: alterar dados de serviço

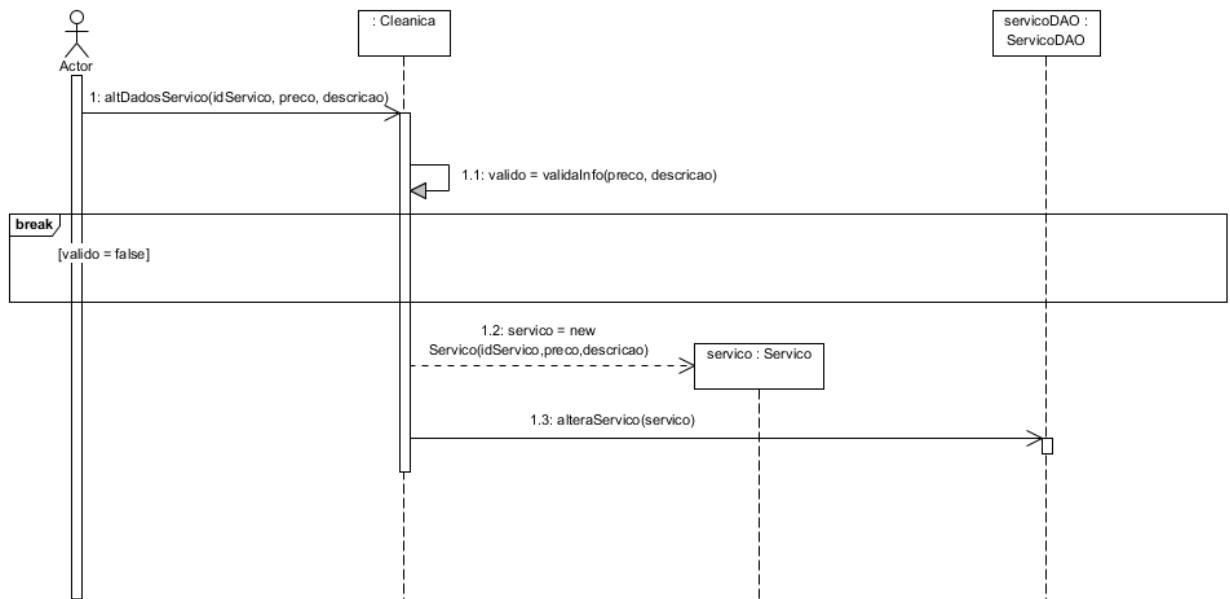


Figura 31 - Especificação do diagrama: alterar dados de serviço

➤ Cliente

2.5.3.Preencher formulário de satisfação

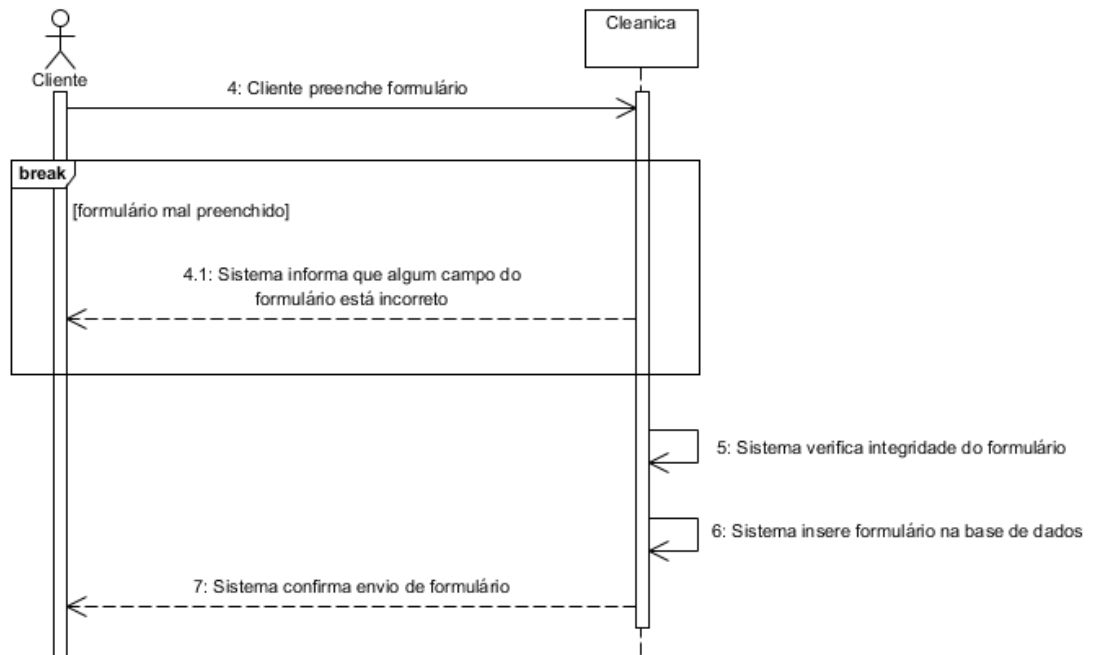


Figura 32 - Diagrama de sequência: preencher formulário de satisfação

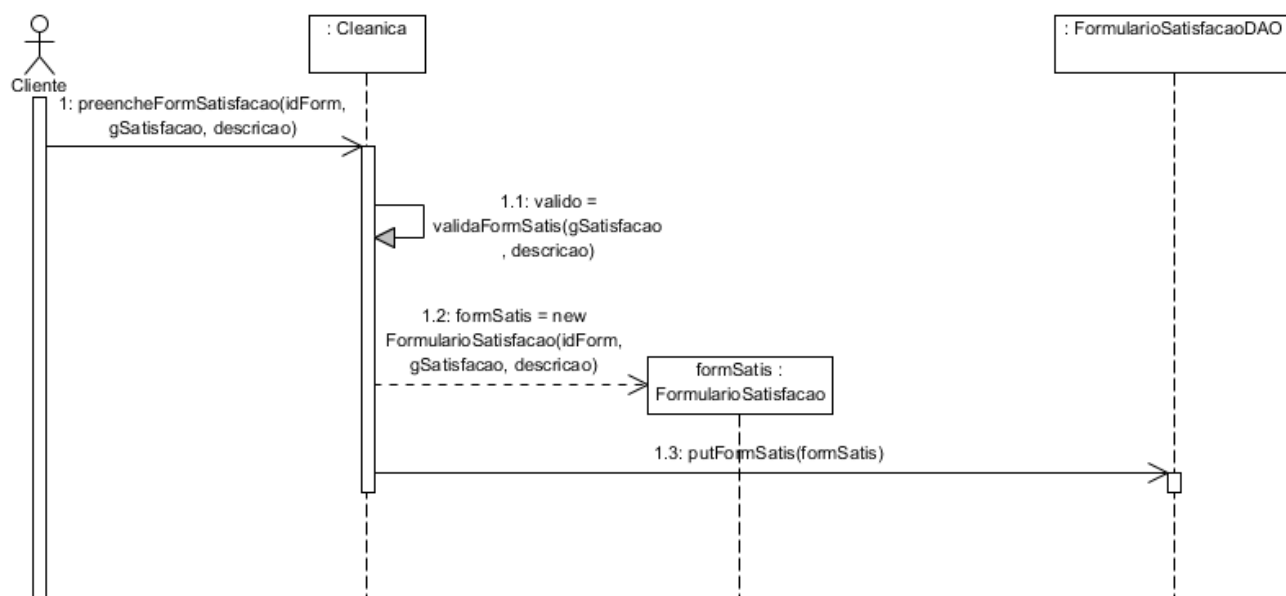


Figura 33 - Especificação do diagrama de sequência: preencher formulário de satisfação

Continua a ser importante referir que apesar de terem sido modelados todos os Use Cases anteriormente apresentados, nesta secção apenas incluímos alguns diagramas. Sendo assim, “Preencher formulário de satisfação” é um dos muitos DSS desenvolvidos. Assim, a figura 33 apresenta um estado final da especificação do diagrama de sequência em que cada classe já se encontra substituída pelo respetivo DAO.

2.5.4. Requisitar serviços

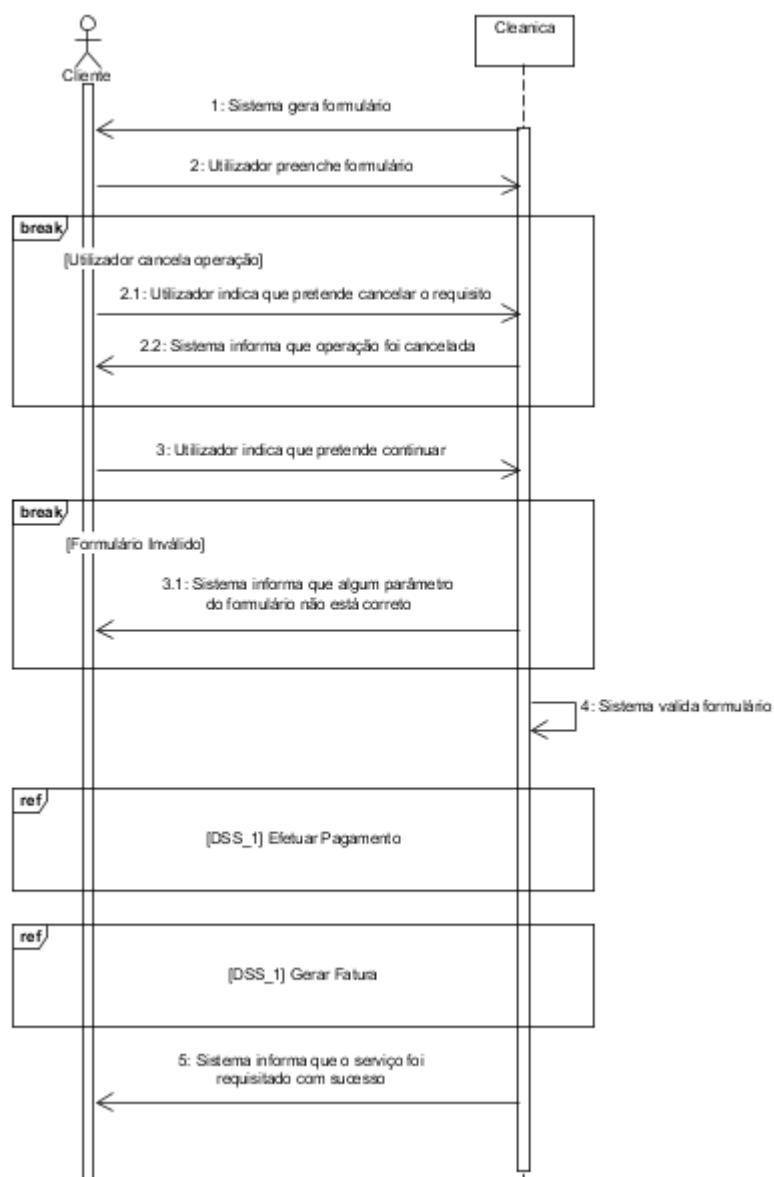


Figura 34 - Diagrama de sequência: requisitar serviço

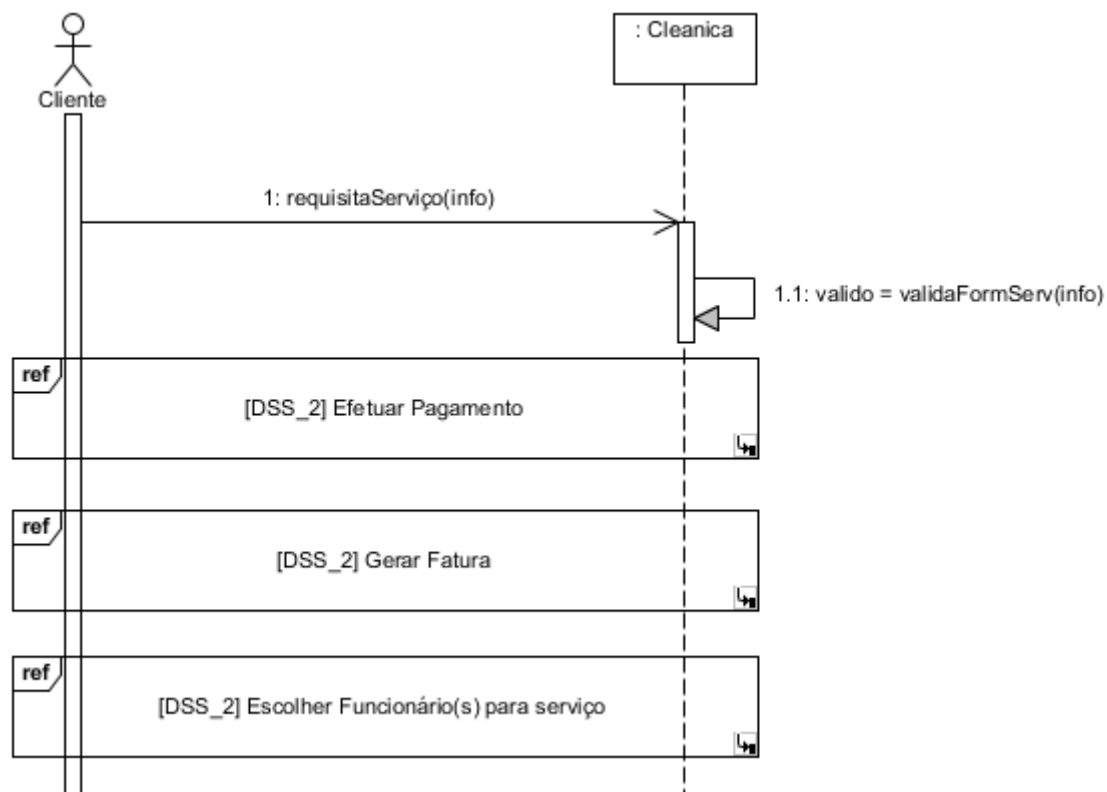


Figura 35 - Especificação do diagrama de sequências: requisitar serviço

➤ Funcionário

2.5.5. Confirmar realização de serviço

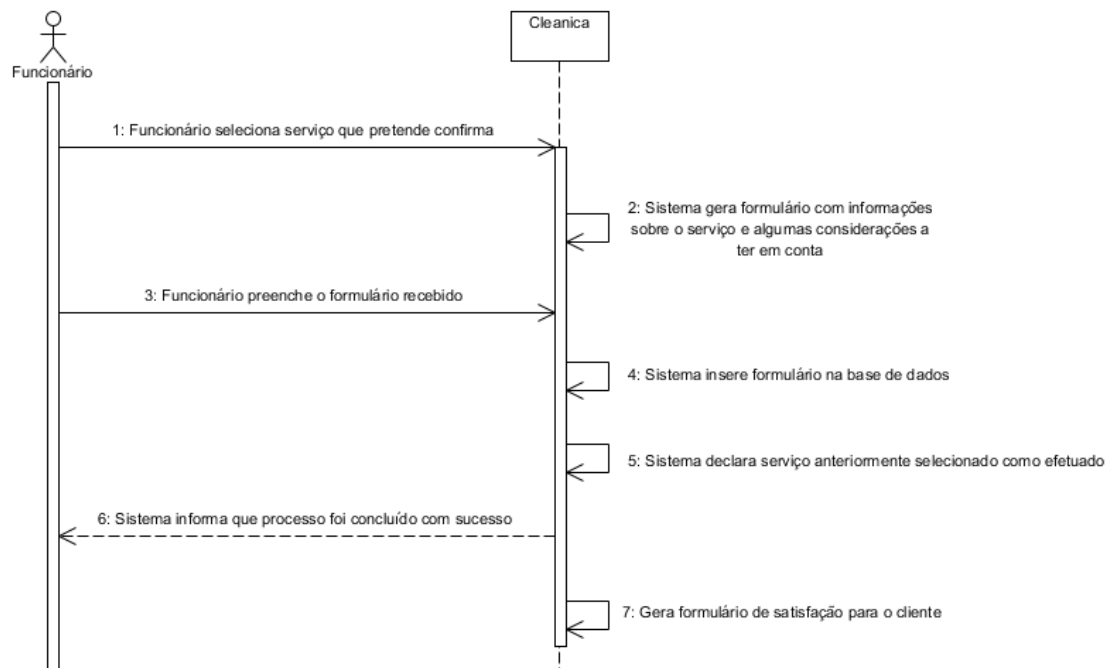


Figura 36 - Diagrama de sequência: confirmar realização de serviço

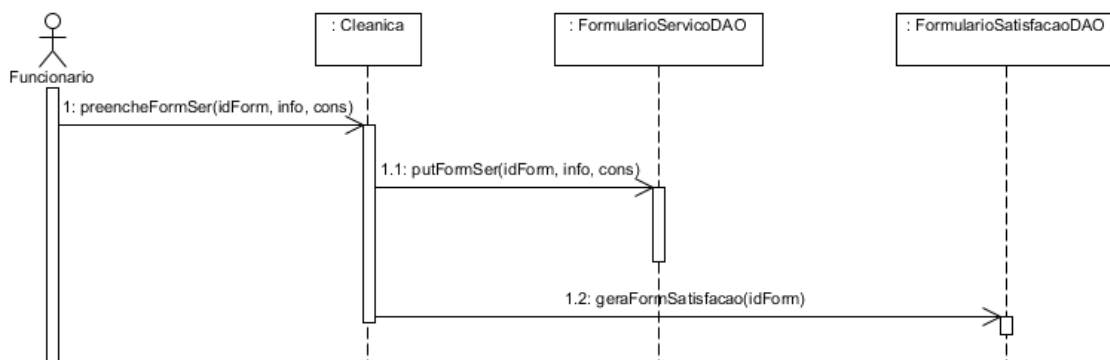


Figura 37 - Especificação do diagrama de sequência: confirmar serviço

2.5.6.Consultar calendário de serviços

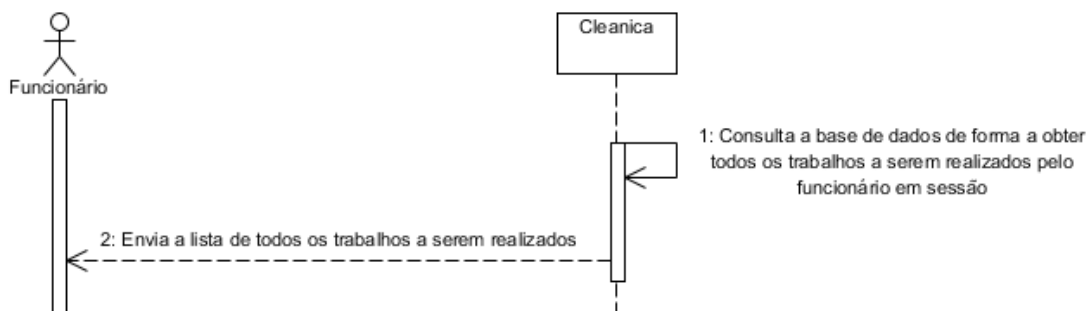


Figura 38 - Diagrama de sequência: consultar calendário de serviços

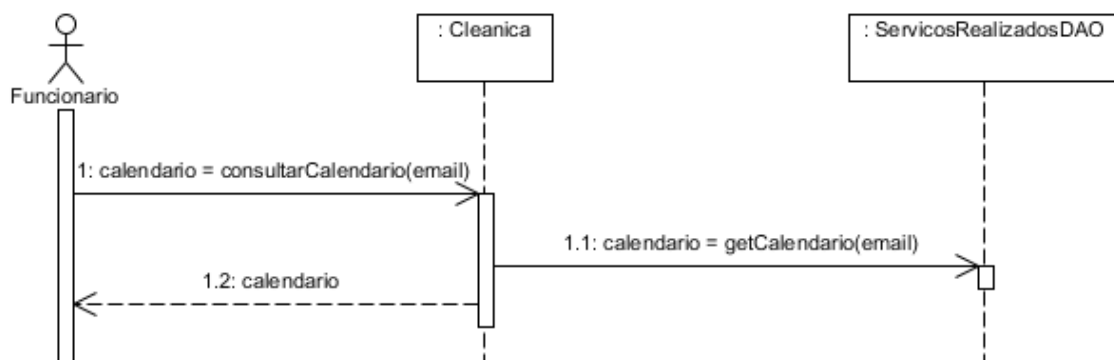


Figura 39 - Especificação do diagrama de sequência: consultar calendário de serviços

2.6. Diagrama de Classes

Decidimos implementar um diagrama de classe pois são diagramas estruturais que permitem ilustrar as classes e o relacionamento entre estas. Assim estes permitem descrever a estrutura de um sistema, modelando ainda os atributos e as operações entre objetos. Nesse sentido, a obtenção deste diagrama resultou de uma análise cuidada do modelo de domínio, identificando assim, quais as classes suscetíveis de serem criadas. Vale a pena realçar a existência da classe Utilizador dentro da Cleanica. Esta tem o intuito de guardar as informações do utilizador em sessão, sendo assim mais fácil aceder a estas sempre que necessário.

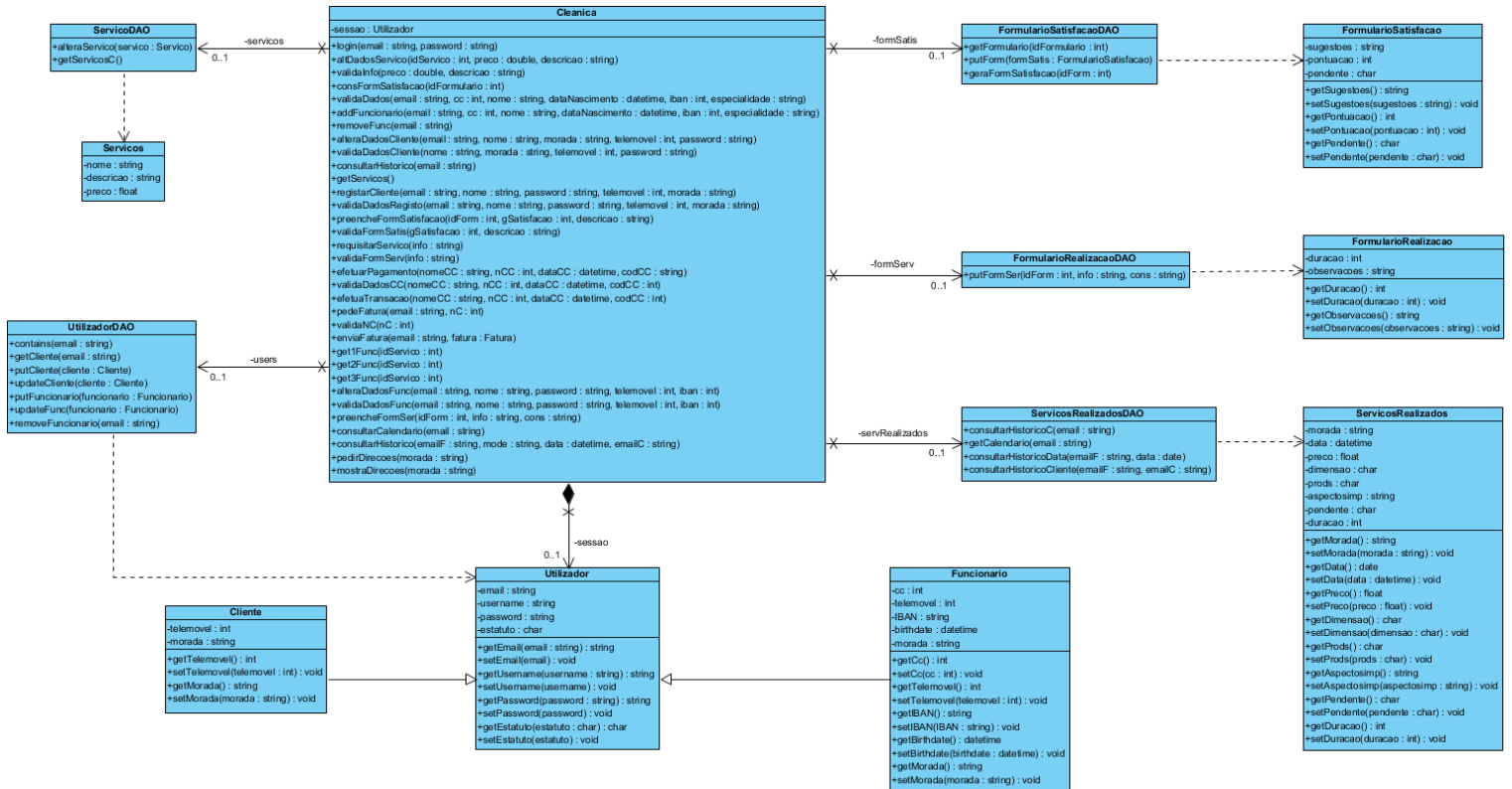


Figura 40 – Diagrama de classes

2.7. Diagrama de Atividade

Sendo que se trata de uma aplicação consideramos que seria útil especificar o comportamento dos diferentes utilizadores. Assim, baseamo-nos nos diagramas de atividade, uma vez que, estes fundamentam-se em modelos de fluxo de dados. Os diagramas que a seguir se apresentam, dizem respeito ao que resulta da interação dos utilizadores com o sistema. Sendo assim, encontra-se modelado o fluxo ocorrido durante a tomada de decisão dos clientes, dos funcionários e do administrador. Cada modelo encontra-se dividido em duas partições fundamentais. A primeira identifica o utilizador de que se trata, enquanto que a segunda se refere à aplicação que irá ser criada, Cleanica.

2.7.1. Diagrama de atividade Administrador

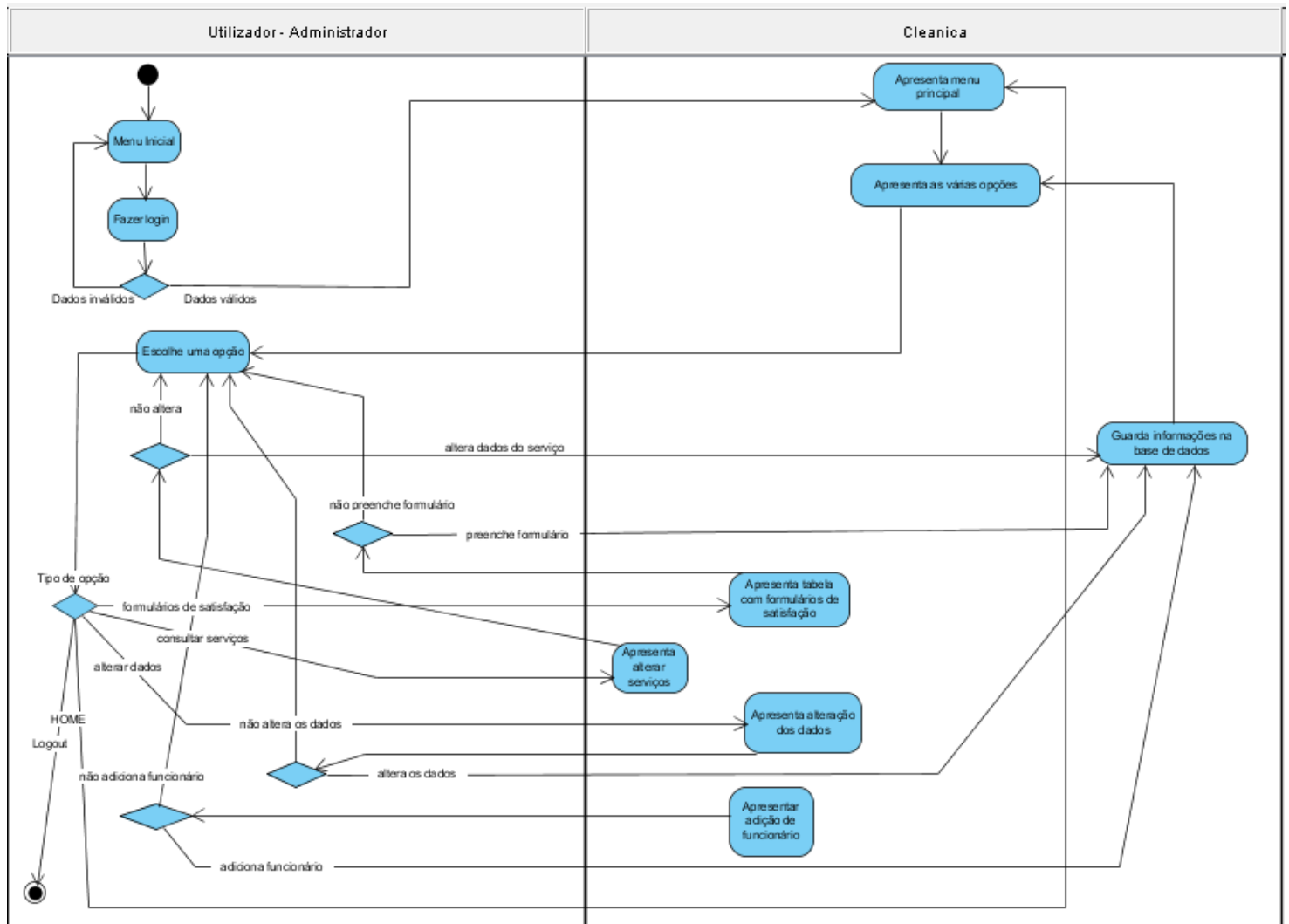


Figura 41 - Diagrama de Atividade Administrador

A figura 41 representa o desfecho das ações tomadas pelo administrador. Este possui um conjunto de opções que têm de ser registadas na base de dados, através da aplicação. Sendo que se trata do administrador, este não pode efetuar registo pois será inserido no sistema, à partida, pelos engenheiros de software. Realçar ainda que é toda a aplicação que cria a parte da interface que será acedida pelo utilizador. Este fluxo, termina apenas quando o administrador carregar no botão de término de sessão.

2.7.2. Diagrama de atividade Cliente

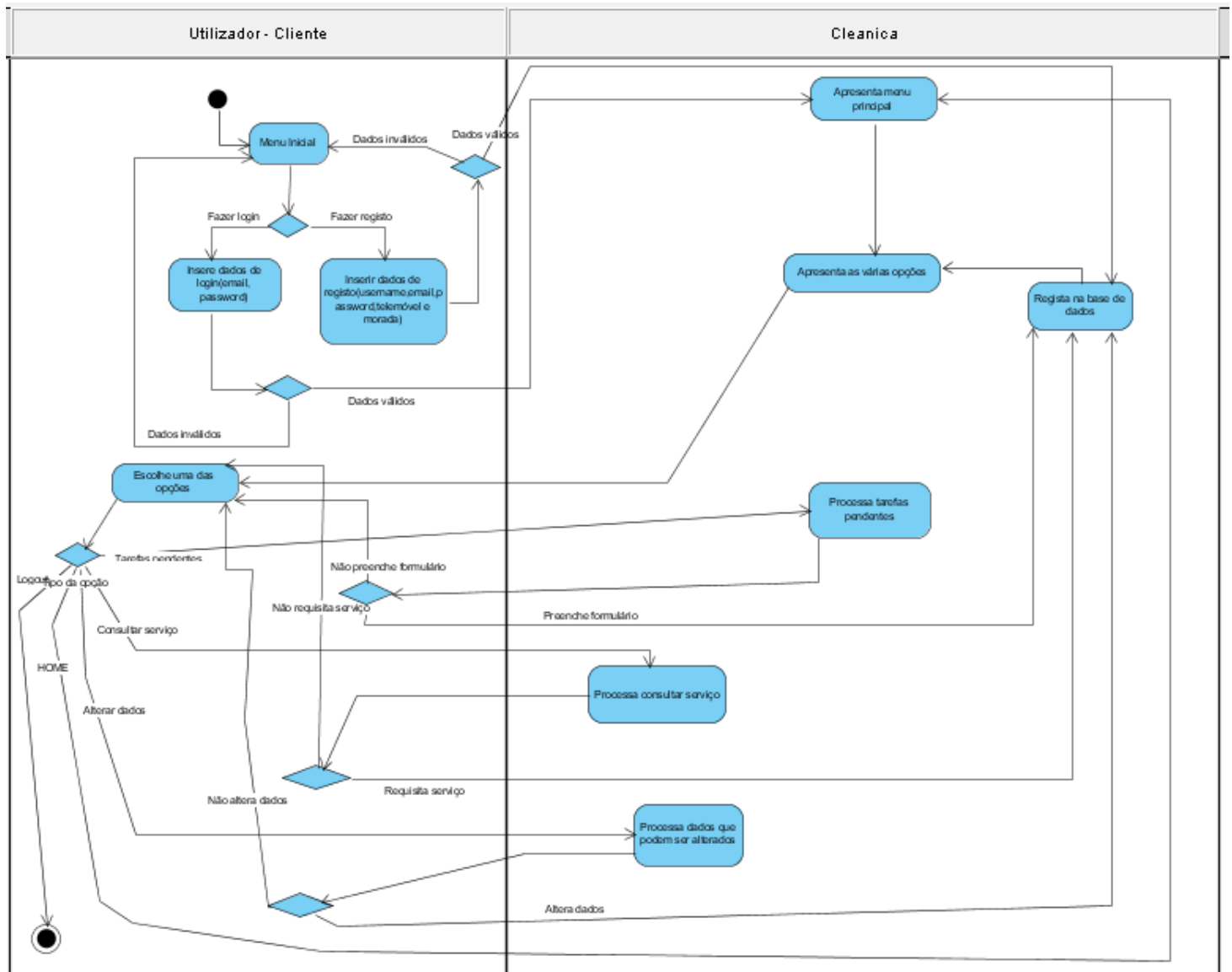


Figura 42 - Diagrama de Atividade Cliente

Este diagrama de atividade pretende ilustrar o fluxo de informação que ocorre quando o cliente interage com a plataforma. Assim, sempre que uma opção é tomada, e é levada até ao fim (não sendo o processo cancelado), as alterações têm de ser registadas. Conforme o diagrama, é a aplicação que trata de registar na base de dados estas alterações. Para além disso, a informação que é apresentada nos diferentes menus ocorre, como é óbvio, do lado da aplicação.

2.7.3. Diagrama de atividade Funcionário

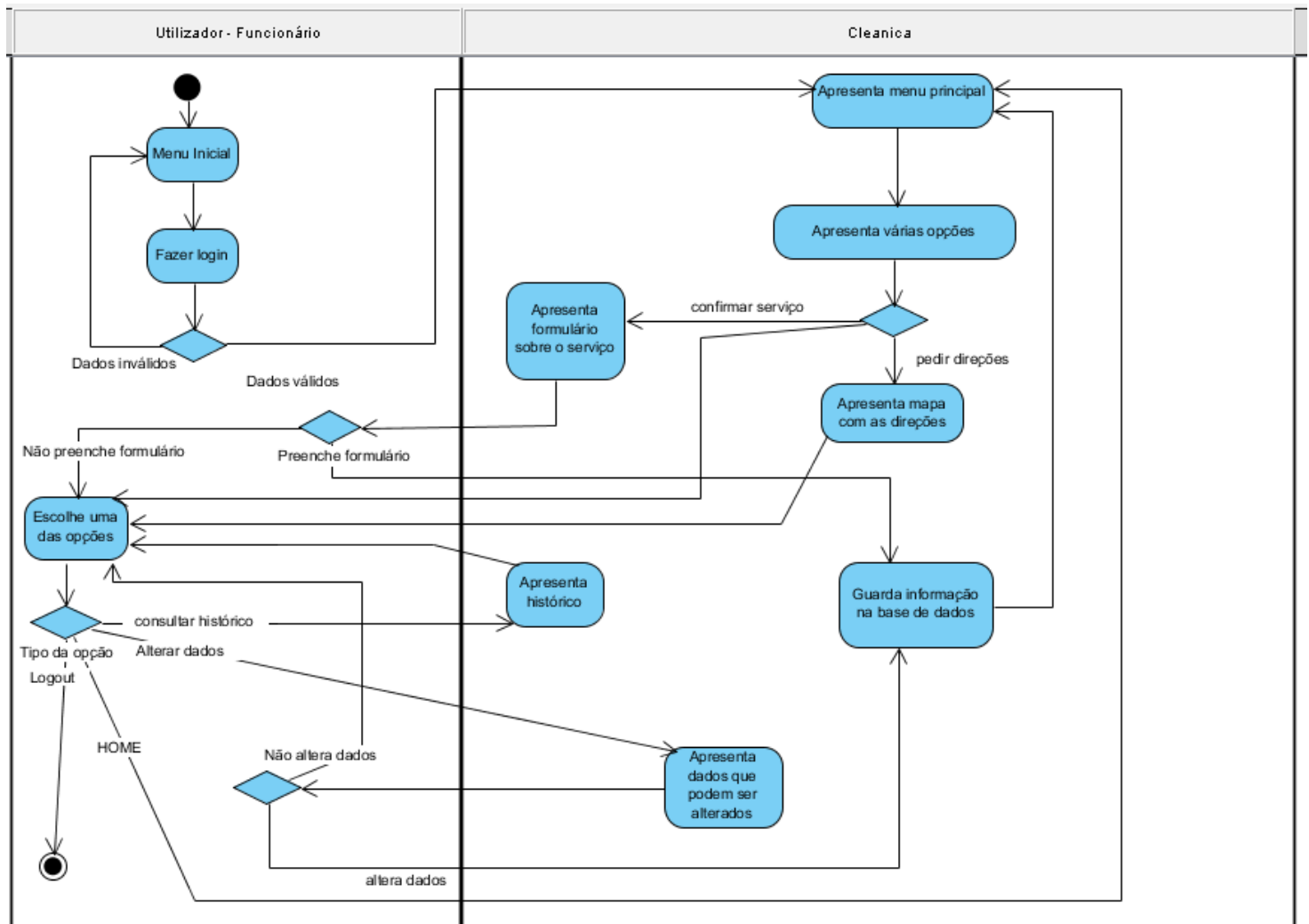


Figura 43 - Diagrama de Atividade Funcionário

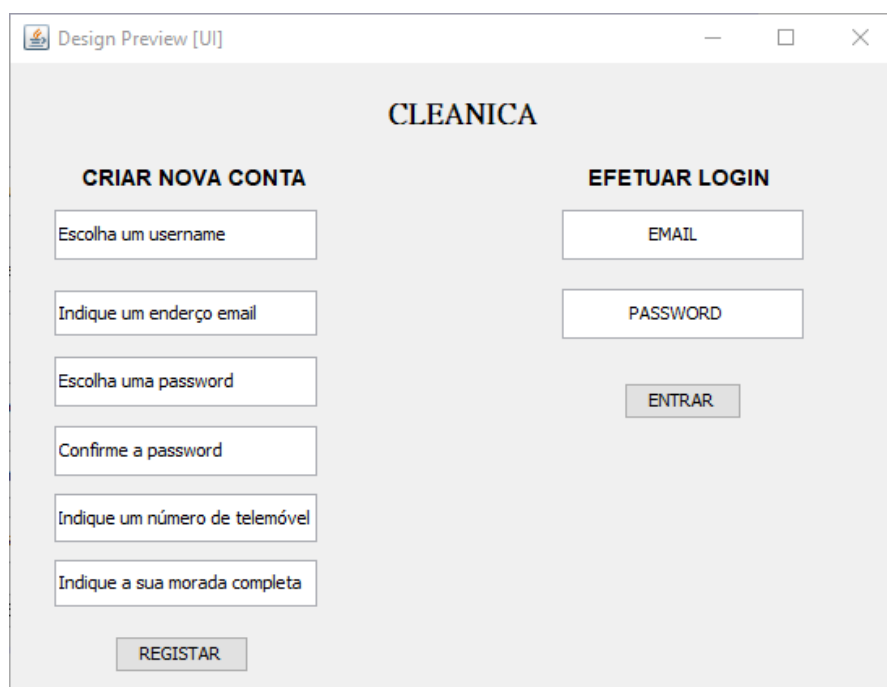
Por último, surge o diagrama de atividades relativo ao funcionário. A particularidade neste diagrama é que no menu “Home” deste, já se encontram algumas opções. Este pode usufruir destas, ou simplesmente, pode utilizar outros serviços tal como os restantes utilizadores.

2.8. Mockup da interface

Foram criados mockup's da interface para que na implementação da aplicação o processo decorra da forma mais simples e estruturada possível.

O menu de login é o que estará de igual forma, disponível para todos os utilizadores. Pensamos que a interface deste deveria ser simples e intuitiva permitindo mais facilmente o registo de novos clientes. Assim, decidimos que o registo e o login dos utilizadores pudessem ser feitos na mesma página. Para além disto, pretendemos apresentar para cada utilizador, um conjunto de opções diferentes, otimizadas para cada tipo de finalidade. Foi criado um menu para cada utilizador, no entanto, cada um destes, terá uma barra (acionada por um botão) onde serão mostradas todas as opções. Assim, de seguida especificamos individualmente cada uma das interfaces. É importante referir que sendo um mockup, trata-se apenas de um protótipo, não sendo de todo, o modelo final.

➤ Menu de login e de registo



A imagem mostra uma janela de design intitulada "Design Preview [UI]" com o título "CLEANICA" no topo. A interface está dividida em duas colunas principais: "CRIAR NOVA CONTA" à esquerda e "EFETUAR LOGIN" à direita. Na coluna de registo, há campos de texto para "Escolha um username", "Indique um endereço email", "Escolha uma password", "Confirme a password", "Indique um número de telemóvel" e "Indique a sua morada completa", seguidos por um botão "REGISTAR". Na coluna de login, há campos para "EMAIL" e "PASSWORD", seguidos por um botão "ENTRAR".

Figura 44 - Mockup Login

2.8.1. Administrador

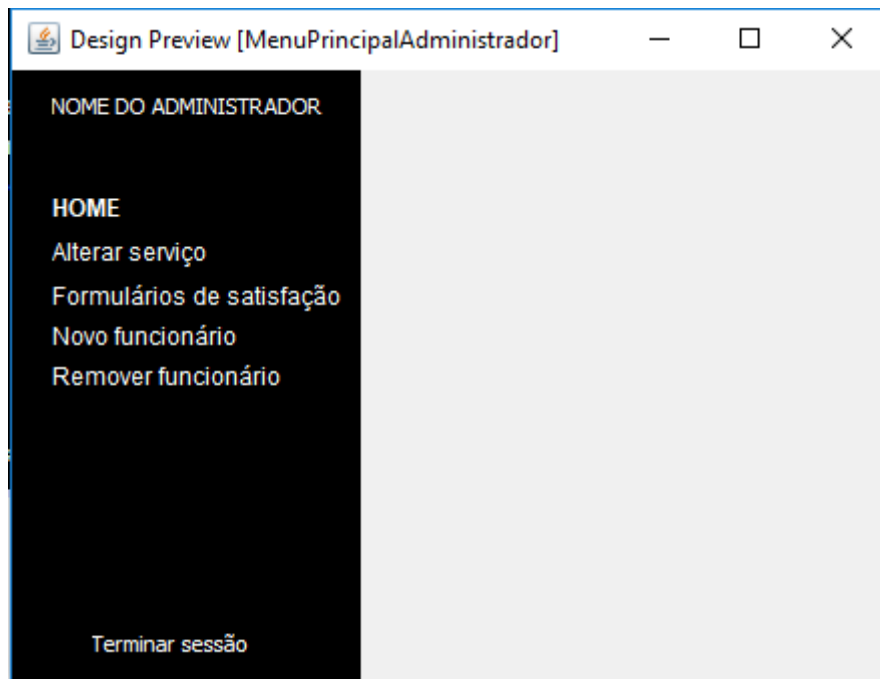


Figura 45 - Mockup Menu principal Administrador

No caso do administrador, o ecrã principal será meramente indicativo, servindo apenas, como um ecrã de boas vindas. Neste caso, as opções fazem alusão às funcionalidades que o administrador possui. Maioritariamente, estas prendem-se com a gestão de recursos. Cada uma destas opções contém o seu próprio menu, abaixo representado.

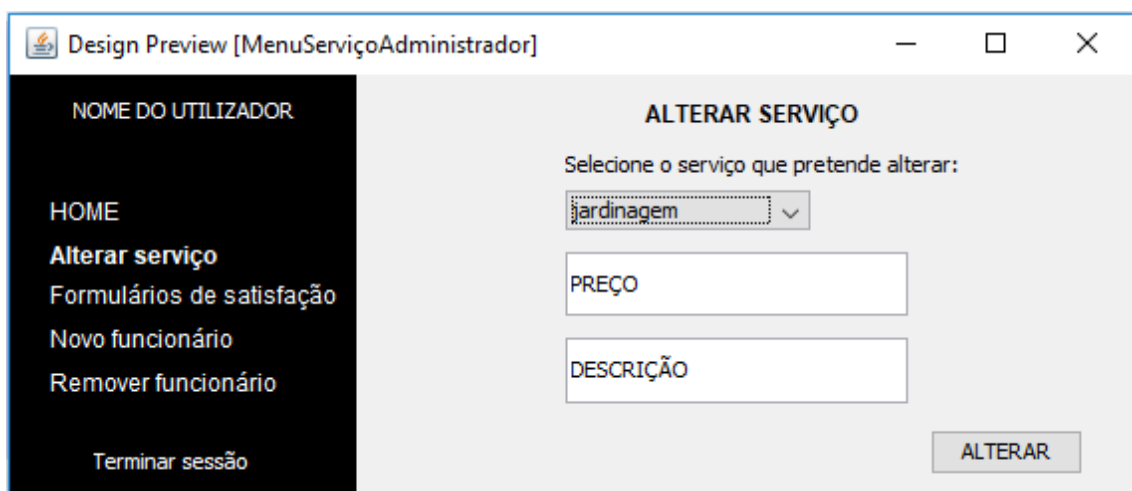


Figura 46 - Mockup Alterar serviço

Design Preview [MenuFormulariosAdministrador]

NOME DO ADMINISTRADOR

HOME

Alterar serviço

Formulários de satisfação

Novo funcionário

Remover funcionário

Terminar sessão

FORMULÁRIOS DE SATISFAÇÃO

Funcionário	Cartão de Cidadão	Pontuação	Sugestões

Figura 47 - Mockup Formulários de satisfação

Design Preview [MenuNovoFuncionarioAdministrador]

NOME DO ADMINISTRADOR

HOME

Alterar serviço

Formulários de satisfação

Novo funcionário

Remover funcionário

Terminar sessão

NOVO FUNCIONÁRIO

NOME

CARTÃO DE CIDADÃO

EMAIL

IBAN

MORADA

TELEMÓVEL

DATA DE NASCIMENTO (dd/mm/aa)

ESPECIALIDADE

ADICIONAR

Figura 48 - Mockup Novo Funcionário

De realçar que na figura 48 apresentam-se todos os dados necessários à criação de um novo funcionário no sistema. Para além disto, a especialidade permite identificar qual o ramo a que o funcionário pertencerá.

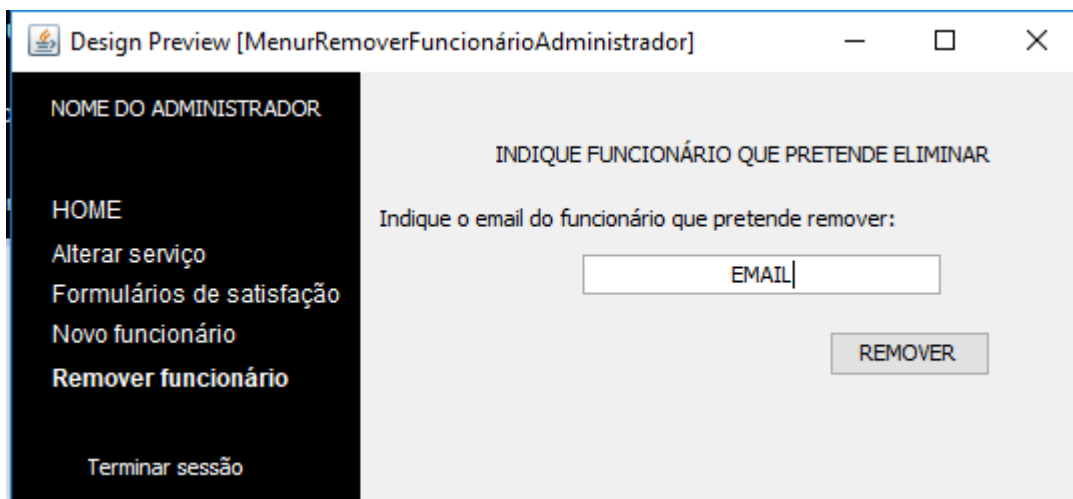


Figura 49 - Mockup Remover funcionário

2.8.2. Cliente

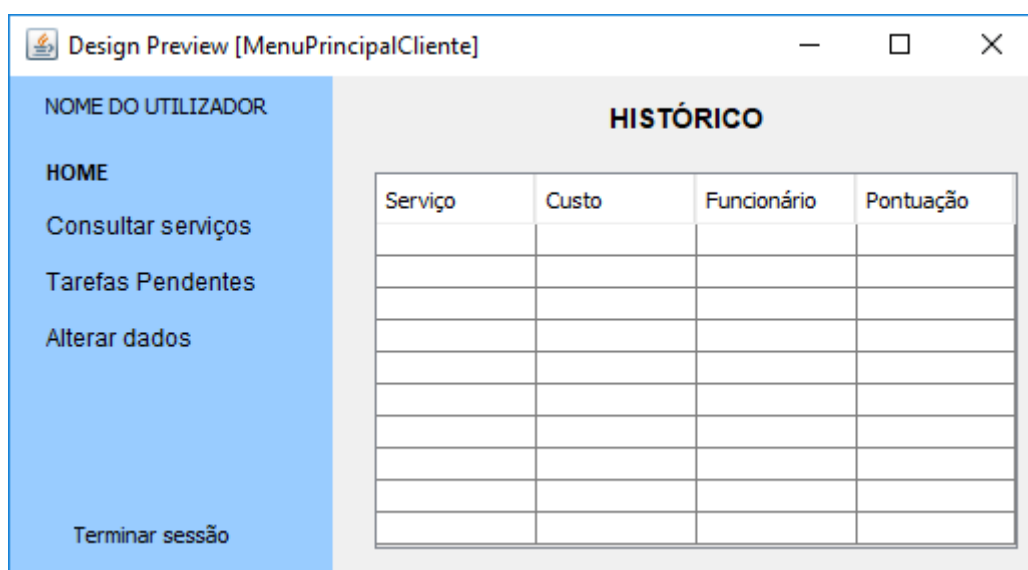


Figura 50 - Mockup Menu principal Cliente

No caso do cliente, o ecrã principal conterà o histórico dos serviços que este já usufruiu. Assim, para além de poder preencher formulários, contidos no separador “Tarefas Pendentes”, este pode ainda alterar dados e consultar serviços. É neste último separador que se dá a possibilidade ao cliente de os requisitar.

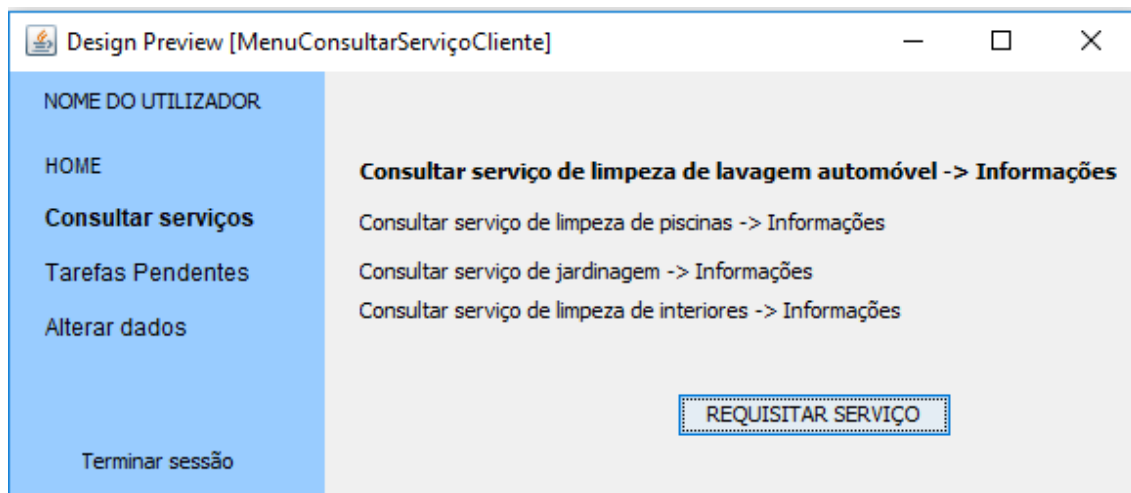


Figura 51 - Mockup Consultar serviço

Ao carregar no botão requisitar serviço, o menu apresentado é substituído pelo menu seguinte:

Figura 52 - Mockup Requisitar serviço lavagem automóvel

Dependendo do tipo de serviço pretendido, o formulário apresentado varia.

Figura 53 - Mockup Requisitar serviço, interiores, exteriores e piscina

A morada é indicada no registo para que o cliente não necessite constantemente de indica-lá aquando a realização de um serviço. No entanto, poderá requerer um serviço para uma outra casa pelo que estas opções são contempladas no formulário. Por fim um cliente poderá alterar dados relativos à conta em que se encontra.

Figura 54 - Mockup Alterar dados (cliente)

2.8.3. Funcionário

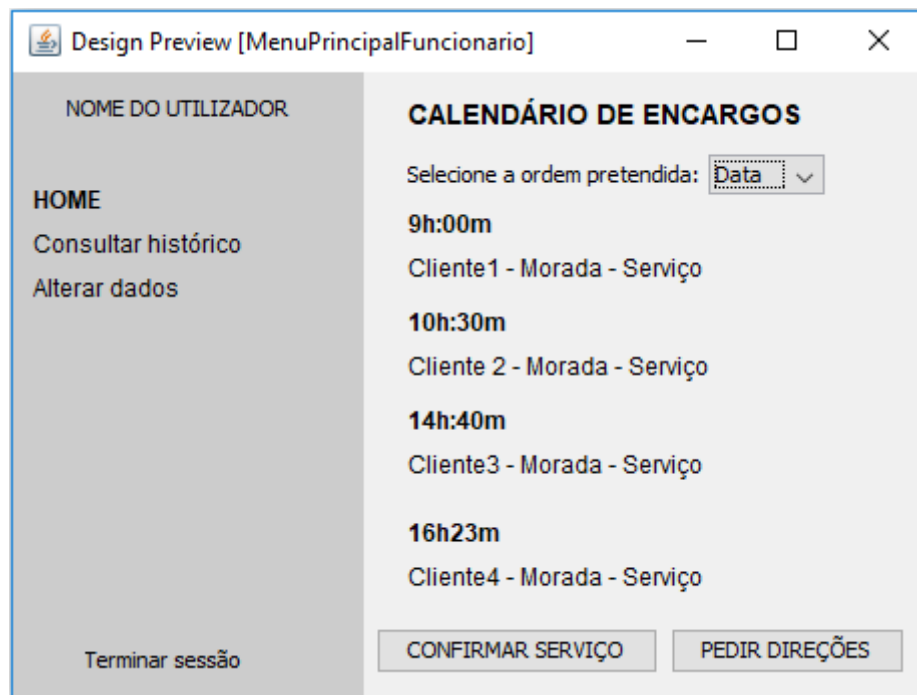


Figura 55 - Mockup Menu principal funcionário

No caso do funcionário, o ecrã de “Home” será relativo ao caderno de encargos, uma vez que, é uma das funcionalidades mais úteis da sua aplicação. Para além disto, aqui se apresentam duas funcionalidades imprescindíveis. “Confirmar serviço” onde é apresentado ao funcionário um formulário simples de confirmação do serviço e ainda, caso este ainda não tenha sido confirmado, o funcionário poderá “Pedir direções” sobre a morada onde atuará. Assim, o sistema GPS aparecerá com todas as informações disponíveis. Como seria de esperar o funcionário terá ainda a possibilidade de consultar o histórico dos trabalhos já realizados e alterar os dados relativos à sua conta. Ambos os menus são semelhantes aos já especificados para os restantes utilizadores.

2.9. Base de dados

A base de dados é um aspeto fulcral para a aplicação que pretendemos criar. Optamos pela utilização de uma base de dados relacional. De facto, Cleanica encontra-se num estado inicial, pelo que o reduzido número de utilizadores justifica a utilização deste tipo de modelo.

2.9.1. Identificação e caracterização das entidades

Nome da entidade	Descrição	Sinónimo	Ação/Ocorrência
Utilizador	Termo geral que descreve as características principais associadas a um utilizador.	-	Um utilizador pode ser um funcionário e/ou um cliente
Funcionário	Termo geral que refere as características associadas ao utilizador funcionário	Empregado	Um funcionário é um utilizador
Cliente	Termo geral que refere as características associadas ao utilizador cliente	-	Um cliente é um utilizador
Serviço	Termo geral que refere as características associadas a um serviço	Tarefa/Encargo	Um serviço contém vários funcionários
Serviços Realizados	Termo geral que refere as características associadas aos serviços que foram efetuados	Trabalho efetuado	Um serviço realizado refere-se a um cliente efetuado por vários funcionários
Formulário de realização	Termo geral que refere as características associadas aos formulários de realização	Formulário de confirmação de serviço	O formulário de realização possui um cliente e um funcionário
Formulário de Satisfação	Termo geral que refere as características associadas aos formulários de realização	Formulário de avaliação de serviço	O formulário de satisfação é relativo a um funcionário, por parte de um cliente

Tabela 1 - Identificação das entidades

2.9.2. Identificação e caracterização dos relacionamentos

Entidade	Multiplicidade	Relação	Entidade	Multiplicidade
Utilizador	0..1	Pode ser	Cliente	1
Utilizador	0..1	Pode ser	Funcionário	1
Cliente	0.. *	Está no	Formulário de Realização	1
Cliente	0.. *	Preenche	Formulário de Satisfação	1
Cliente	0.. *	Requisita	ServiçosRealizados	1
Funcionário	1	Tem	Serviço	0.. *
Funcionário	0.. *	Realiza	ServiçosRealizados	0.. *
Funcionário	0.. *	Preenche	Formulário de Realização	1
Funcionário	0.. *	Está no	Formulário de Satisfação	1

Tabela 2 - Caracterização dos relacionamentos

2.9.3. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos

Entidade	Atributos	Descrição	Dados	Null	Multi-valorado
Utilizador	Email	Email do Utilizador (Identifica unicamente um utilizador)	Variável até 50 caracteres	Não	Não
	Username	Username do Utilizador	Variável até 20 caracteres	Não	Não
	Password	Palavra Passe do Utilizador	Variável até 20 caracteres	Não	Não

	Estatuto	Estatuto do Utilizador (C/F/A)	1 Caracter	Não	Não
Cliente	Telemóvel	Telemóvel do Cliente	Inteiro entre 0 e MAXINT	Não	Não
	Morada	Morada do Cliente	Variável até 100 caracteres	Não	Não
Funcionário	CC	Cartão de cidadão de um Funcionário	Inteiro entre 0 e MAXINT	Não	Não
	Telemóvel	Telemóvel do Funcionário	Inteiro entre 0 e MAXINT	Não	Não
	IBAN	Número da conta bancária do Funcionário	Variável até 25 caracteres	Não	Não
	DataNascimento	Data de Nascimento do Funcionário	Date	Não	Não
	Morada	Morada do Funcionário	Variável até 100 caracteres	Não	Não
Serviço	idServiço	Identifica unicamente um serviço	Inteiro entre 0 e MAXINT	Não	Não
	Nome	Nome de um serviço	Variável até 15 caracteres	Não	Não
	Descrição	Descrição do serviço	Variável até 100 caracteres	Não	Não
	Preço	Preço do Serviço	Float (4)	Não	Não
	PreçoSProdutos	Preço de Serviço sem Produtos de Limpeza	Float (4)	Não	Não
	idServiçosRealizado	Identifica unicamente um serviço realizado/a realizar	Inteiro entre 0 e MAXINT	Não	Não

Serviços Realizados	Morada	Morada de um serviço realizado/a realizar	Variável até 100 caracteres	Não	Não
	Data	Data de um serviço realizado/a realizar	DateTime	Não	<u>Não</u>
	Preço	Preço de um serviço realizado/a realizar	Float (4)	Não	Não
	Dimensão	Dimensão do serviço realizado/a realizar (P/M/G)	1 Caracter	Não	Não
	Prods	Produtos de limpeza para o serviço realizado/a realizar (S/N)	1 Caracter	Não	Não
	AspectosImp	Alguns aspectos importantes sobre o serviço realizado/a realizar	Variável até 150 caracteres	Sim	Não
	Pendente	Se o serviço já foi realizado (S/N)	1 Caracter	Não	Não
	Duração	Duração Estimada do Serviço realizado	Inteiro entre 0 e MAXINT	Não	Não
Formulário de Satisfação	Id_FormulárioS	Identifica unicamente um formulário de satisfação	Inteiro entre 0 e MAXINT	Não	Não
	Pontuação	Pontuação atribuída a funcionário	Inteiro entre 0 e MAXINT	Sim	Não
	Sugestões	Sugestões sobre serviços futuros	Variável até 100 caracteres	Sim	Não
	Pendente	Indica se o formulário já foi ou não preenchido (S/N)	1 Caracter	Não	Não
	Id_FormulárioR	Identifica	Inteiro	Não	Não

Formulário de Realização		unicamente um Formulário de Realização	entre 0 e MAXINT		
	Duração	Duração do serviço realizado	Inteiro entre 0 e MAXINT	Não	Não
	Observações	Notas sobre o serviço realizado	Variável até 100 caracteres	Sim	Não

Tabela 3 - Identificação e associação dos atributos com as entidades

2.10. Modelo lógico

O modelo lógico trata-se de uma importante fase na modelação do sistema. Sendo assim, este foi desenvolvido em conformidade com ao que mais tarde pretendemos dar resposta. De facto, tivemos em consideração a forma como os dados seriam acedidos numa fase posterior. Para além disso, algumas decisões foram tomadas de acordo com os diagramas produzidos até então. Foi na conciliação destes dois factos que surgiu o seguinte modelo lógico².

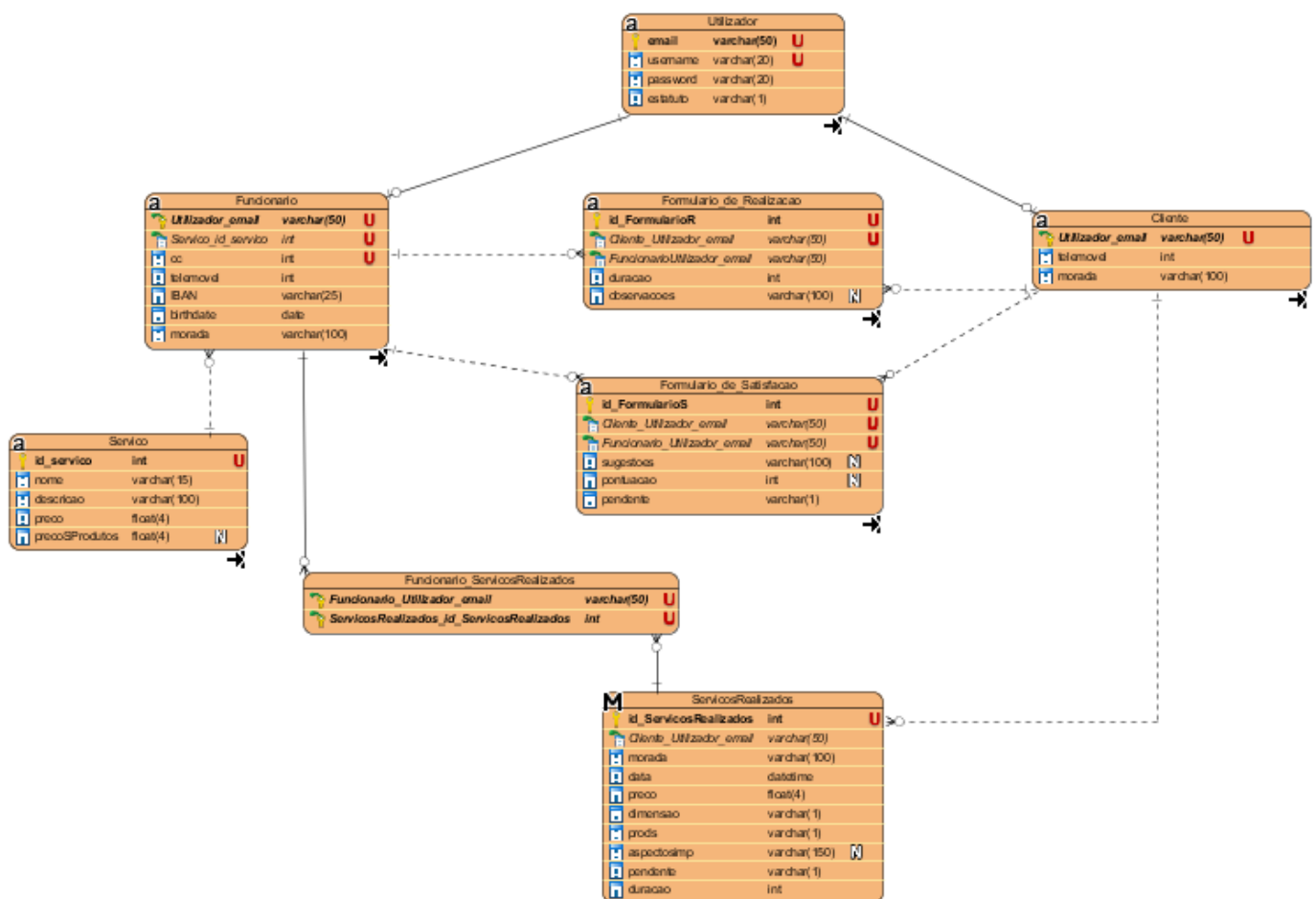


Figura 56 - Modelo lógico

² O modelo lógico surgiu após uma caracterização cuidada da base de dados. Procuramos seguir toda a metodologia recomendada, pelo que, o modelo concetual também foi desenvolvido e encontra-se em anexo. A base de dados foi criada desde raiz com a preocupação que o resultado final se encontrasse normalizado.

Decidimos criar uma tabela geral designada por “utilizadores”. A tabela “utilizadores” tem um relacionamento 1:1 com duas outras tabelas com o nome “Funcionário” e “Cliente” (entidades fracas). Esta decisão foi tomada pois cada um dos utilizadores possui atributos diferentes, para além disso, mais tarde seria benéfica a existência destas tabelas no manuseamento dos dados. Assim, o utilizador funcionário faz-se valer dos atributos de cartão de cidadão, email, telemóvel, IBAN, data de nascimento e morada. Sendo campos obrigatórios no registo deste, nenhum dos atributos poderá ser nulo. Como chave primária, optamos pelo email. Este é individual para cada trabalhador. Como chave candidata poderíamos ter o Cartão de Cidadão, no entanto, a definição do email como chave primária irá facilitar ao administrador a gestão de recursos humanos. De igual modo, existirá a tabela “Cliente”, esta possuirá o email como atributo, também este, chave primária. De facto, cada cliente tem um email que o identifica singularmente. A morada, por sua vez, terá de ser outro atributo necessário. Nenhum destes atributos poderá ser nulo. A tabela principal, identificada como “utilizador” não tendo qualquer atributo extra é identificada como sendo a tabela onde serão guardados os dados do administrador.

Um funcionário tem apenas um serviço, no entanto, um serviço possui vários funcionários. Nesse sentido, surge a tabela “Serviço”, esta tem como atributos o nome, a descrição e o preço dos serviços existentes. Decidimos adicionar um ID à tabela, que funcionará como chave primária, pois permite uma identificação única dos serviços. Todos os atributos são não nulos. Para além disso, o funcionário possui ainda um relacionamento 1:N para uma outra tabela, designada por “formulário de realização”. Esta possui como atributos, a duração e as observações (único que poderá ser NULL). A chave primária será novamente um ID por nós adicionado. Um cliente e um funcionário terão associados através de um relacionamento 1:N “formulários de satisfação”. Esta será uma tabela que terá atributos designados por sugestões (poderá tomar o valor de NULL) e pontuação (terá valor NULL, no momento de criação do formulário). Possuirá ainda um *varchar(1)* que permite identificar se o formulário se encontra ou não pendente (S ou N). Por fim, o facto do cliente poder requisitar serviços implica que estes possam ser guardados na base de dados, para que possa ser possível dar resposta a certas funcionalidades (como o caso do histórico, por exemplo). Assim, existirá uma tabela “ServiçosRealizados” que possuirá um relacionamento 1:N para a tabela Cliente e outra relação N:N para a tabela funcionário. Esta tabela de serviços realizados possuirá como atributos, a morada, data, preço, dimensão, produtos, aspetos simples e ainda um atributo que permite identificar se um serviço se encontra ou não pendente. Nenhum destes atributos pode adquirir o valor de NULL. Como chave primária, mais uma vez, destaca-se a criação

de um ID. Para além disso, realça-se apenas que como NULL apenas podemos ter os “aspetos importantes” caso o cliente não pretenda preencher este campo.

Os tamanhos dos diferentes atributos foram decididos em conformidade para ocupar o menor espaço possível, destacam-se, assim, os tipos, *varchar*, *int* e *float*.

2.11.Modelo físico

Depois de validarmos todo o modelo junto do nosso cliente decidimos fazer a transição do modelo lógico para o modelo físico. Uma vez que, a ferramenta predileta será o SQL Server, usamos a funcionalidade disponibilizada pelo *Visual Paradigm* para gerar o código necessário. Este poderá ser consultado na íntegra em anexo. Tivemos ainda em consideração a constante dos utilizadores no sistema e pensamos na criação de *views* para que estes apenas tenham acesso aos dados próprios das suas funcionalidades.

3. Implementação

3.1. Apresentação e caracterização geral da arquitetura do sistema

Na sua generalidade o sistema segue o que foi estipulado nas fases anteriores. Foram criados vários menus, de acordo com o utilizador que iniciava a sessão. Como poderemos verificar, o projeto segue o mockup descrito. Nesse sentido, é apresentada uma barra lateral que permite aceder facilmente a todas as funcionalidades que o sistema pretende dar resposta. No entanto, a modelação apresentada, ainda que geral, baseia-se em alguns aspetos na linguagem de programação *JAVA*, sendo que como o trabalho foi desenvolvido usando *asp.net*, existem alguns aspetos que diferem. Um desses foi desde logo a ligação à base de dados, muito mais intuitiva em *asp.net*. Apresentamos alguns exemplos, sendo que, os restantes podem ser encontrados em anexo.

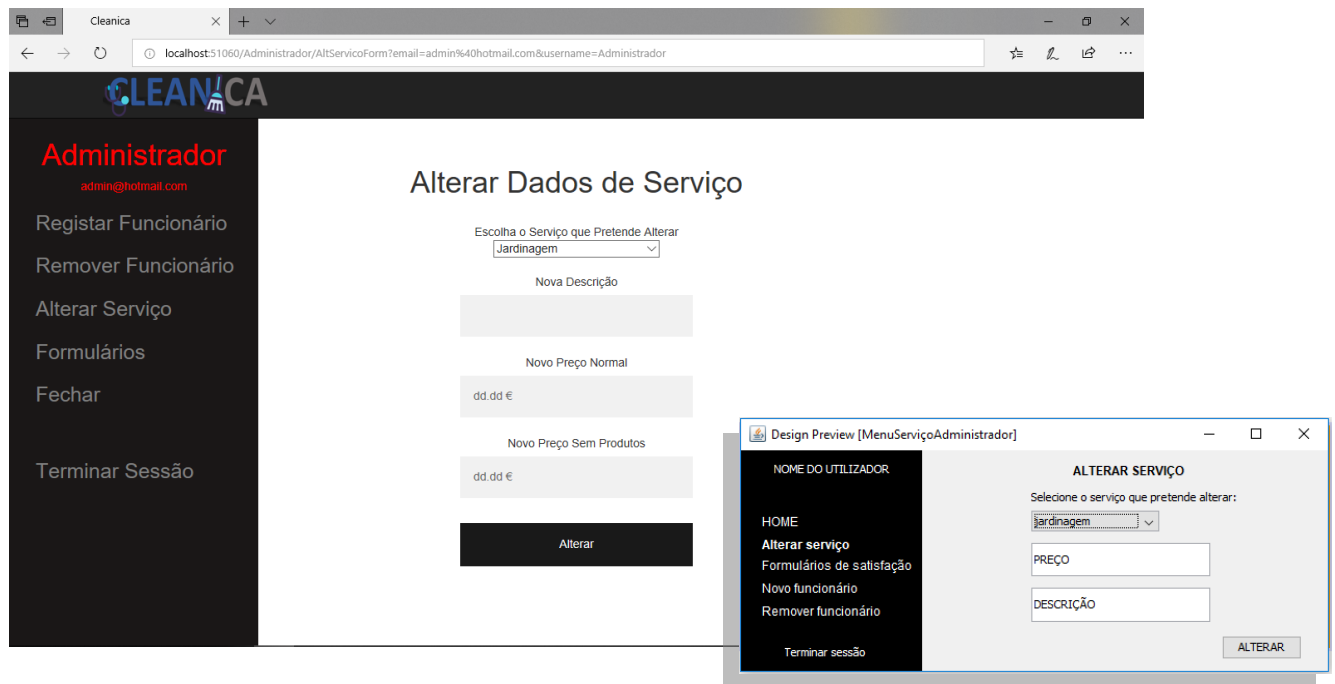
➤ Menu de Login

The screenshot displays the Cleanica web application's login and registration interface. The main page features a header with the 'CLEANICA' logo. Below the header, there are two columns. The left column is for registration, titled 'Ainda não está registado? Registe-se aqui em menos de 1 minuto.' It contains input fields for Email, Username, Password, Telemóvel, and Morada, followed by a 'Registar' button. The right column is for login, titled 'Já Está Registado? Faça o Login Aqui', and contains input fields for Email and Password, followed by a 'Login' button. A small inset window in the bottom right corner shows a design preview of the same interface.

Figura 57 - Implementação do menu de Login

3.1.1. Administrador

➤ Alterar dados de serviço



➤ Remover funcionário

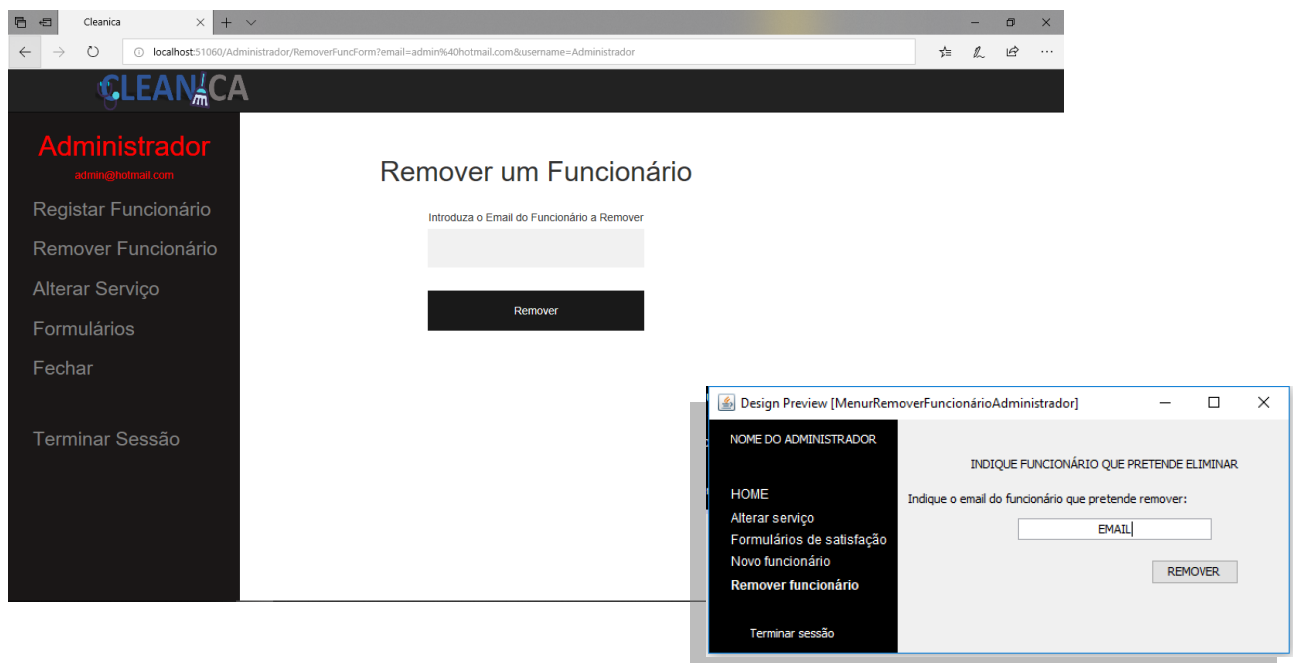


Figura 58 - Implementação do menu Alterar Dados de Serviço

3.1.2. Cliente

➤ Alterar dados

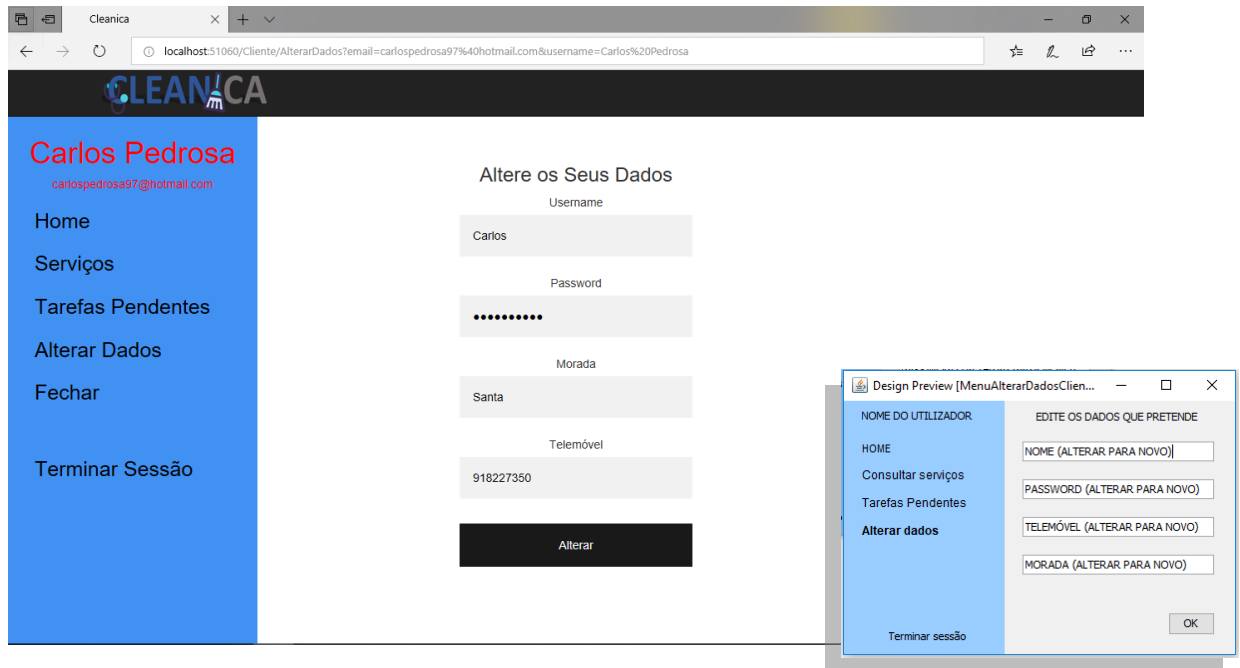


Figura 60 - Implementação do menu de alterar dados

➤ Requisitar serviço

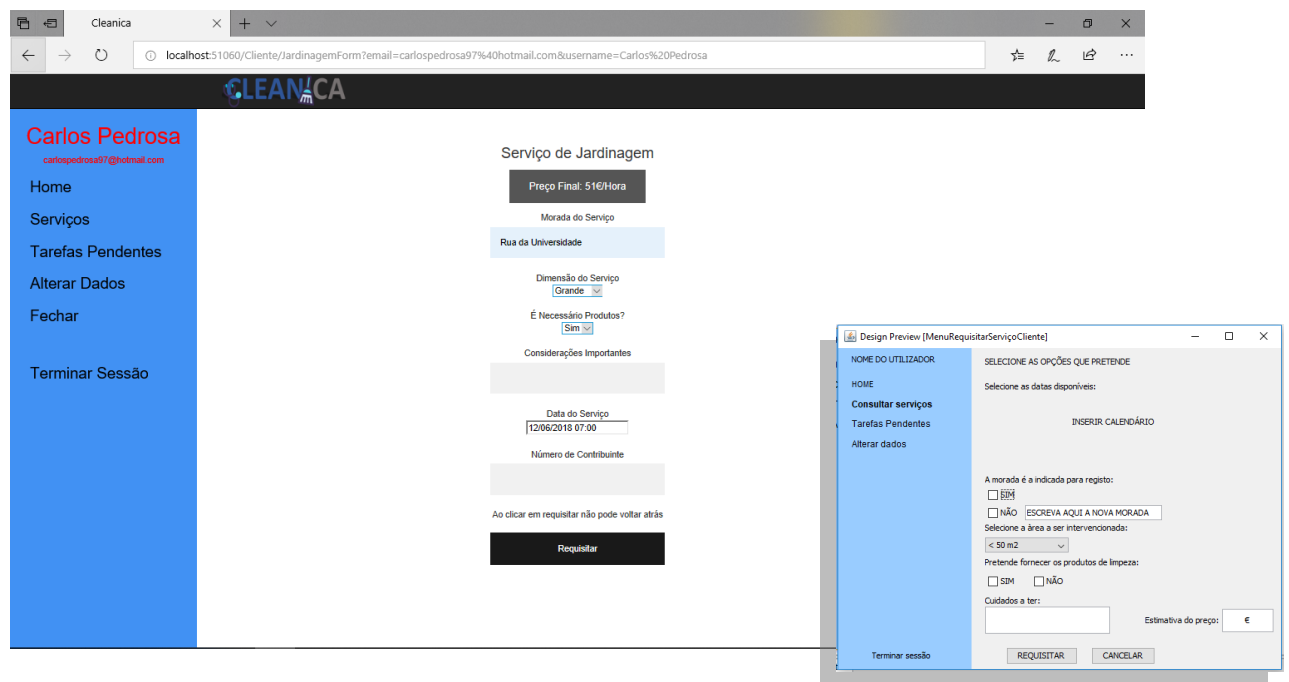


Figura 61 - Implementação do menu requisitar serviço

3.1.3. Funcionário

➤ Home

The screenshot shows the Cleanica web application interface. The browser address bar indicates the URL: `localhost:51060/Funcionario?email=cr7%40hotmail.com&username=Cristiano%20Ronaldo&info=Login%20Como%20Funcion%C3%A1rio%20Bem%20Sucedido`. The page header displays the Cleanica logo. The user's name, **Cristiano Ronaldo**, and email, `cr7@hotmail.com`, are shown in the top left. A green box highlights the message "Login Como Funcionário Bem Sucedido". The main heading is "Serviços a Realizar". Below this is a table with the following data:

Data	Morada	Cliente	Preço	Clique para Preencher
10/06/2018 09:00:00	Primeira Avenida	rodrigosporguista@hotmail.com	15 €	Preencher

The sidebar menu on the left contains the following items: Home, Histórico de Serviços, Alterar Dados, Fechar, and Terminar Sessão. A design preview window titled "Design Preview [MenuPrincipalFuncionario]" is open on the right, showing a menu structure with "HOME" (Consultar histórico, Alterar dados) and "CALENDÁRIO DE ENCARGOS" (Seleção a ordem pretendida: Data). The calendar lists four service entries: 9h:00m, 10h:30m, 14h:40m, and 16h:23m, each associated with a client and a service. The preview window also includes buttons for "Terminar sessão", "CONFIRMAR SERVIÇO", and "PEDIR DIREÇÕES".

Figura 62 - Implementação do menu Home

De forma geral, podemos concluir que o resultado final foi de encontro à modelação do projeto efetuada na fase anterior.

3.2. Descrição dos diversos módulos

A implementação do projeto segue o modelo *Model-view-controller* (MVC). Assim, consideramos que o *software* desenvolvido se encontra separado em três camadas. A camada *Model* é responsável pela leitura e escrita dos dados. A camada *View* tem como função a exibição dos dados, sendo que no nosso caso, a linguagem usada foi *HTML*. Por último, a camada *Controller* é responsável por todas as requisições dos utilizadores. Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, pelo que esta é a principal vantagem da utilização deste tipo de modelo.

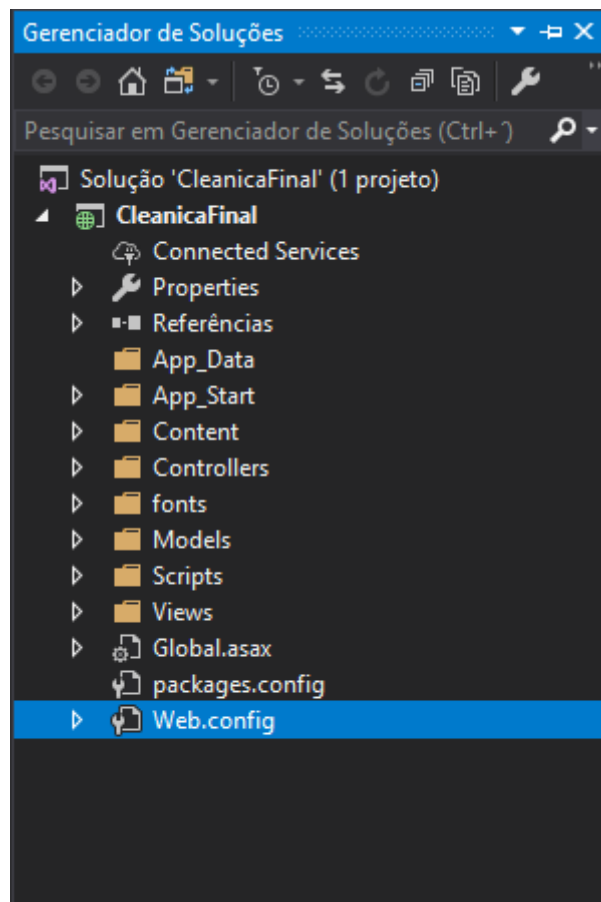


Figura 63 - Divisão do trabalho segundo Model-View-Controller

3.3. Plano de desenvolvimento e distribuição do trabalho

Como já referido na primeira fase, utilizamos a ferramenta *project* disponibilizada pela *Microsoft* para planejar todo o trabalho. Neste sentido, uma vez que o grupo de projeto é extenso foi possível dividir algumas tarefas de modo a economizar tempo bastante útil. Para além disso, este planeamento permitiu uma estimativa aproximada do custo da mão de obra em todo projeto. Assim, de seguida apresentaremos o diagrama de *GANTT* final (com a segunda e terceira etapa).

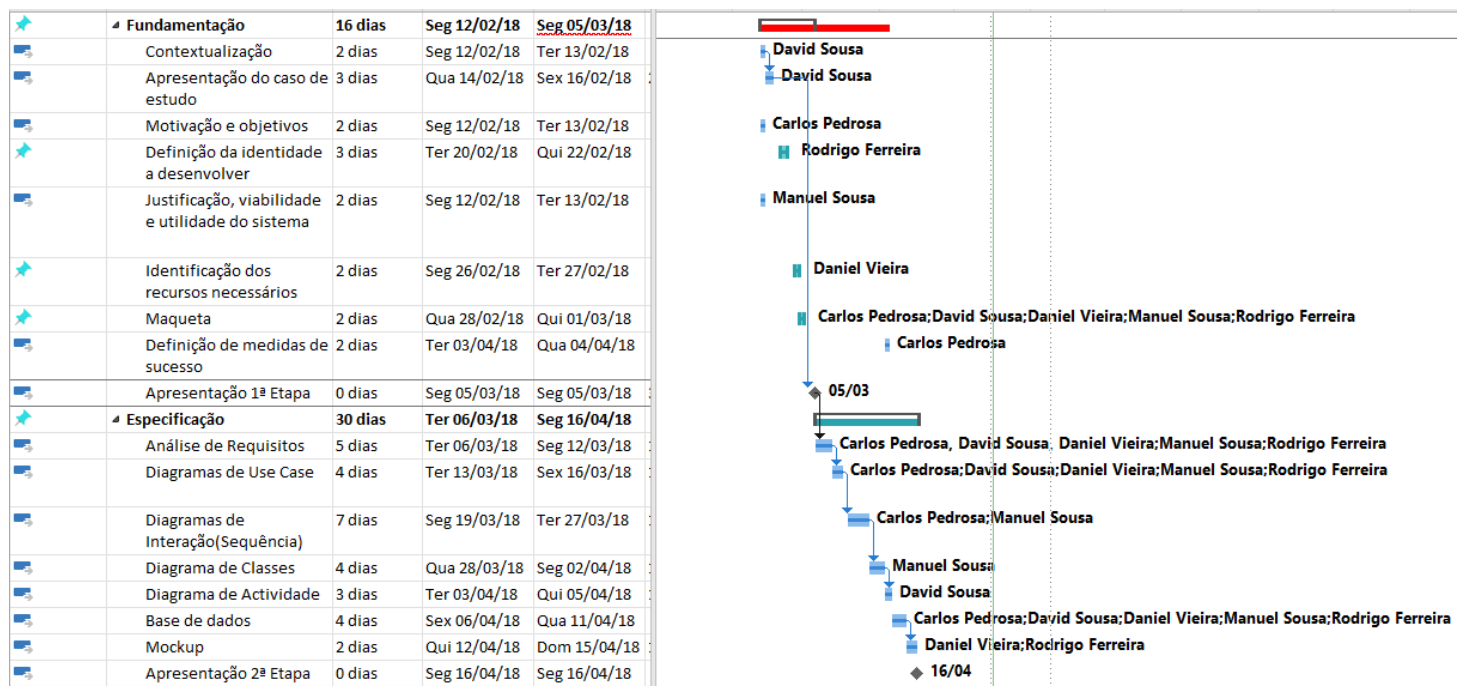


Figura 65 - Planeamento da primeira e segunda etapa

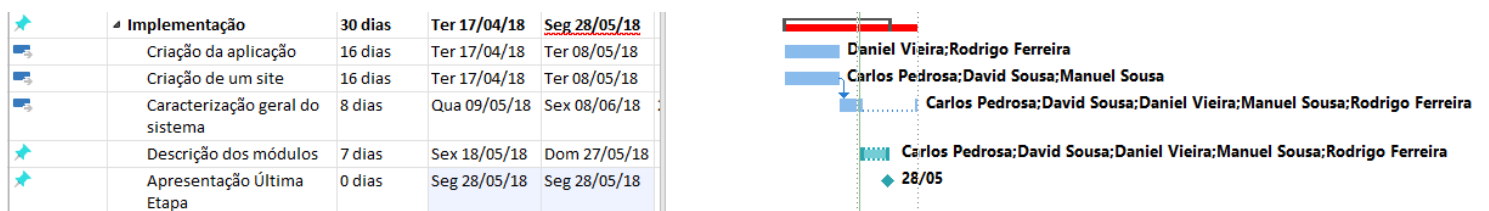


Figura 64 - Planeamento da terceira etapa

Fazendo uma estimativa do preço, sendo que cada trabalhador recebe 8€/h e considerando que cada dia tem oito horas, teremos que foi dispendido em todo o projeto um total de aproximadamente 15680 €. Em conclusão podemos dizer que procuramos seguir o plano estabelecido, no entanto, alguns desvios foram efetuados. Apesar de tudo, este revelou-se bastante útil, na medida em que permitiu organizar todo o trabalho que necessitava de ser efetuado.

3.4. Ferramentas utilizadas e implementação das várias funcionalidades

A aplicação foi desenvolvida com recurso a ferramentas da Microsoft, entre estas se destaca o *Visual Studio* e *SQL Server*. No entanto, de modo, a ir de encontro aos requisitos tivemos de recorrer a algumas API externas que possibilitaram a implementação de todos os recursos.

3.4.1. Localização

Sendo a localização um dos muitos requisitos do trabalho recorreremos ao redireccionamento para o *website* da Microsoft, o *BING Maps*, que nos permitiu obter a posição de um funcionário. Desta forma, foi possível calcular o caminho mais curto para o local onde iria ser realizado o serviço.

3.4.2. Serviço de *mailing*

Sempre que um pagamento é efetuado, de acordo com os requisitos é necessário que seja enviado um email de confirmação com a correspondente fatura. Para dar resposta a este requisito, uma vez mais, recorreremos ao uso de uma API externa, *System.Net.Mail*, também disponibilizada pela Microsoft que tem como objetivo a interação com um serviço de *mailing*.

3.5. Validação do software desenvolvido

Depois de efetuados todos os procedimentos necessários decidimos reunir-nos junto do nosso cliente para, deste modo, validarmos o modelo desenvolvido. Demonstramos todas as potencialidades do nosso projeto, indicando ao dono de *Cleanica* tudo o que poderia fazer com a seu novo *website*. Efetivamente, este indicou que o trabalho ia ao encontro ao que inicialmente tinha pedido e que todos os propósitos tinham sido cumpridos.

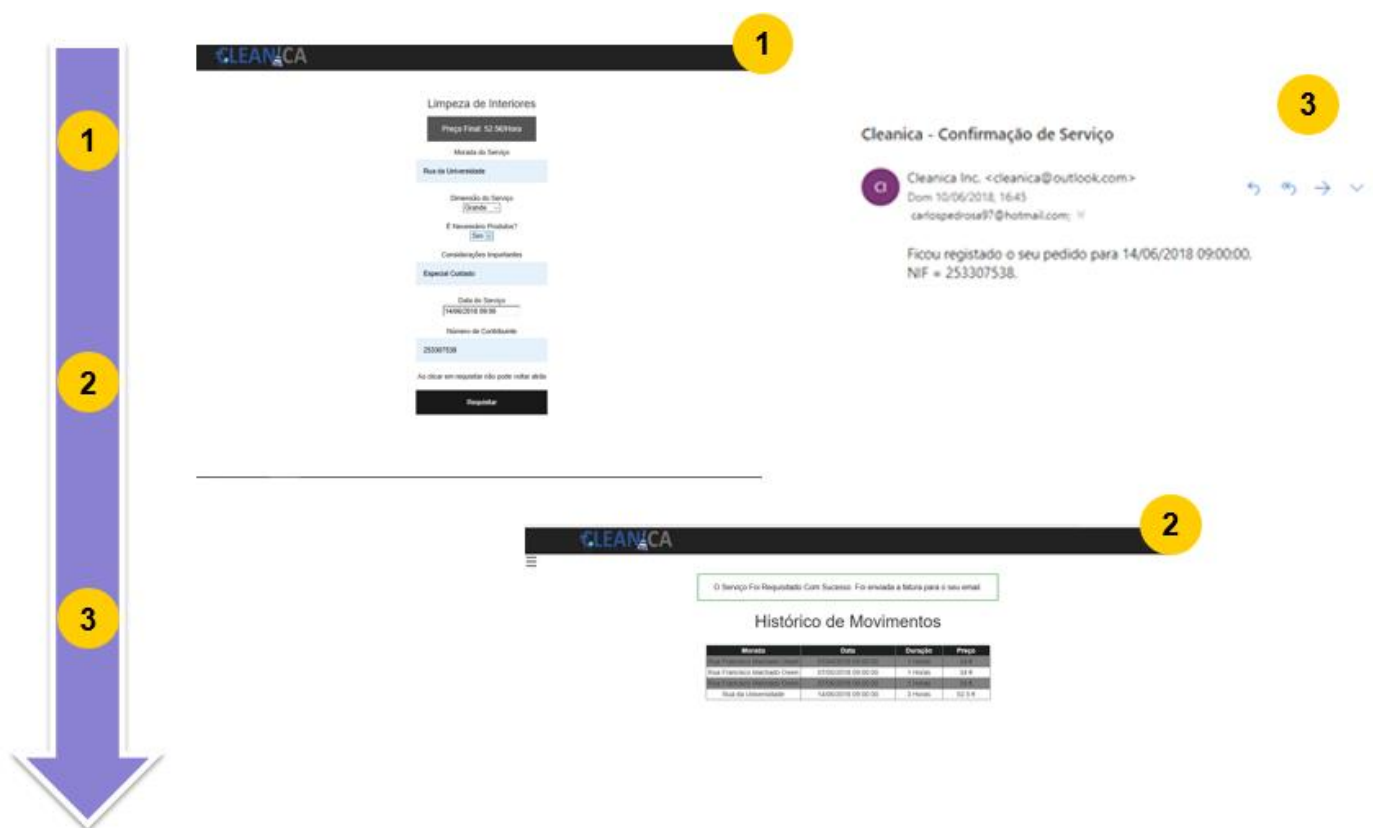


Figura 66 - Esquematização da requisição de um serviço

Através do diagrama acima é possível perceber todos os passos dados pelo/atraves sistema para que a requisição de um serviço cumpra todos os objetivos.

4. Conclusões e perspectivas de trabalho futuro

De acordo com o objetivo do projeto consideramos que este foi atingido, uma vez que, foi possível dar resposta a todos os requisitos pretendidos pelo cliente. No entanto, ao longo do projeto foram surgindo alguns desafios que na sua maioria foram ultrapassados.

De facto, o presente relatório encontra-se dividido por três etapas que seguimos para que fosse possível obter um resultado integral, conciso e suscetível de ser utilizado de forma prática. Nesse sentido, depois de feita a contextualização, toda a modelação que consideramos necessária foi desenvolvida. O principal desafio com que nos deparamos, nesta etapa, foi na escolha de quais seriam os melhores modelos a implementar. No entanto, apresentamos aqueles que à partida seriam mais úteis e percetíveis. Para além disso, o grupo foi por vezes obrigado a recuar, para que a definição dos requisitos fosse a mais correta possível. Contudo, estes recuos permitiram tornar todo o trabalho mais coerente e consistente.

Na fase respetiva à implementação, a principal dificuldade foi na utilização de ferramentas “novas”, nunca utilizadas pelos elementos do grupo. Contudo, uma análise prévia do que poderia ser feito e uma interação contínua com este *software* permitiram ultrapassar todos os obstáculos. Para além disso, o diagrama de *GANTT* também não foi cumprido na sua totalidade, pelo que este facto dificultou um pouco todo o trabalho. Concluimos, desta forma, que um trabalho como este deveria ser tratado com mais antecedência, de modo a, prevenir eventuais erros.

Em suma, consideramos que todos os objetivos foram cumpridos na medida em que o trabalho desenvolvido foi implementado segundo a modelação efetuada nas fases anteriores. Assim, todo o projeto foi essencial para um aprimoramento das nossas aptidões relativamente aos conteúdos da UC.

Referências

- <https://www.consumerbarometer.com/en/trending/?countryCode=PT&category=TRN-NOFILTER-ALL> acedido no dia 23/02/2018 pelas 15h16m;
- <https://repositorio.iscte-iul.pt/handle/10071/3164> acedido no dia 23/02/2018 pelas 15h40m;
- Thomas Connolly, TC, Carolyn Begg, CB, (2005), *Database System – A Pratical Approach to Designm Implementation, and Managements*, 4ªedição, Harlow: Pearson Education Limited.

Lista de Siglas e Acrónimos

GPS	<i>Global Positioning System</i>
DSS	Diagrama de Sequência de Sistemas
API	<i>Application Programming Interface</i>
HTML	Hypertext Markup Language
UC	Unidade Curricular

Anexos

I. Anexo 1 – Entrevista a “Cleanica”

1ª Pergunta: Porque é que recorreu aos nossos serviços?

R.: Recorri aos vossos serviços porque pretendo mudar a forma como a minha empresa funciona. De facto, a nossa empresa é bastante embrionária e apesar de termos os nossos clientes gostaríamos de ampliar a área de atuação e mudar a forma como contactamos com as pessoas. Pensamos por isso, sendo que é um assunto bastante em voga, contar convosco para a construção de uma aplicação web ou até mesmo para uma aplicação móvel.

2ª Pergunta: O que pretende que a aplicação faça?

R.: Sinceramente, como objetivos primários gostaríamos que aplicação permitisse transmitir os nossos serviços de forma mais interativa. Era importante que conciliasse os nossos recursos e que estes fossem destacados de acordo com o local de atuação. Efetivamente, apenas pretendemos deixar de usar a marcação por chamadas pois o elevado número destas torna-se bastante difícil de gerir. Pretendemos algo automático.

3ª Pergunta: Pretende alguma funcionalidade extra na aplicação de que fala?

R.: Sim, gostaria que fosse possível aos clientes pagarem a fatura através da aplicação, bem como, gostaria que fosse a aplicação que se encarregasse de notificar os trabalhadores dos serviços que estes terão de efetuar. A nossa empresa usa um método inovador relativamente aos funcionários. Estes não têm de estar fisicamente num local pelo que estão espalhados por várias áreas. Assim sempre que algum é chamado, é necessário ligar da empresa para este, ou reencaminhar a chamada, sendo que se o funcionário não poder ir imediatamente efetuar o trabalho será necessário fazer outro tipo de gestão. Como pode ver algo bastante confuso. É isso que pretendemos mudar.

4ª Pergunta: Tem alguma preferência a nível de interface ou qualquer tipo de função?

R.: Não, apenas que usem o logótipo da empresa. De resto, fica à vossa responsabilidade qualquer funcionalidade extra que pretendam implementar, desde que, fique dentro do orçamento e que tudo fique pronto dentro do prazo limite.

5ª Pergunta: Uma última pergunta, como funciona a sua empresa relativamente aos produtos que são necessários para a realização dos serviços?

R.: Geralmente nós garantimos os produtos necessários, sempre que um colaborador necessita de novos produtos o funcionário adquire os produtos necessários, enviando, posteriormente a fatura. No entanto, se o cliente preferir pode ser este a facultar os produtos necessários por sua conta e risco. O preço variará de acordo com a opção tomada. Enviaremos uma tabela com os preços dos mais variados serviços, penso que será útil e facilitará o desenvolvimento da aplicação.

II. Anexo 2 – Tabela de preços

Serviços	Com produtos	Sem produtos
Limpeza de automóveis	6€/hora	4€/hora
Limpeza de piscinas	5,10€/hora	3,10€/hora
Limpeza de interiores	7,15€/hora	5,15€/hora
Limpeza de exteriores (jardins)	8,50€/hora	6,50€/hora

Tabela 4 - Tabela de preços disponibilizada pela empresa

III. Diagramas de Sequência

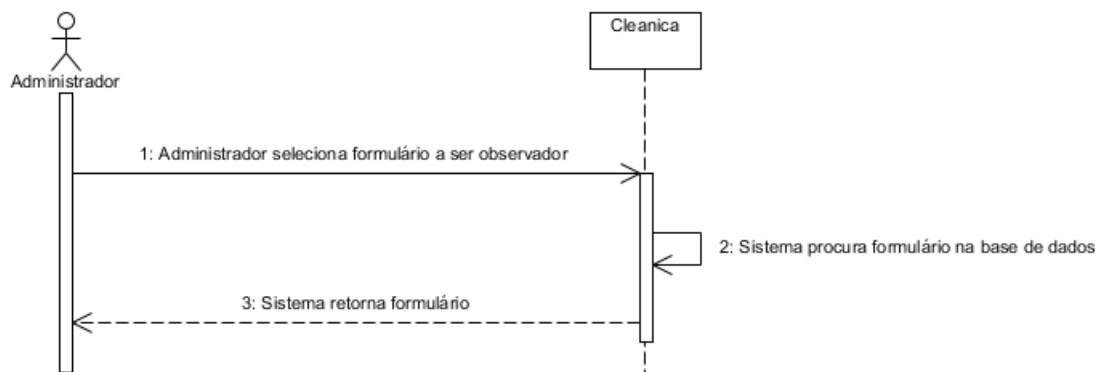


Figura 67 - Diagrama de sequência (Administrador): consultar formulário de satisfação

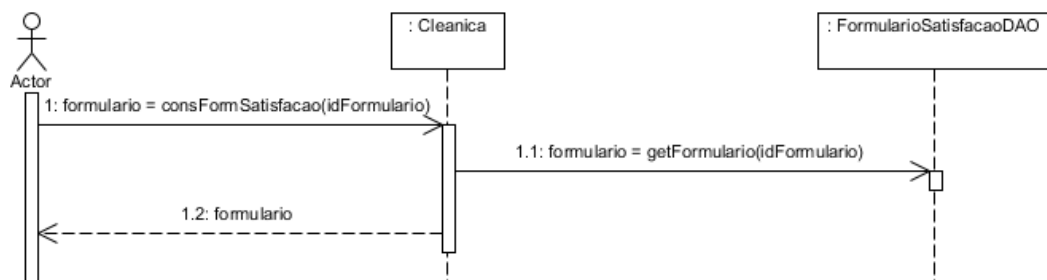


Figura 68 - Especificação do diagrama de sequência (Administrador): consultar formulário de satisfação

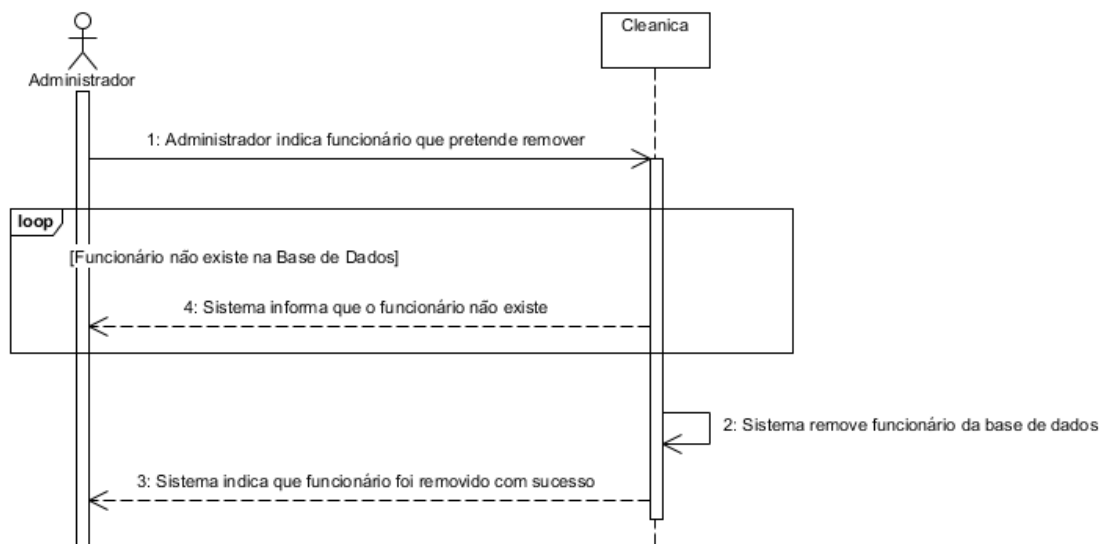


Figura 69 - Diagrama de sequência (Administrador): remover funcionário

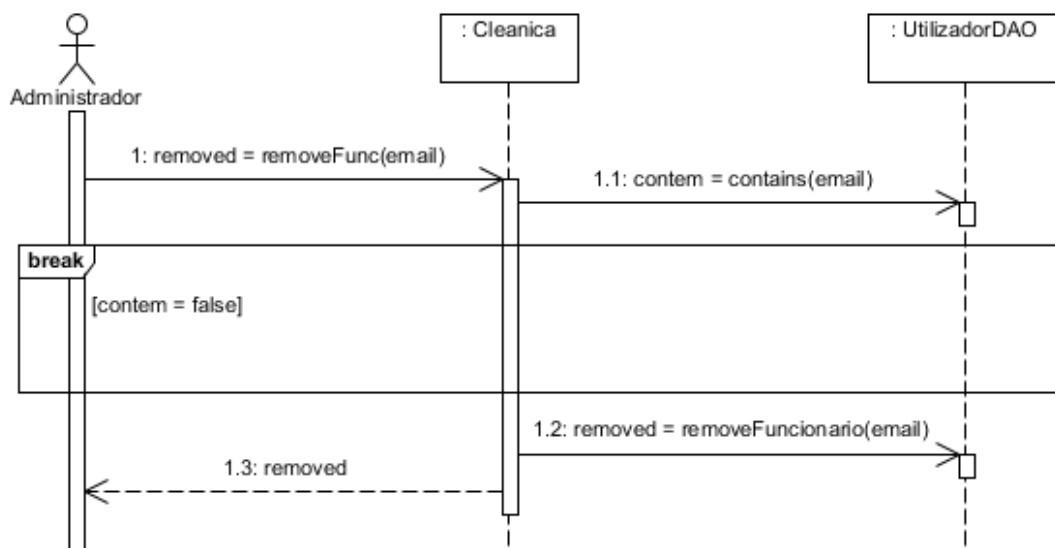


Figura 70 - Especificação do diagrama de sequência (Administrador): remover funcionário

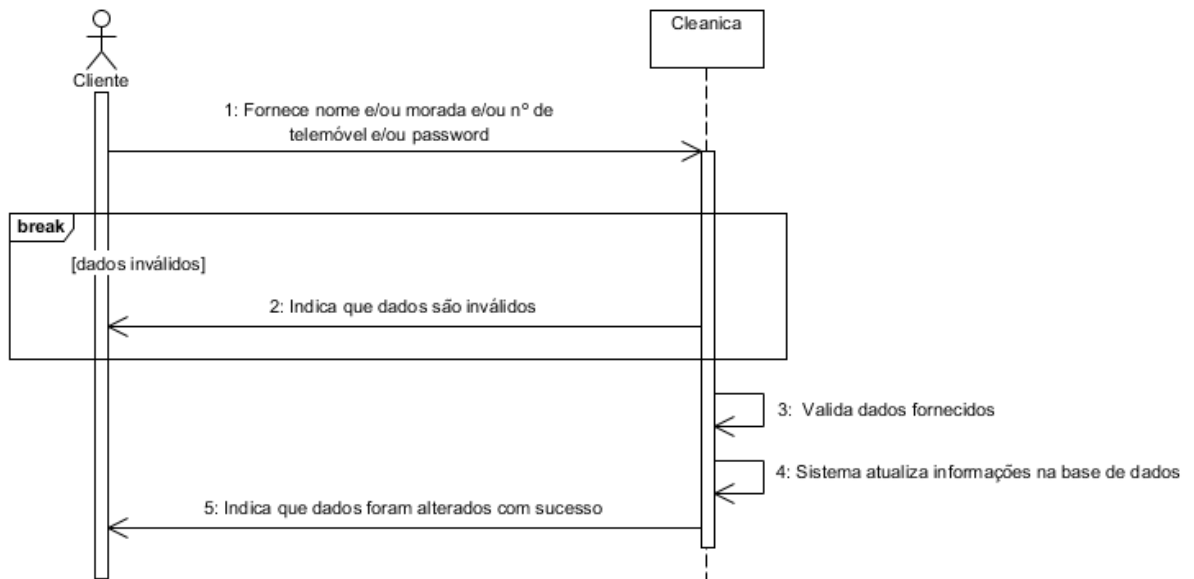


Figura 71 - Diagrama de sequência (cliente): alterar dados

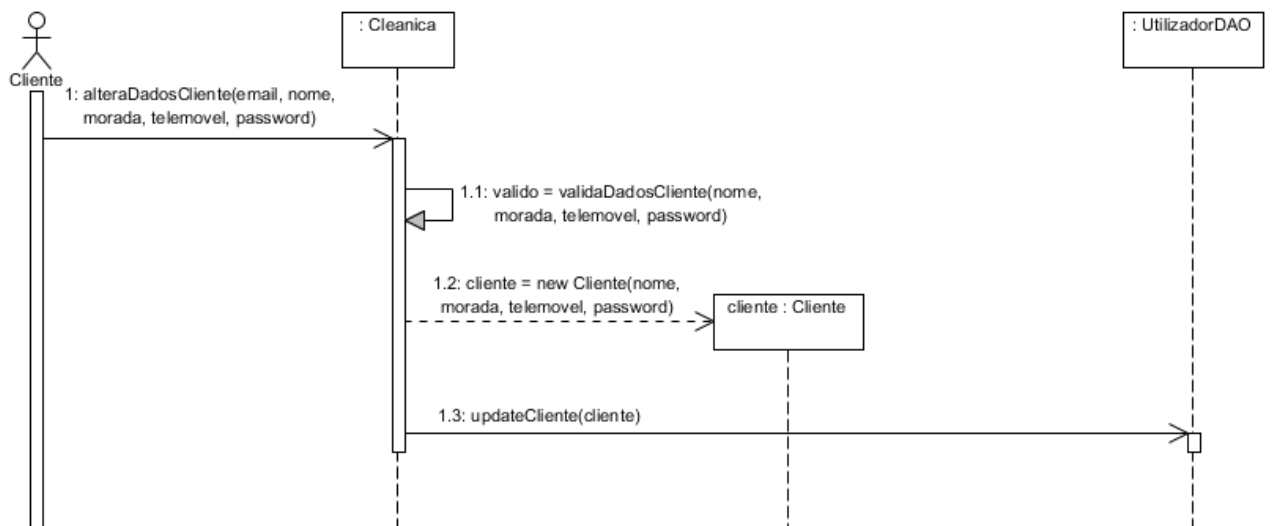


Figura 72 - Especificação do diagrama de sequência (cliente): alterar dados

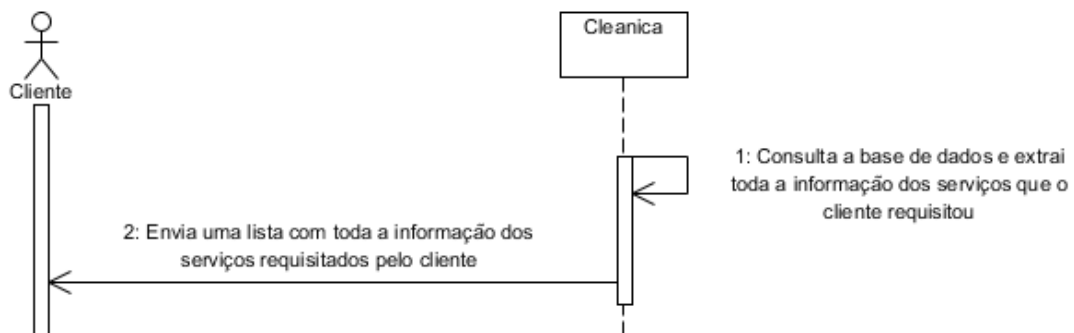


Figura 73 - Diagrama de sequência (cliente): consultar histórico

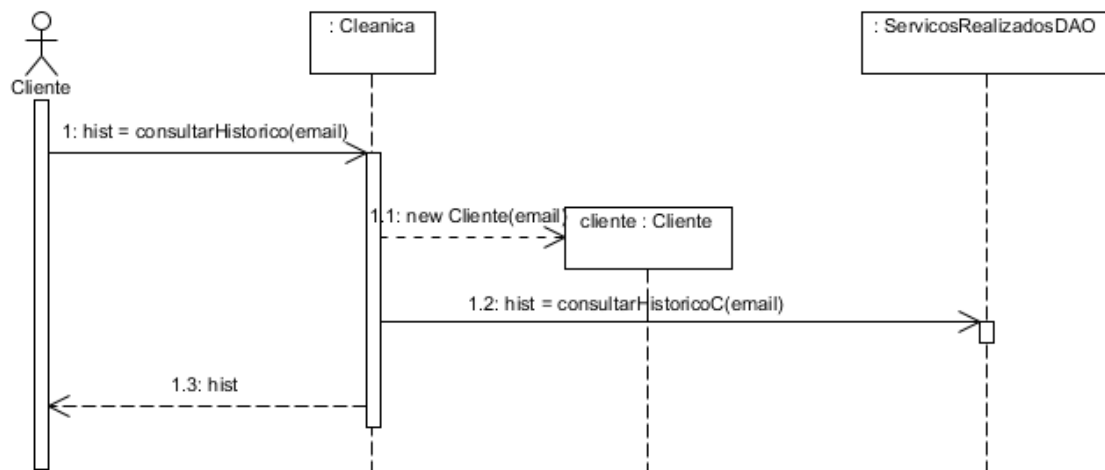


Figura 74 - Especificação do diagrama de sequência (cliente): consultar histórico

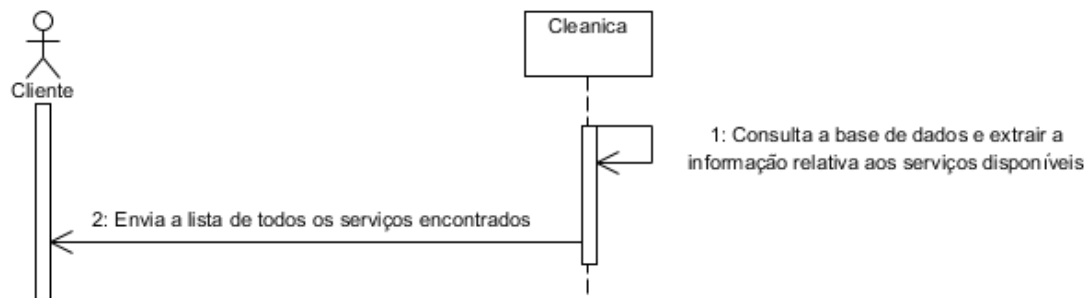


Figura 75 - Diagrama de sequência (cliente): consultar serviço

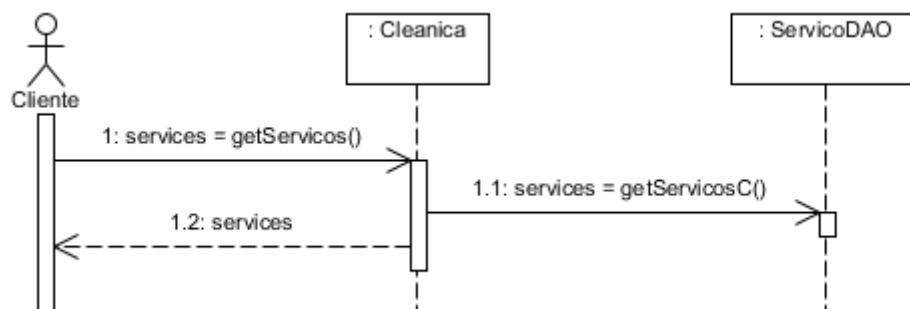


Figura 76 - Especificação do diagrama de sequência (cliente): consultar serviço

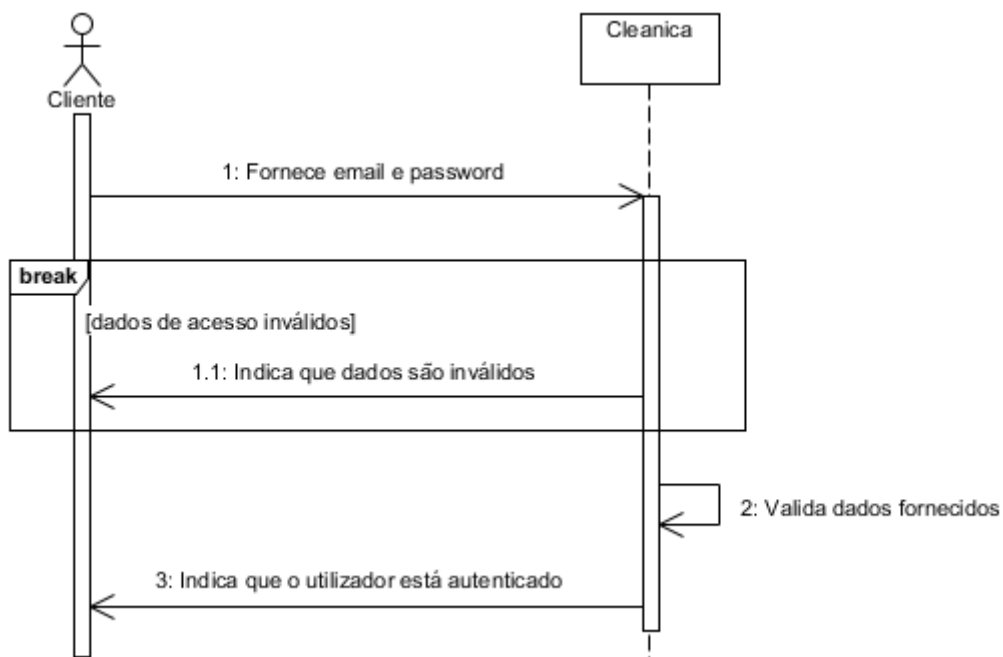


Figura 77 - Diagrama de sequência: efetuar Login

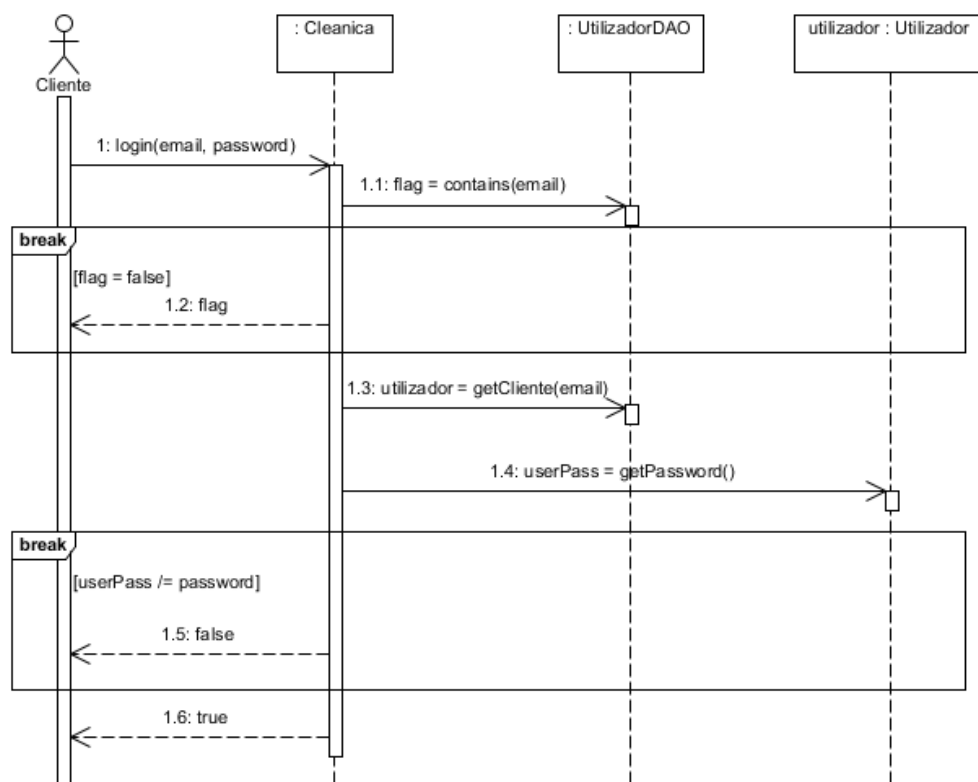


Figura 78 - Especificação do diagrama de sequência: efetuar Login

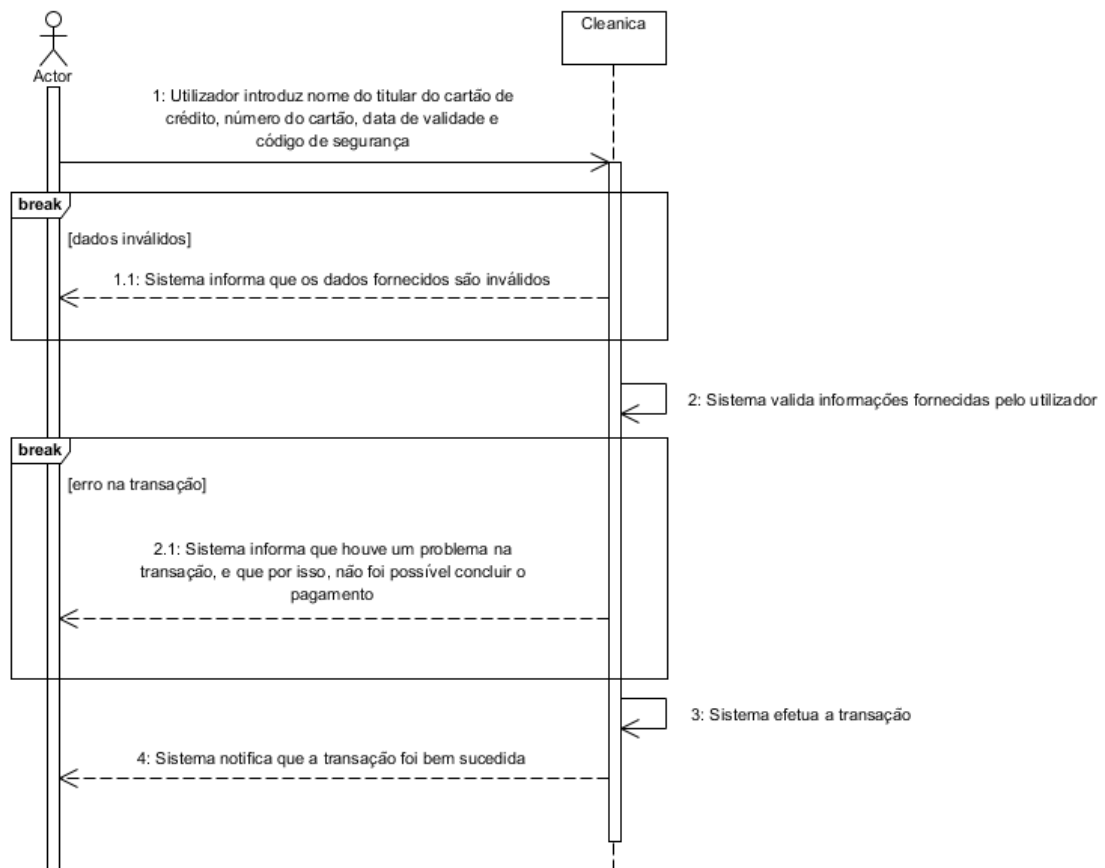


Figura 79 - Diagrama de sequência: efetuar pagamento

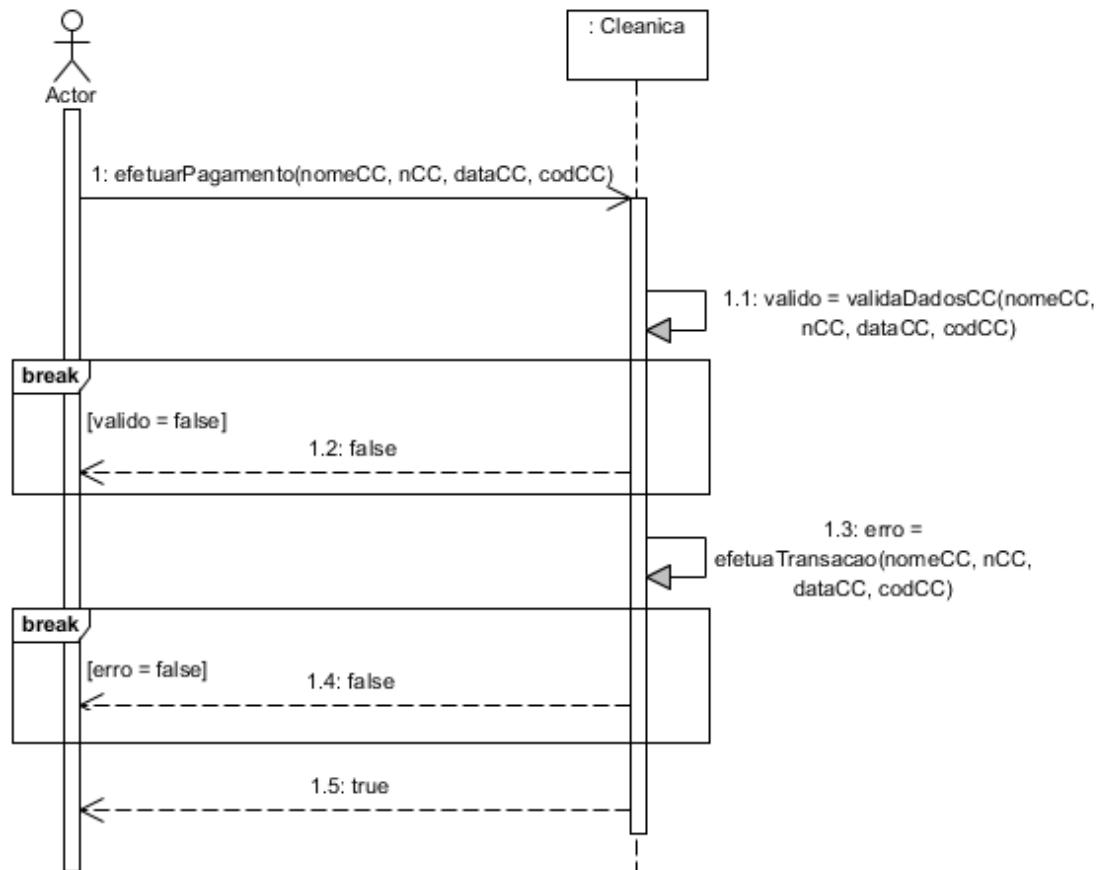


Figura 80 - Especificação do diagrama de sequência: efetuar pagamento

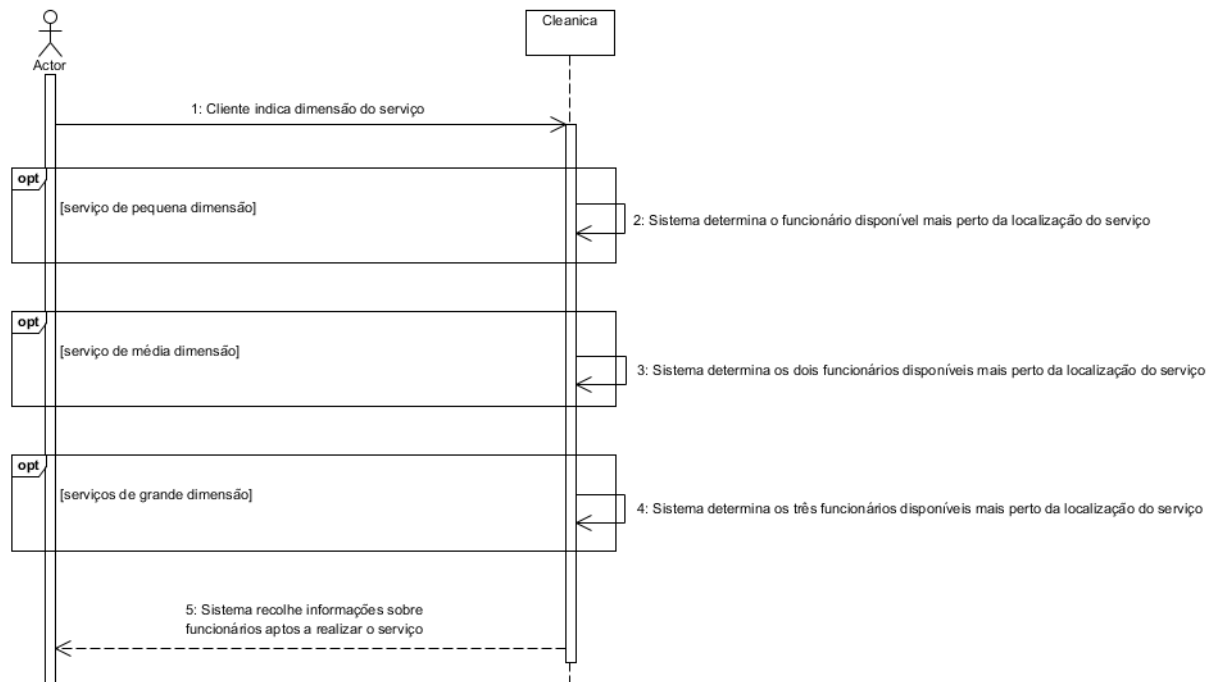


Figura 81 - Diagrama de sequência: escolher funcionário(s)

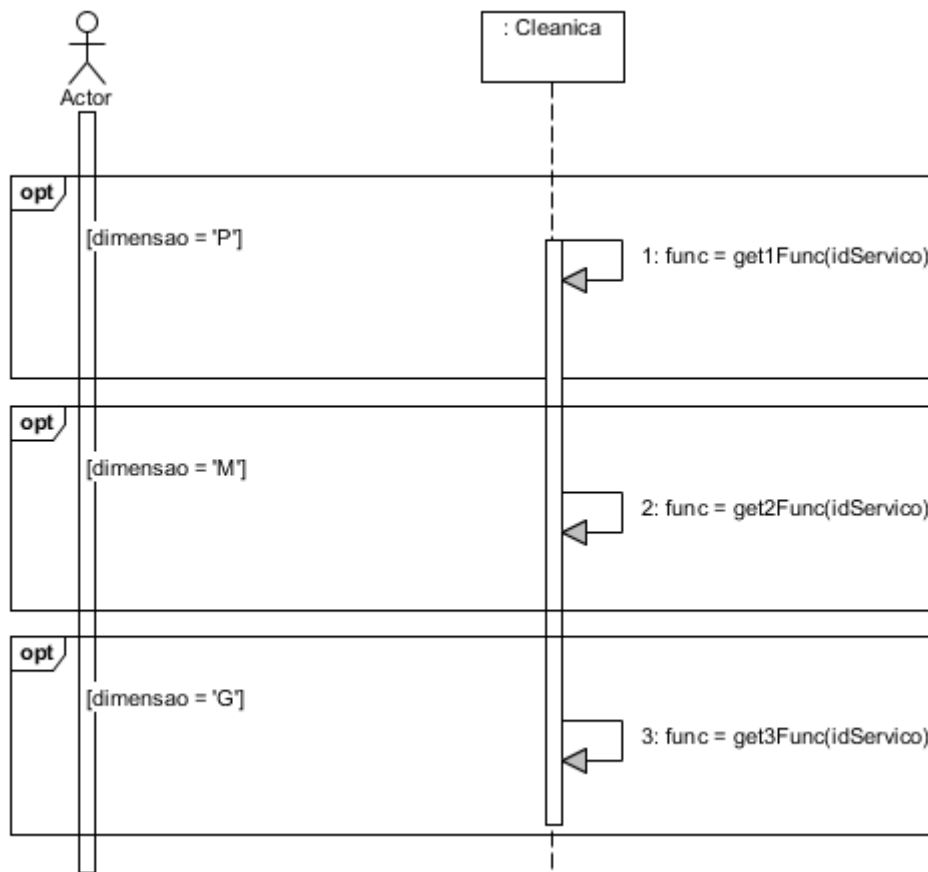


Figura 82 - Especificação do diagrama de sequência: escolher funcionário(s)

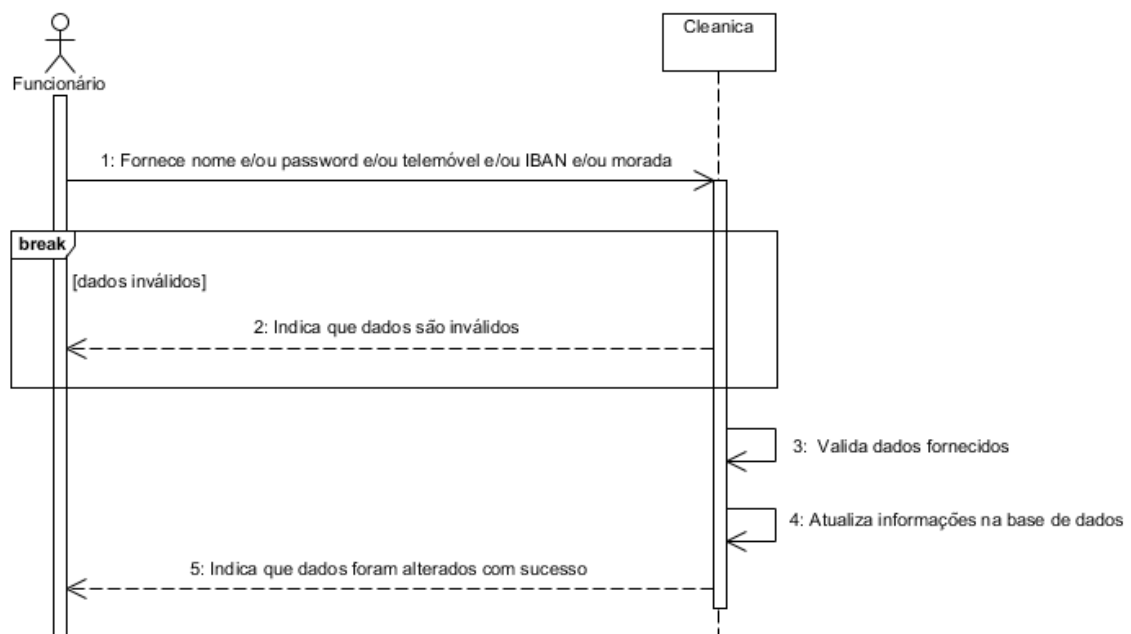


Figura 83 - Diagrama de sequência (funcionário): alterar dados

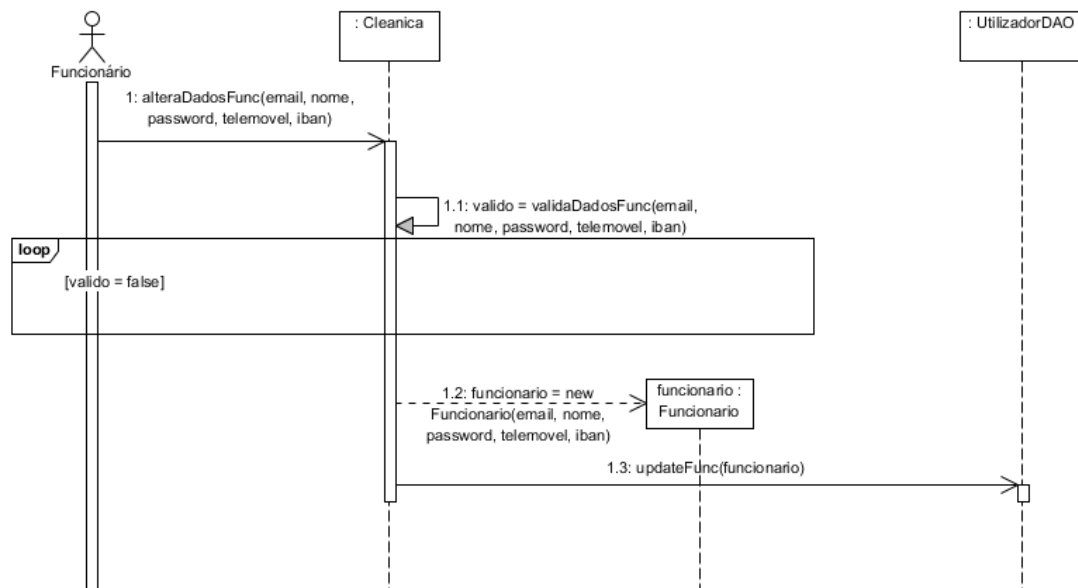


Figura 84 - Especificação do diagrama de sequência (funcionário): consultar serviço

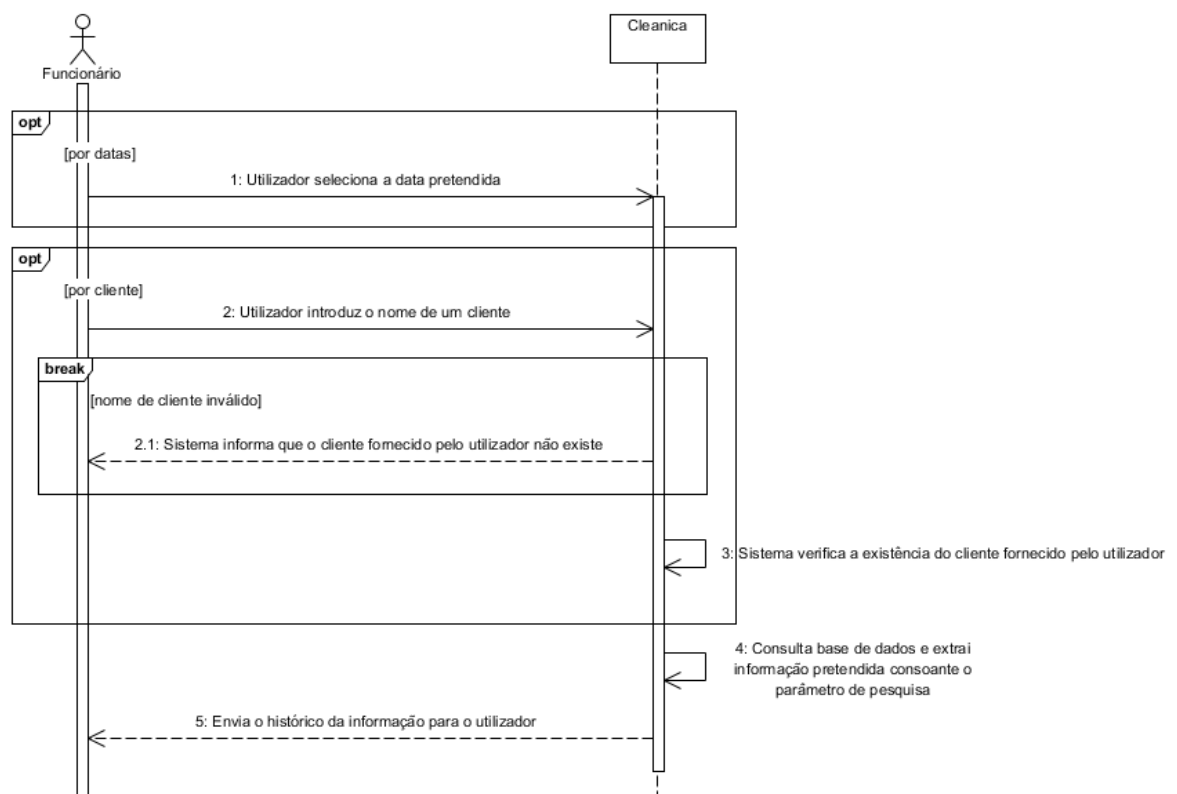


Figura 85 - Diagrama de sequência (funcionário): consultar histórico de encargos

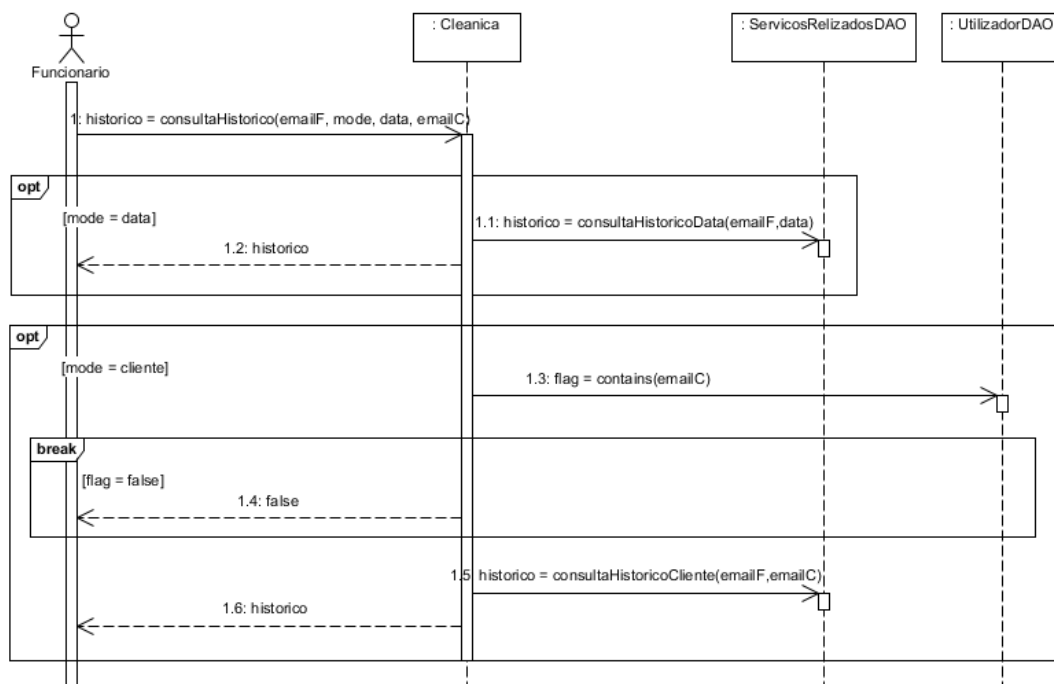


Figura 86 - Especificação do diagrama de sequência (funcionário): consultar histórico de encargos

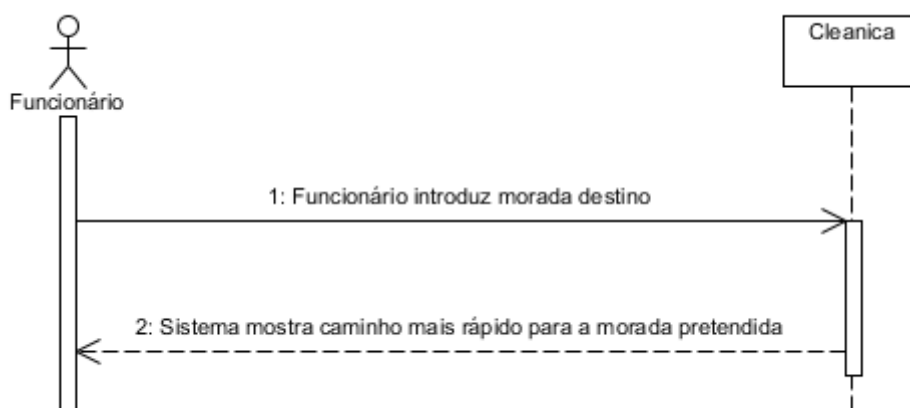


Figura 87 - Diagrama de sequência (funcionário): pedir direções

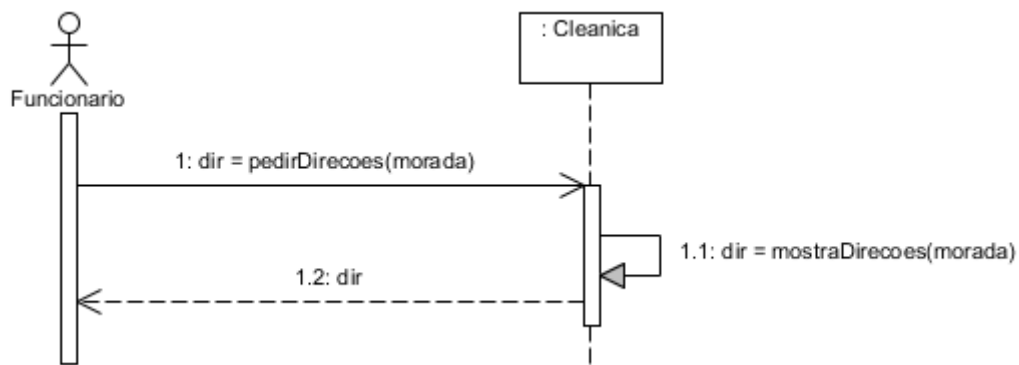


Figura 88 - Especificação do diagrama de sequência (funcionário): pedir direções

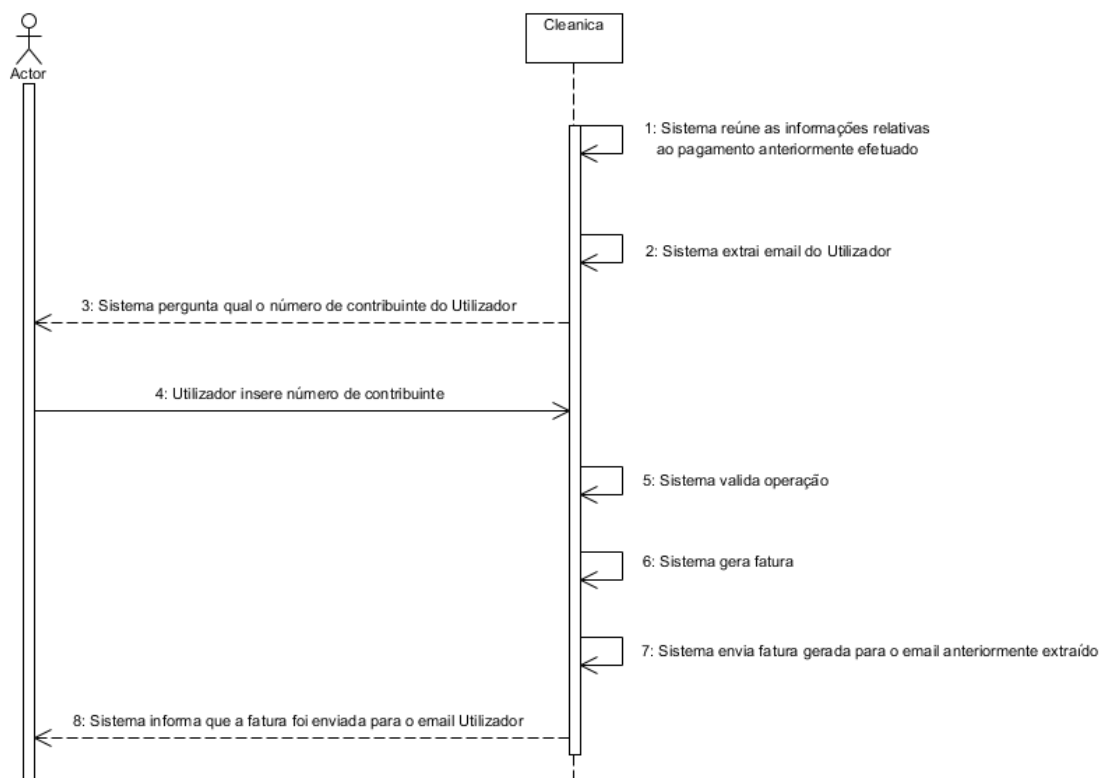


Figura 89 - Diagrama de sequência (funcionário): gerar fatura

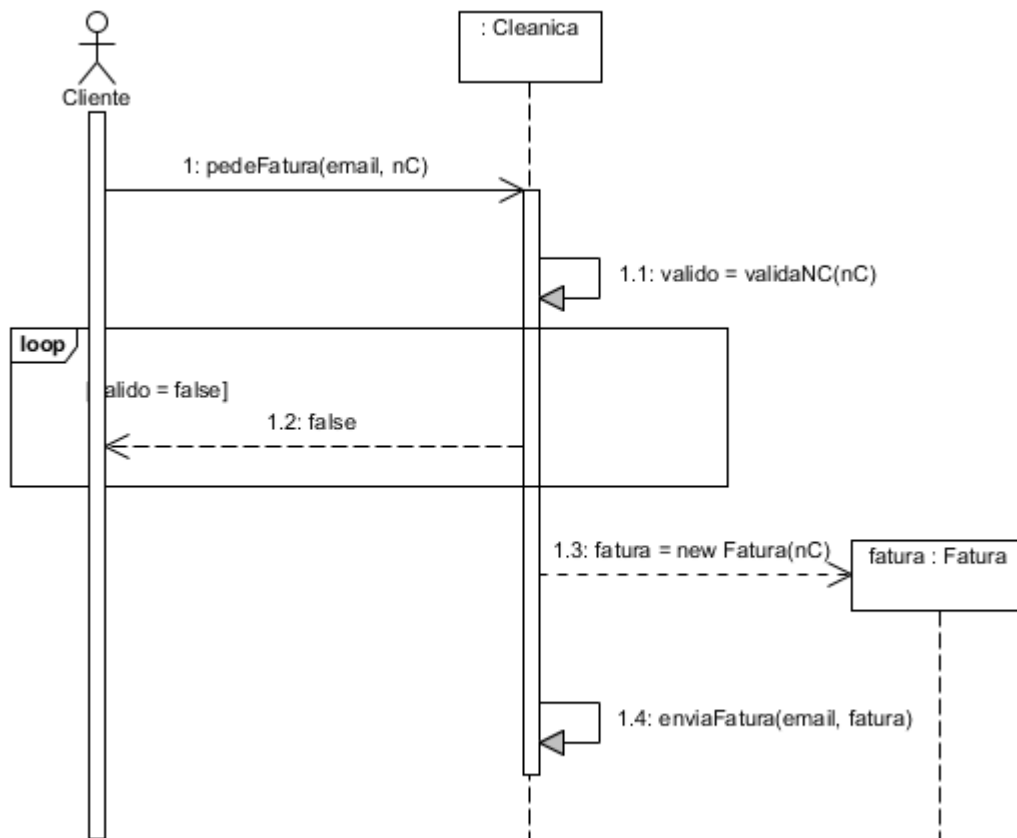


Figura 90 - Especificação do diagrama de sequência (funcionário): gerar fatura

IV. Modelo Concetual

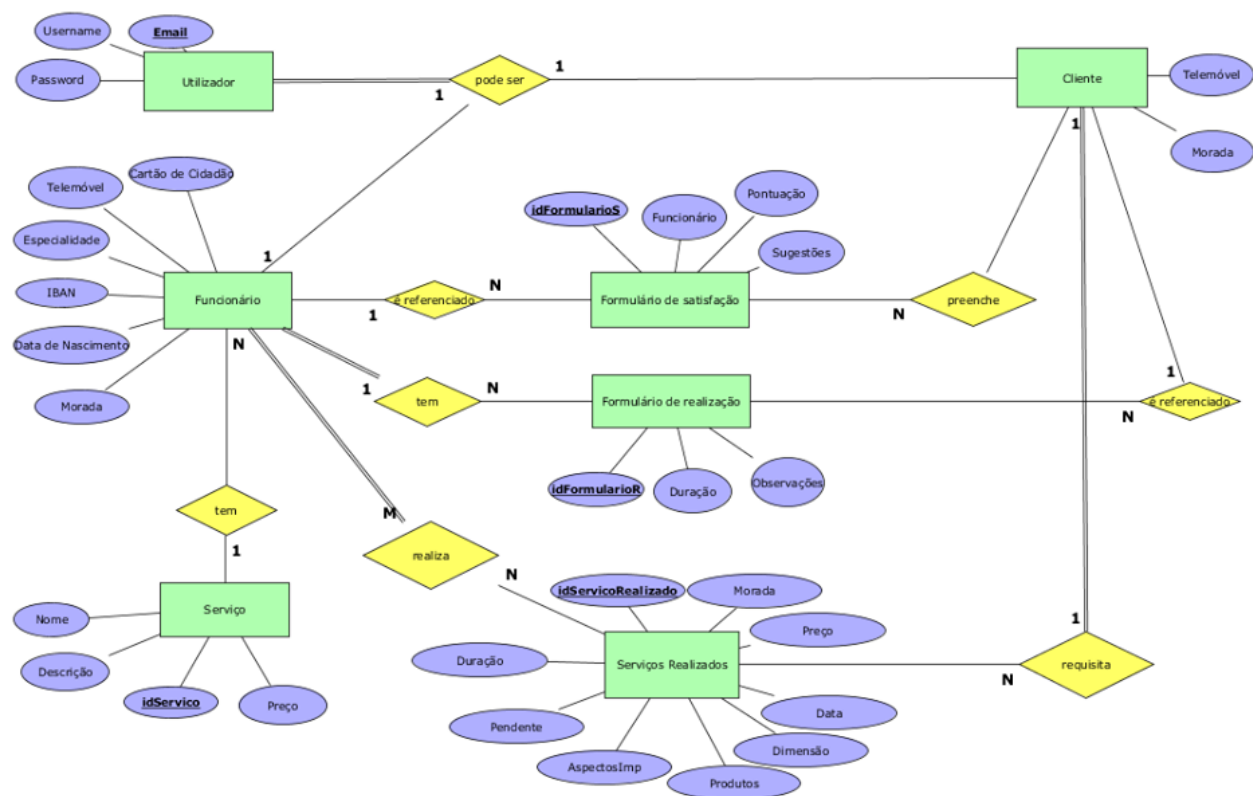


Figura 91 - Modelo Concetual

V. Modelo físico no *SQL Server*

```
CREATE TABLE Cliente (  
  Utilizador_email varchar(50) NOT NULL,  
  telemovel      int NOT NULL,  
  morada         varchar(100) NOT NULL,  
  PRIMARY KEY (Utilizador_email));  
  
CREATE TABLE Formulario_de_Realizacao (  
  id_FormularioR      int IDENTITY NOT NULL,  
  Cliente_Utilizador_email  varchar(50) NOT NULL UNIQUE,  
  FuncionarioUtilizador_email varchar(50) NOT NULL,  
  duracao             int NOT NULL,  
  observacoes         varchar(100) NULL,  
  PRIMARY KEY (id_FormularioR));  
  
CREATE TABLE Formulario_de_Satisfacao (  
  id_FormularioS      int IDENTITY NOT NULL,  
  Cliente_Utilizador_email  varchar(50) NOT NULL UNIQUE,  
  Funcionario_Utilizador_email varchar(50) NOT NULL UNIQUE,  
  sugestoes           varchar(100) NULL,  
  pontuacao           int NULL,  
  pendente            varchar(1) NOT NULL,  
  PRIMARY KEY (id_FormularioS));  
  
CREATE TABLE Funcionario (  
  Utilizador_email  varchar(50) NOT NULL,  
  Servico_id_servico int NOT NULL UNIQUE,  
  cc               int NOT NULL UNIQUE,  
  telemovel        int NOT NULL,  
  IBAN              varchar(25) NOT NULL,  
  birthdate         date NOT NULL,  
  morada            varchar(100) NOT NULL,  
  PRIMARY KEY (Utilizador_email));  
  
CREATE TABLE Funcionario_ServicosRealizados (  
  Funcionario_Utilizador_email  varchar(50) NOT NULL UNIQUE,
```



```

ServicosRealizados_id_ServicosRealizados int NOT NULL UNIQUE,
PRIMARY KEY (Funcionario_Utilizador_email,
ServicosRealizados_id_ServicosRealizados));
CREATE TABLE Servico (
id_servico    int IDENTITY NOT NULL,
nome          varchar(15) NOT NULL,
descricao     varchar(100) NOT NULL,
preco         float(4) NOT NULL,
precoSProduto float(4) NOT NULL,
precoSProdutos float(4) NULL,
PRIMARY KEY (id_servico));
CREATE TABLE ServicosRealizados (
id_ServicosRealizados int IDENTITY NOT NULL,
Cliente_Utilizador_email varchar(50) NOT NULL,
morada               varchar(100) NOT NULL,
data                 datetime NOT NULL,
preco                float(4) NOT NULL,
dimensao             varchar(1) NOT NULL,
prods                varchar(1) NOT NULL,
aspectosimp          varchar(150) NULL,
pendente             varchar(1) NOT NULL,
duracao              int NOT NULL,
PRIMARY KEY (id_ServicosRealizados));
CREATE TABLE Utilizador (
email  varchar(50) NOT NULL,
username varchar(20) NOT NULL UNIQUE,
password varchar(20) NOT NULL,
estatuto varchar(1) NOT NULL,
PRIMARY KEY (email));
ALTER TABLE Funcionario ADD CONSTRAINT FKFuncionari612998 FOREIGN KEY
(Servico_id_servico) REFERENCES Servico (id_servico);
ALTER TABLE Formulario_de_Realizacao ADD CONSTRAINT FKFormulario179732
FOREIGN KEY (FuncionarioUtilizador_email) REFERENCES Funcionario (Utilizador_email);
ALTER TABLE Formulario_de_Satisfacao ADD CONSTRAINT FKFormulario936538 FOREIGN
KEY (Funcionario_Utilizador_email) REFERENCES Funcionario (Utilizador_email);
ALTER TABLE Funcionario ADD CONSTRAINT FKFuncionari216562 FOREIGN KEY
(Utilizador_email) REFERENCES Utilizador (email);
ALTER TABLE Cliente ADD CONSTRAINT FKCliente209000 FOREIGN KEY (Utilizador_email)
REFERENCES Utilizador (email);

```

```

ALTER TABLE Formulario_de_Realizacao ADD CONSTRAINT FKFormulario590656
FOREIGN KEY (Cliente_Utilizador_email) REFERENCES Cliente (Utilizador_email);
ALTER TABLE Formulario_de_Satisfacao ADD CONSTRAINT FKFormulario286464 FOREIGN
KEY (Cliente_Utilizador_email) REFERENCES Cliente (Utilizador_email);
ALTER TABLE Funcionario_ServicosRealizados ADD CONSTRAINT FKFuncionari157923
FOREIGN KEY (Funcionario_Utilizador_email) REFERENCES Funcionario (Utilizador_email);
ALTER TABLE Funcionario_ServicosRealizados ADD CONSTRAINT FKFuncionari343238
FOREIGN KEY (ServicosRealizados_id_ServicosRealizados) REFERENCES
ServicosRealizados (id_ServicosRealizados);
ALTER TABLE ServicosRealizados ADD CONSTRAINT FKServicosRe591143 FOREIGN KEY
(Cliente_Utilizador_email) REFERENCES Cliente (Utilizador_email);

ALTER TABLE Funcionario DROP CONSTRAINT FKFuncionari612998;
ALTER TABLE Formulario_de_Realizacao DROP CONSTRAINT FKFormulario179732;
ALTER TABLE Formulario_de_Satisfacao DROP CONSTRAINT FKFormulario936538;
ALTER TABLE Funcionario DROP CONSTRAINT FKFuncionari216562;
ALTER TABLE Cliente DROP CONSTRAINT FKCliente209000;
ALTER TABLE Formulario_de_Realizacao DROP CONSTRAINT FKFormulario590656;
ALTER TABLE Formulario_de_Satisfacao DROP CONSTRAINT FKFormulario286464;
ALTER TABLE Funcionario_ServicosRealizados DROP CONSTRAINT FKFuncionari157923;
ALTER TABLE Funcionario_ServicosRealizados DROP CONSTRAINT FKFuncionari343238;
ALTER TABLE ServicosRealizados DROP CONSTRAINT FKServicosRe591143;
DROP TABLE Cliente;
DROP TABLE Formulario_de_Realizacao;
DROP TABLE Formulario_de_Satisfacao;
DROP TABLE Funcionario;
DROP TABLE Funcionario_ServicosRealizados;
DROP TABLE Servico;
DROP TABLE ServicosRealizados;
DROP TABLE Utilizador;

```

VI. Arquitetura do sistema

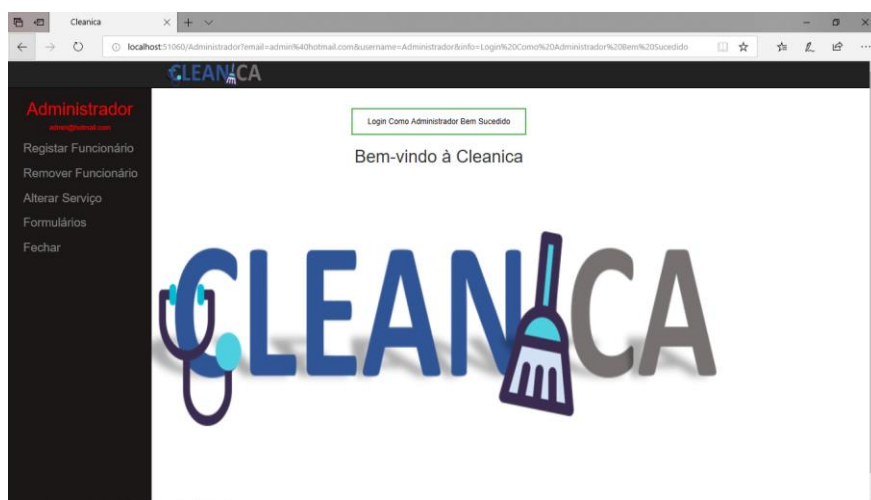
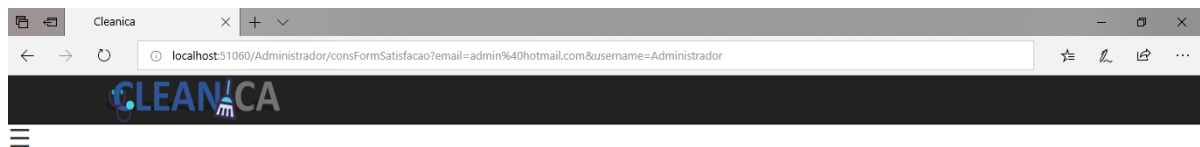


Figura 92 - Implementação do menu inicial (Administrador)

A imagem mostra a mesma interface web, mas com o formulário "Registrar um novo funcionário" aberto. O formulário contém campos para: "Email", "Username", "Password", "Serviço" (com uma lista suspensa selecionando "Jardinagem"), "Cartão Cidadão", "Telemóvel", "IBAN", "Data Nascimento" (com máscara "dd-mm-aaaa hh:mm:ss") e "Morada". O menu lateral permanece o mesmo, com a opção "Registrar Funcionário" destacada.

Figura 93 - Implementação do menu registrar novo funcionário (Administrador)



Formulários de Satisfação

Cliente	Funcionário	Pontuação	Sugestões
carlospedrosa97@hotmail.com	aaraujo@hotmail.com	5	
carlospedrosa97@hotmail.com	htaveira@hotmail.com	5	
mgcsousa@outlook.pt	ibareira@hotmail.com	4	
mgcsousa@outlook.pt	jsantos@hotmail.com	4	
davidjoseiteixeira10@hotmail.com	cteixeira@hotmail.com	3	
rodrigosporguista@hotmail.com	cr7@hotmail.com	4	
danyvieira_96@hotmail.com	fpedrosa@hotmail.com	5	
danyvieira_96@hotmail.com	jbarroso@hotmail.com	5	

Figura 94 - Implementação do menu Formulários de Satisfação (Administrador)

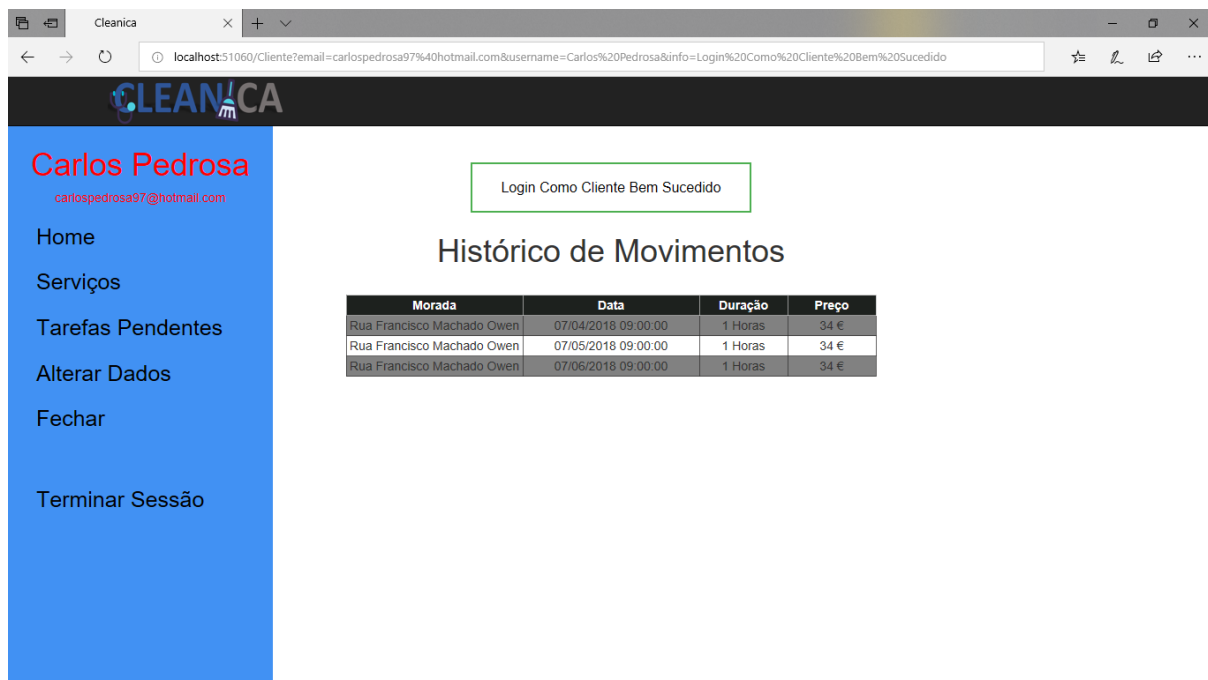


Figura 95 - Implementação do menu consultar histórico (Cliente)

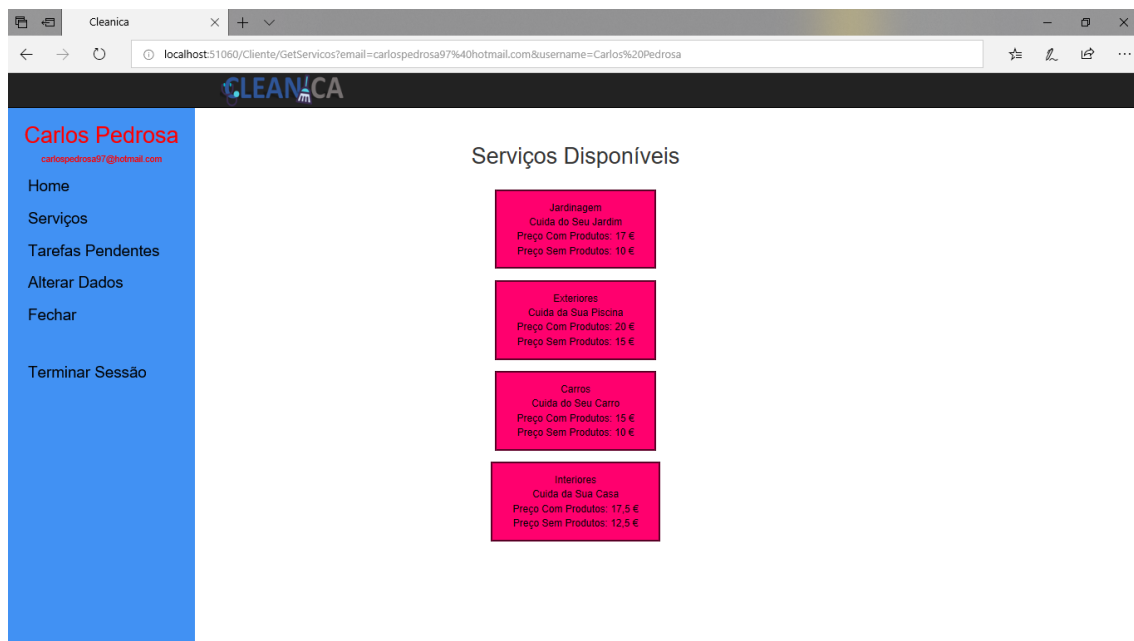


Figura 96 - Implementação do menu serviços disponíveis (Cliente)

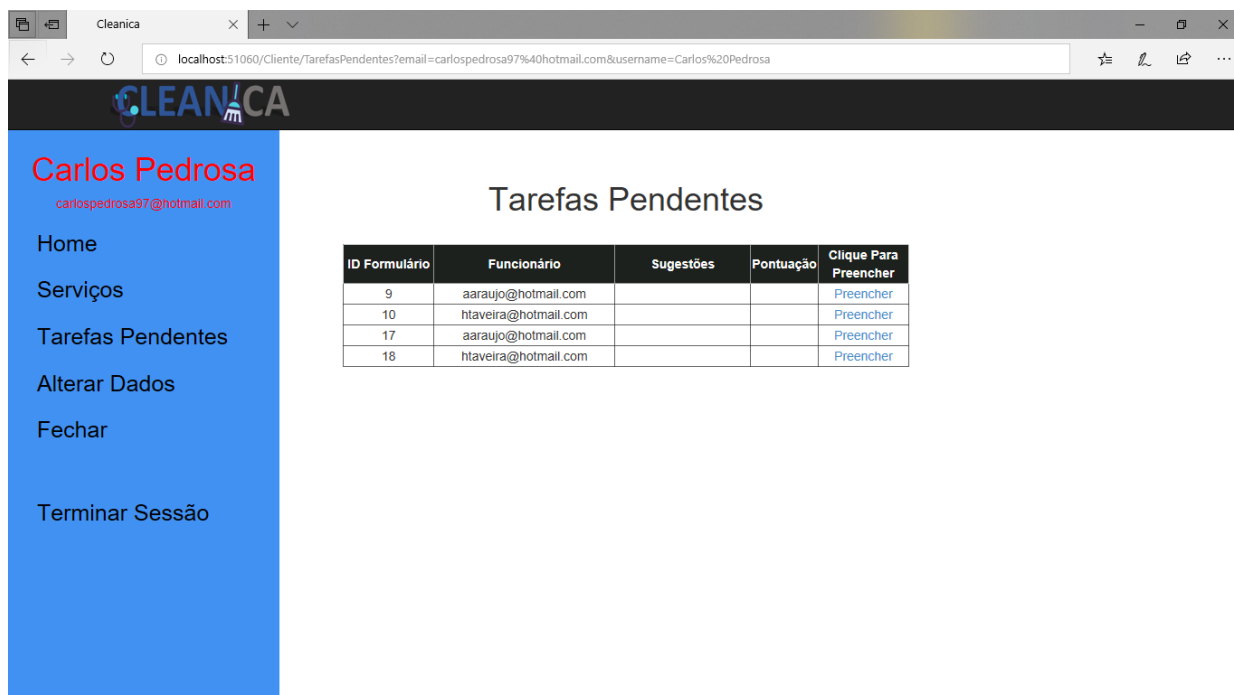


Figura 97 - Implementação do menu tarefas pendentes (Cliente)

Cleanica

Cristiano Ronaldo
cr7@hotmail.com

- Home
- Histórico de Serviços
- Alterar Dados
- Fechar
- Terminar Sessão

Preencher Formulário de Realização

Cliente
rodrigospotinguista@hotmail.com

Data
06/10/2018

Duração

Observações

Submeter

Figura 98 - Implementação do menu preencher formulário de realização (funcionário)

Cleanica

Cristiano Ronaldo
cr7@hotmail.com

- Home
- Histórico de Serviços
- Alterar Dados
- Fechar
- Terminar Sessão

Alterar Dados

Username
Cristiano

Password

Morada
Green

Telemóvel
918908793

IBAN
PT50111111111111111111111111111111

Alterar

Figura 99 - Implementação do menu alterar dados (funcionário)



Histórico de Serviços

Cliente	Morada	Data	Preço	Dimensão	Produtos
rodrigospotinguista@hotmail.com	Primeira Avenida	10/04/2018 09:00:00	15 €	P	S
rodrigospotinguista@hotmail.com	Primeira Avenida	10/05/2018 09:00:00	15 €	P	S

Figura 100 - Implementação do menu Histórico de Serviços (funcionário)