



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

Abril, 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

Abril, 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. O presente trabalho contém ainda uma maquete onde se mostra um esquema que a aplicação final seguirá.

Por fim, apresentamos toda a modelação necessária, bem como a 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. Levantamento de Requisitos	8
1.1. Método de levantamento e análise de requisitos adotado	8
1.2. Requisitos levantados e respetiva análise	8
2. Modelação do Software	14
2.1. Modelo de domínio	14
2.2. Use cases	15
2.3. Especificação dos Use Cases	19
Administrador	19
2.3.1. Registrar novo funcionário	19
2.3.2. Remover funcionário	20
2.3.3. Alterar dados de serviço	21
2.3.4. Consultar formulários de satisfação	21
Cliente	22
2.3.5. Efetuar registo	22
2.3.6. Consultar serviços	22
2.3.7. Preencher formulários de satisfação	23
2.3.8. Consultar histórico	23
2.3.9. Requisitar serviço	23
2.3.9.1. Requisitar serviço de lavagem de automóvel	24
2.3.9.2. Requisitar serviço de jardinagem	25
2.3.9.3. Requisitar serviço de limpeza de piscinas	26
2.3.9.4. Requisitar serviço de limpeza de interiores	27
2.3.9.5. Includes: Efetuar Pagamento, escolher funcionários para serviço e gerar fatura	28
2.3.10. Alterar dados	29
Funcionário	30
2.3.11. Consultar calendário de serviços	30
2.3.12. Consultar histórico de encargos	30
2.3.13. Confirmar realização de serviço	31
2.3.14. Pedir direções	32
2.3.15. Alterar dados funcionário	32

2.4. Diagramas de sequência de sistemas	33
Administrador	33
2.4.1. Registrar novo funcionário	33
2.4.2. Alterar dados de serviço	34
Cliente	35
2.4.3. Preencher formulário de satisfação	35
2.4.4. Requisitar serviços	37
Funcionário	38
2.4.5. Confirmar realização de serviço	38
2.4.6. Consultar calendário de serviços	39
2.5. Diagrama de classes	40
2.6. Diagrama de atividades	41
2.6.1. Diagrama de atividade Administrador	42
2.6.2. Diagrama de atividade Cliente	43
2.6.3. Diagrama de atividade Funcionário	44
3. Mockup da interface	45
3.1. Administrador	46
3.2. Cliente	48
3.3. Funcionário	51
4. Base de dados	52
4.1. Identificação e caracterização das entidades	52
4.2. Identificação e caracterização dos relacionamentos	53
4.3. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos	53
4.4. Modelo lógico	57
4.5. Modelo físico	59
 Anexos	
I. Diagramas de Sequência	63
II. Modelo Conceitual	75
III. Modelo físico no SQL Server	76

Índice de Figuras

Figura 1 - Diagrama de Use Cases gerais	15
Figura 2 - Subdiagrama administrador	16
Figura 3 - Subdiagrama do cliente	17
Figura 4 - Subdiagrama do cliente, serviços	17
Figura 5 - Subdiagrama funcionário	18
Figura 6 - Especificação do Use Case Registrar funcionário	19
Figura 7 - Especificação do Use Case Remover funcionário	20
Figura 8 - Especificação do Use Case Alterar dados de serviço	21
Figura 9 - Especificação do Use Case Consultar formulários de satisfação	21
Figura 10 - Especificação do Use Case Efetuar registo	22
Figura 11 - Especificação do Use Case Consultar serviços	22
Figura 12 - Especificação do Use Case Preencher formulários de satisfação	23
Figura 13 - Especificação do Use Case Consultar histórico	23
Figura 14 - Especificação do Use Case Requisitar serviço de lavagem de automóvel	24
Figura 15 - Especificação do Use Case Requisitar serviço de jardinagem	25
Figura 16 - Especificação do Use Case Requisitar serviço de jardinagem	26
Figura 17 - Especificação do Use Case Requisitar serviço de limpeza de interiores	27
Figura 18 - Especificação do Use Case Efetuar Pagamento	28
Figura 19 - Especificação do Use Case Escolher funcionário(s) para o serviço	28
Figura 20 - Especificação do Use Case Gerar fatura	29
Figura 21 - Especificação do Use Case Alterar dados	29
Figura 22 - Especificação do Use Case Consultar calendário de serviços	30
Figura 23 - Especificação do Use Case Consultar histórico de encargos	31
Figura 24 - Especificação do Use Case Confirmar realização do serviço	31
Figura 25 - Especificação do Use Case Pedir direções	32
Figura 26 - Especificação do Use Case Alterar dados funcionário	32
Figura 27 - Diagrama de sequência: registar novo funcionário	33
Figura 28 - Especificação do diagrama de sequência: registar funcionário	34
Figura 29 - Diagrama de sequência: alterar dados de serviço	34

Figura 30 - Especificação do diagrama: alterar dados de serviço	35
Figura 31 - Diagrama de sequência: preencher formulário de satisfação	35
Figura 32 - Especificação do diagrama de sequência: preencher formulário de satisfação	36
Figura 33 - Diagrama de sequência: requisitar serviço	37
Figura 34 - Especificação do diagrama de sequências: requisitar serviço	38
Figura 35 - Diagrama de sequência: confirmar realização de serviço	38
Figura 36 - Especificação do diagrama de sequência: confirmar serviço	39
Figura 37 - Diagrama de sequência: consultar calendário de serviços	39
Figura 38 - Especificação do diagrama de sequência: consultar calendário de serviços	39
Figura 39 - Diagrama de classes	40
Figura 40 - Diagrama de Atividade Administrador	42
Figura 41 - Diagrama de Atividade Cliente	43
Figura 42 - Diagrama de Atividade Funcionário	44
Figura 43 - Mockup Login	45
Figura 44 - Mockup Menu principal Administrador	46
Figura 45 - Mockup Alterar serviço	46
Figura 46 - Mockup Formulários de satisfação	47
Figura 47 - Mockup Novo Funcionário	47
Figura 48 - Mockup Remover funcionário	48
Figura 49 - Mockup Menu principal Cliente	48
Figura 50 - Mockup Consultar serviço	49
Figura 51 - Mockup Requisitar serviço lavagem automóvel	49
Figura 52 - Mockup Requisitar serviço, interiores, exteriores e piscina	50
Figura 53 - Mockup Alterar dados (cliente)	50
Figura 54 - Mockup Menu principal funcionário	51
Figura 55 - Modelo lógico	57
Figura 56 - Diagrama de sequência (Administrador): consultar formulário de satisfação	63
Figura 57 - Especificação do diagrama de sequência (Administrador): consultar formulário de satisfação	63
Figura 58 - Diagrama de sequência (Administrador): remover funcionário	64
Figura 59 - Especificação do diagrama de sequência (Administrador): remover funcionário	64
Figura 60 - Diagrama de sequência (cliente): alterar dados	65
Figura 61 - Especificação do diagrama de sequência (cliente): alterar dados	65
Figura 62 - Diagrama de sequência (cliente): consultar histórico	65
Figura 63 - Especificação do diagrama de sequência (cliente): consultar histórico	66

Figura 64 - Diagrama de sequência (cliente): consultar serviço	66
Figura 65 - Especificação do diagrama de sequência (cliente): consultar serviço	66
Figura 66 - Diagrama de sequência: efetuar Login	67
Figura 67 - Especificação do diagrama de sequência: efetuar Login	67
Figura 68 - Diagrama de sequência: efetuar pagamento	68
Figura 69 - Especificação do diagrama de sequência: efetuar pagamento	69
Figura 70 - Diagrama de sequência: escolher funcionário(s)	69
Figura 71 - Especificação do diagrama de sequência: escolher funcionário(s)	70
Figura 72 - Diagrama de sequência (funcionário): alterar dados	70
Figura 73 - Especificação do diagrama de sequência (funcionário): consultar serviço	71
Figura 74 - Diagrama de sequência (funcionário): consultar histórico de encargos	71
Figura 75 - Especificação do diagrama de sequência (funcionário): consultar histórico de encargos	72
Figura 76 - Diagrama de sequência (funcionário): pedir direções	72
Figura 77 - Especificação do diagrama de sequência (funcionário): pedir direções	73
Figura 78 - Diagrama de sequência (funcionário): gerar fatura	73
Figura 79 - Especificação do diagrama de sequência (funcionário): gerar fatura	74
Figura 80 - Modelo Concetual	75

Índice de Tabelas

Tabela 1 - Identificação das entidades	52
Tabela 2 - Caracterização dos relacionamentos	53
Tabela 3 - Identificação e associação dos atributos com as entidades	56

1. Levantamento de Requisitos

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.

1.2. Requisitos levantados e respetiva análise

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*

- 4.1. O sistema deverá apresentar um menu com a opção: “Alterar serviços”;
- 4.2. O sistema deverá permitir alterar o preço e a descrição do serviço;
- 4.3. O sistema deverá confirmar a alteração do serviço e atualizar a base de dados.

- **RF4:** *o administrador poderá consultar os formulários de satisfação*

- 5.1. O sistema deverá apresentar uma área relativa à consulta de formulários de satisfação;
- 5.2. O sistema apresentará todos os formulários relativos ao funcionário indicado ordenados por data ou nome.

➤ **Cliente:**

- **RF1:** *O cliente poderá registrar-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 deste 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 e preço.

- **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 cliente 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, no início de cada dia de trabalho enviara uma notificação ao funcionário.

- **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 permitira 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.

1.2.2. 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 poder utilizar o sistema a ser desenvolvido deverá possuir este serviço.

- **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 encriptar todos 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. Modelação do Software

Por forma a implementar de modo conciso e organizado todo o software 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.

2.1. 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 desenvolver. Nesse sentido, consideramos que este se trata de um modelo fulcral pois serve como um ponto de partida para toda a modelação.

2.2. Use cases

Os Use Case 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 Case 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 consequentemente um melhor entendimento, subdividimos os diagramas.

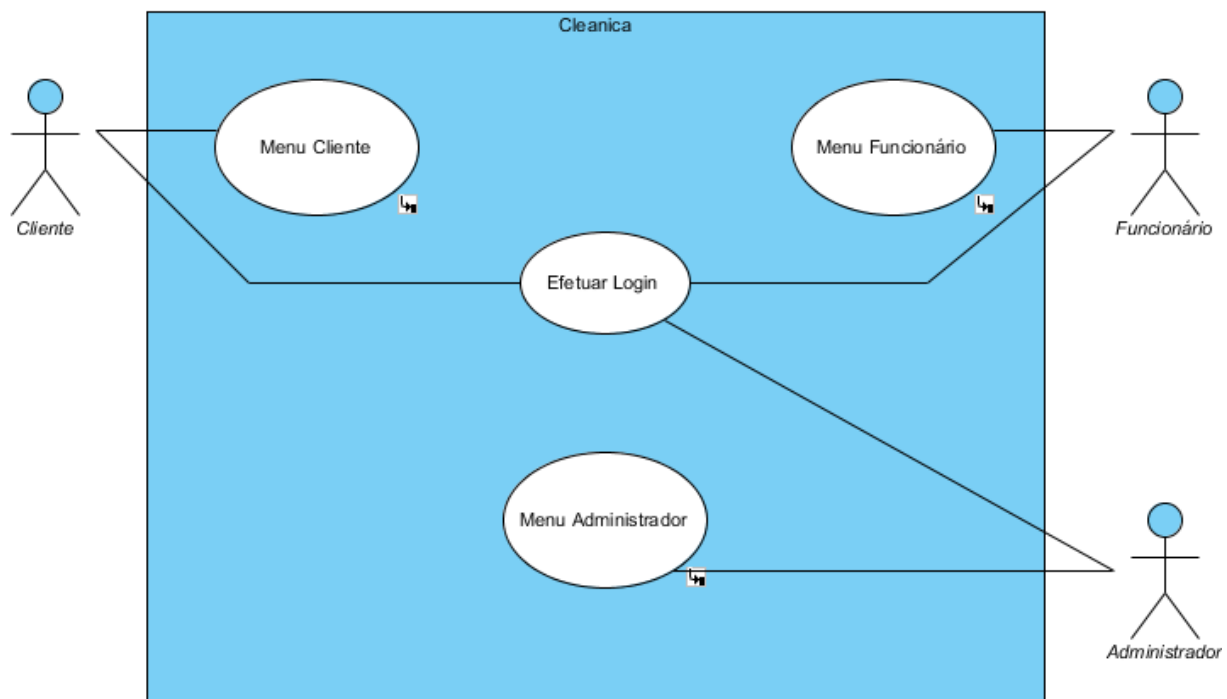


Figura 1 - Diagrama de Use Cases gerais

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 referencia às funcionalidades que cada um dos utilizadores possuirá.

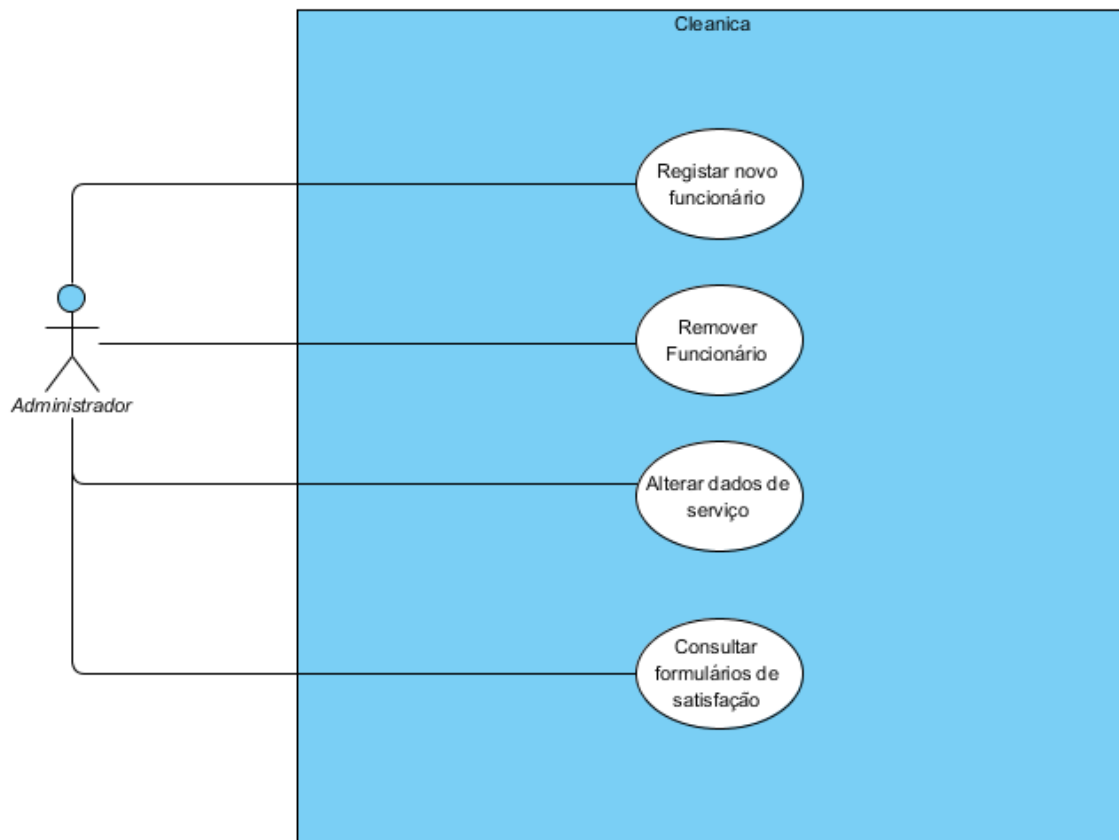


Figura 2 - Subdiagrama administrador

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.

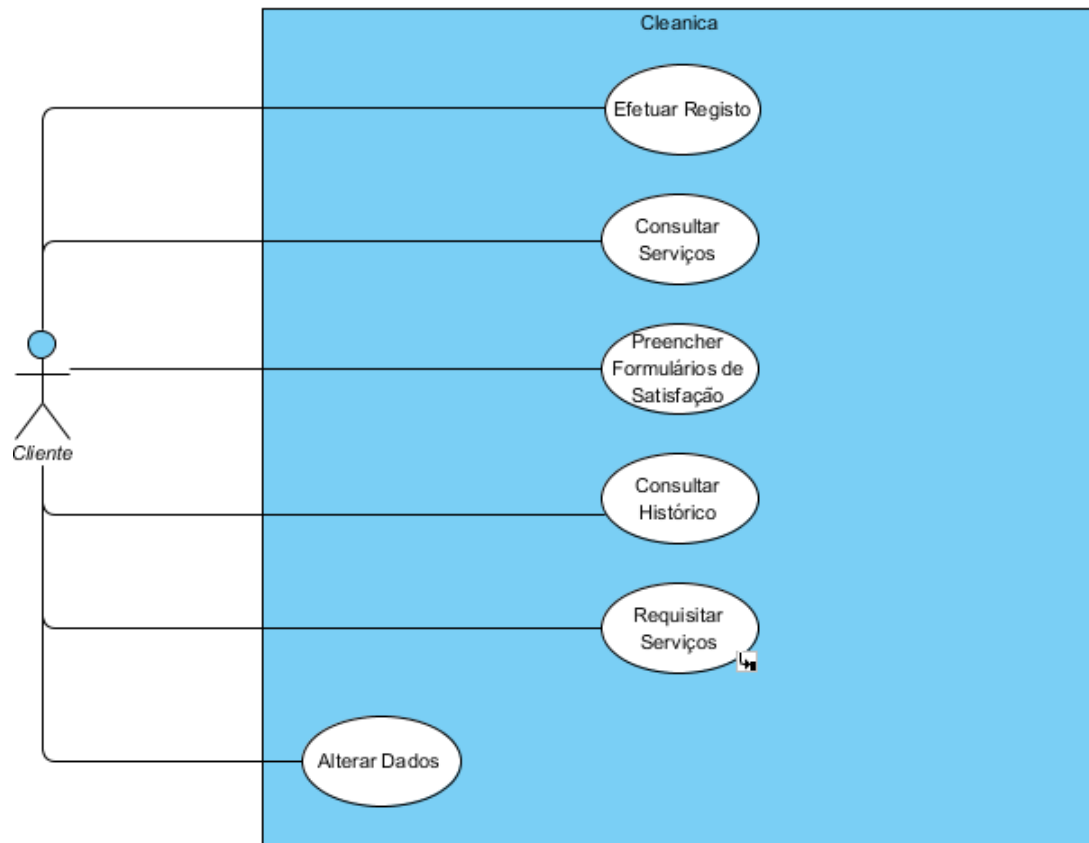


Figura 3 - Subdiagrama do cliente

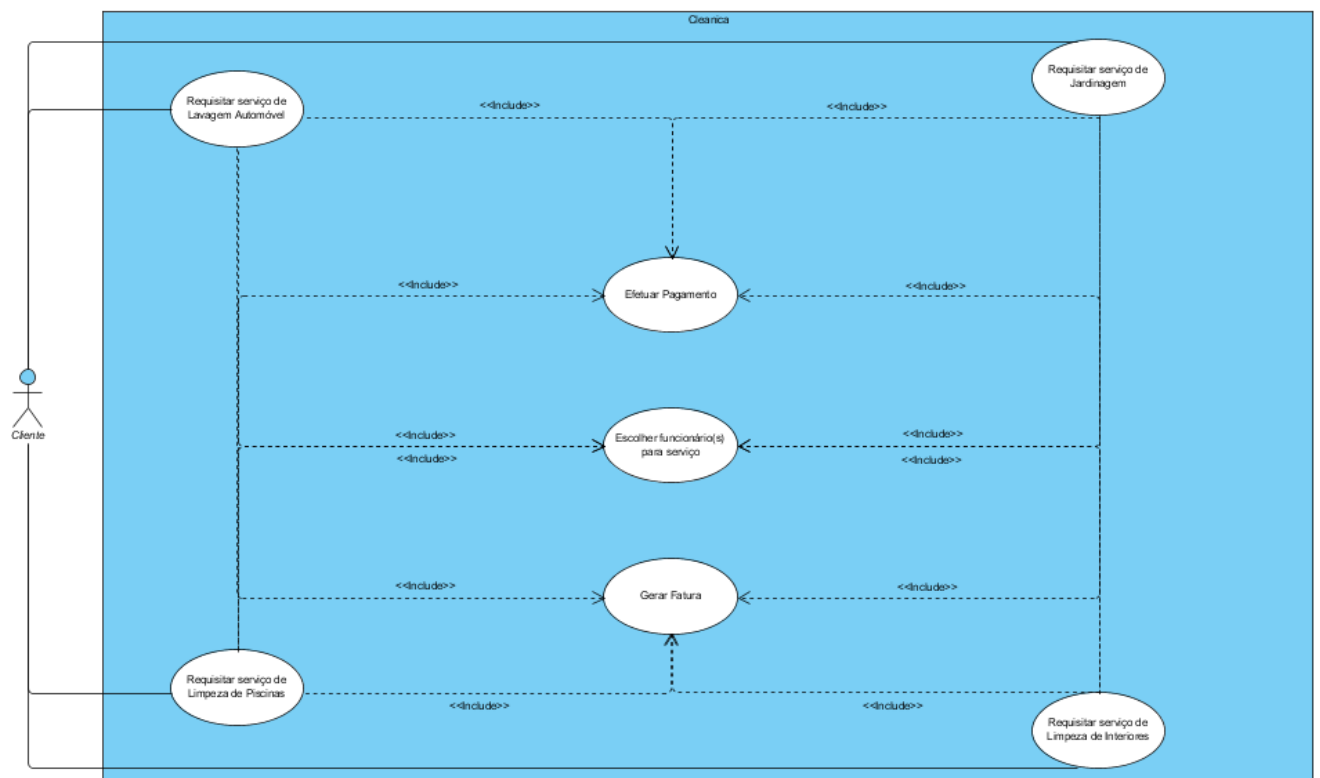


Figura 4 - Subdiagrama do cliente, serviços

Neste subdiagrama são notórias quais as funcionalidades que a aplicação ira fornecer a cada cliente. Uma vez que “Requisitar serviço” é uma função complexa, decidimos criar Use Cases que a detalham melhor. Para além disso, cada cliente possuirá uma panóplia de ferramentas ao seu dispor que permitiram gerir a sua conta da melhor forma possível.

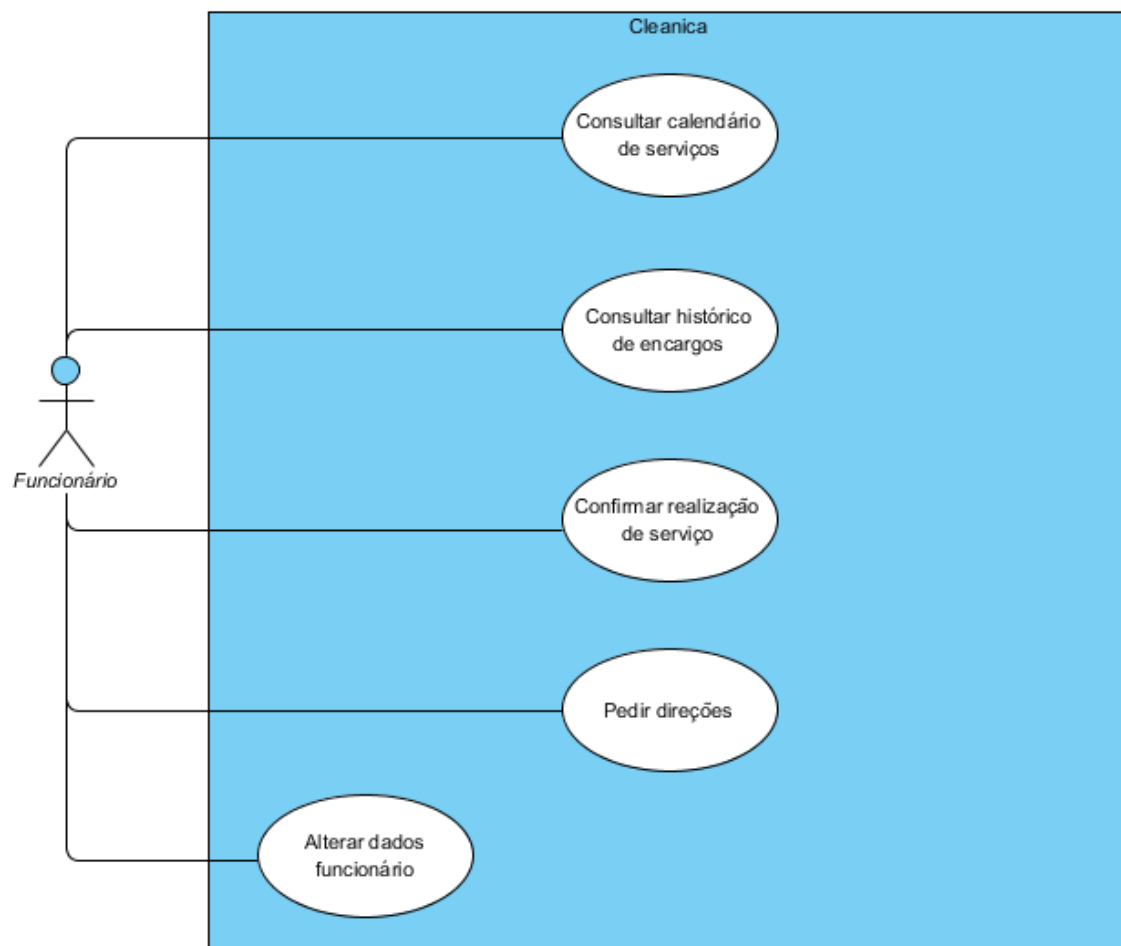


Figura 5 - Subdiagrama funcionário

À semelhança do que acontece com o cliente o funcionário também possui uma extensão da aplicação com a suas próprias funcionalidades. Assim, esta apresenta toda a informação de forma bastante útil para que o funcionário possa gerir todo serviço que efetua de forma simples e intuitiva.

2.3. Especificação dos Use Cases

➤ Administrador

2.3.1. Registrar 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 6 - Especificação do Use Case Registrar 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 devera ser anotada, já que, cada funcionário possui uma vocação para uma área. O sistema trata de validar todos os dados, indicando eventuais erros.

2.3.2. 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 7 - 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.

2.3.3. 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 8 - Especificação do Use Case Alterar dados de serviço

Este use case surge para especificar o facto de o funcionário 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.

2.3.4. 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 9 - Especificação do Use Case Consultar formulários de satisfação

Por último surge o Use Case “Consultar formulários de satisfação” que permite ao administrador perceber o empenho envolvido 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

2.3.5. Efetuar registo

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

Figura 10 - 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.

2.3.6. Consultar serviços

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

Figura 11 - Especificação do Use Case Consultar serviços

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

2.3.7. Preencher formulários de satisfação

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

Figura 12 - Especificação do Use Case Preencher formulários de satisfação

O cliente pode preencher um formulário relativo ao serviço que fora executado em sua casa. 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.

2.3.8. Consultar histórico

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

Figura 13 - Especificação do Use Case Consultar histórico

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

2.3.9. 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.

2.3.9.1. 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
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 14 - 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, 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 funcionário para os serviço que acaba de ser requisitado.

2.3.9.2. 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
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 jardinagem

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

2.3.9.3. 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 16 - 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 de cliente poder optar por fornecer os produtos de limpeza o que vai ao encontro à política apresentada por Cleanica.

2.3.9.4. 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	
	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 17 - 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.

2.3.9.5. 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
Excepção 1 [dados inválidos] (passo 2)		Actor Input	System Response
	1		Sistema informa que os dados fornecidos são inválidos
Excepção 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 18 - 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 19 - 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 20 - 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 enviar gerar a fatura que será enviada como confirmação para o email.

2.3.10. Alterar dados

Preconditions	Estar autenticado no sistema		
Post-conditions			
Flow of Events		Actor Input	System Response
	1	Fornece nome e/ou morada e/ou n° 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 21 - 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

2.3.11. 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 22 - Especificação do Use Case Consultar calendário de serviços

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

2.3.12. 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 extrai 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 23 - Especificação do Use Case Consultar histórico de encargos

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

2.3.13. 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 24 - 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, consultando assim, considerações relativas a este.

2.3.14. Pedir direções

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

Figura 25 - 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.

2.3.15. Alterar dados funcionário

Preconditions	Estar autenticado no sistema	
Post-conditions		
Flow of Events		Actor Input
	1	Fornecer 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
	1	Indica que dados são inválidos

Figura 26 - 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.

2.4. Diagramas de sequência de sistemas

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 apresentados seriam uma 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

2.4.1. Registar novo funcionário

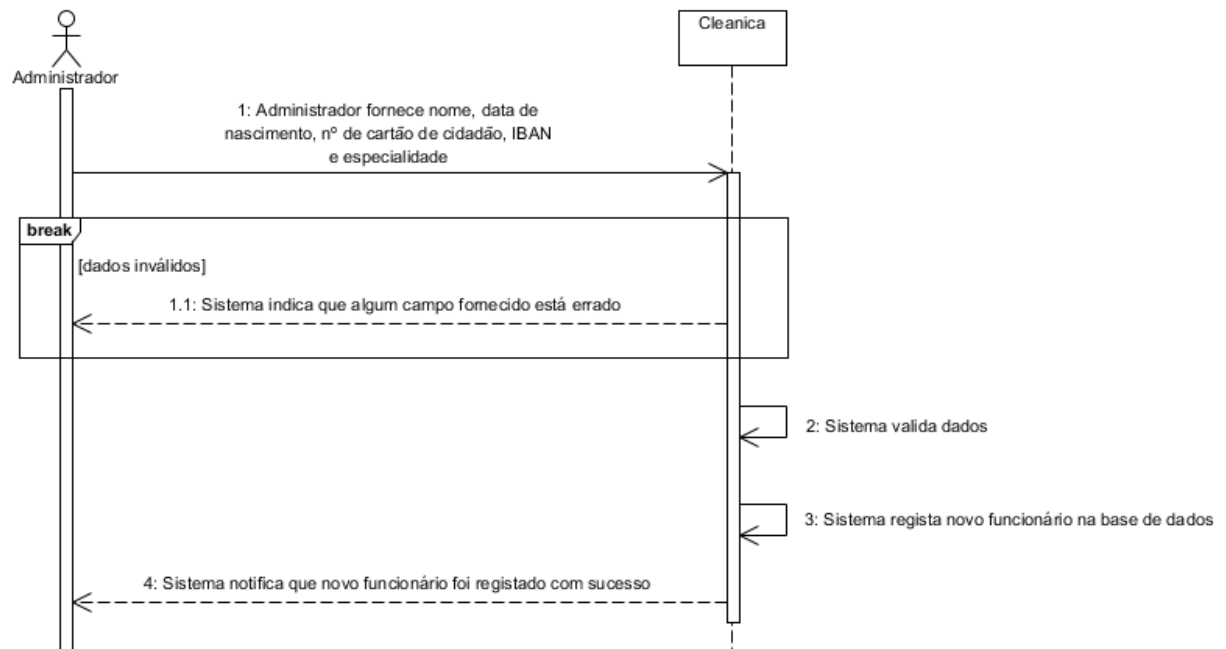


Figura 27 - Diagrama de sequência: registar novo funcionário

Seguidamente, apresentaremos a especificação deste DSS. Ao longo desta secção são apresentados os respetivos diagramas de sequência e a sua especificação.

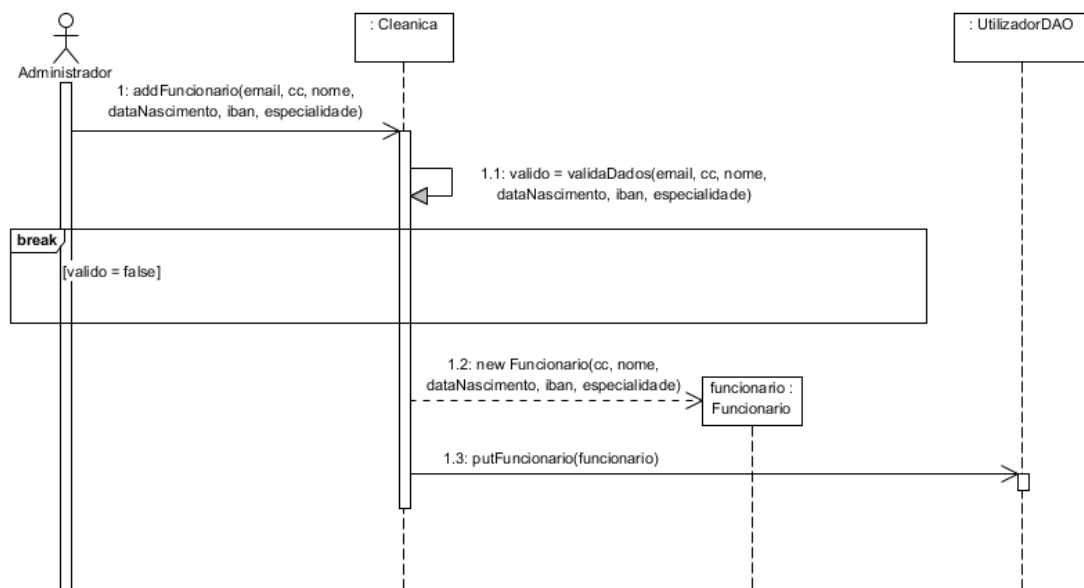


Figura 28 - Especificação do diagrama de sequência: registrar funcionario

2.4.2. Alterar dados de serviço

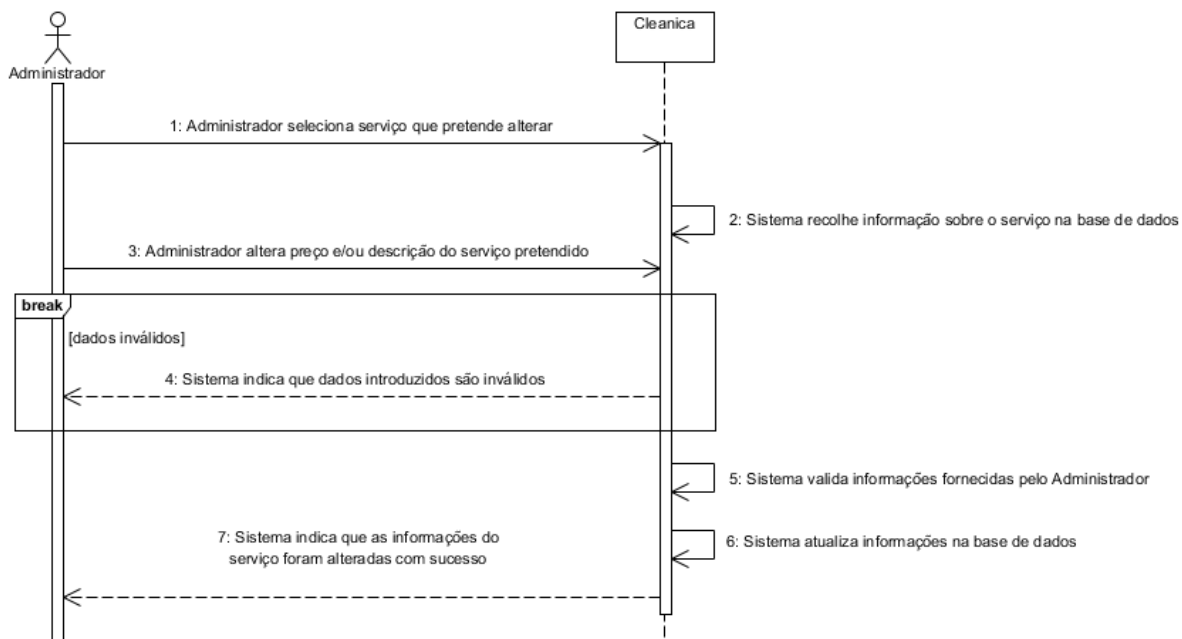


Figura 29 - Diagrama de sequência: alterar dados de serviço

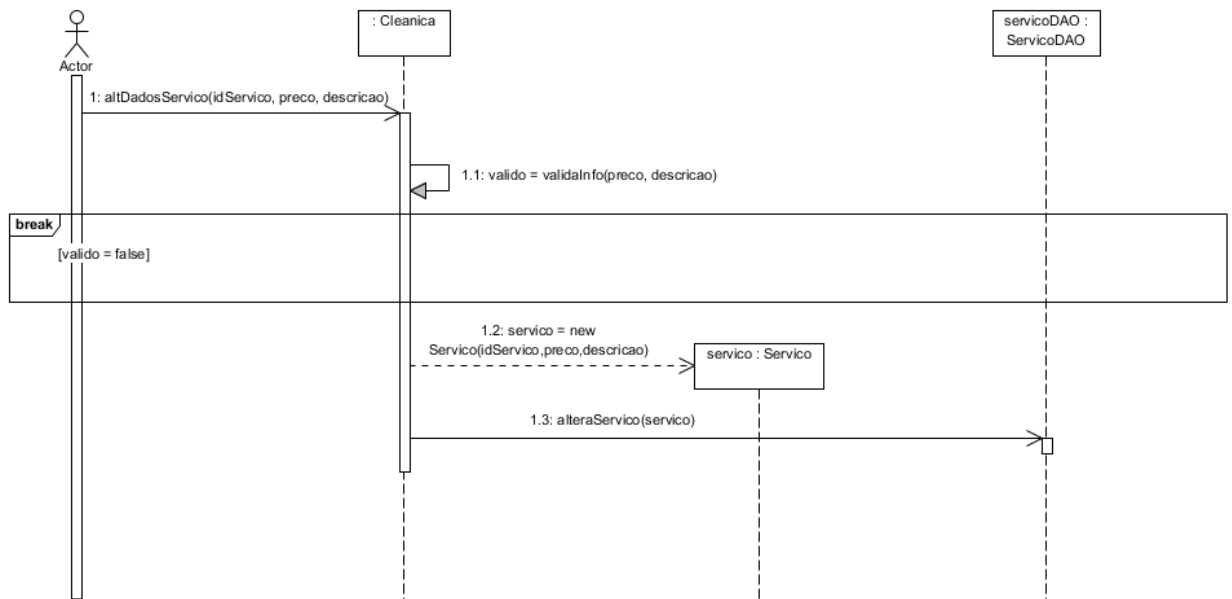


Figura 30 - Especificação do diagrama: alterar dados de serviço

➤ Cliente

2.4.3. Preencher formulário de satisfação

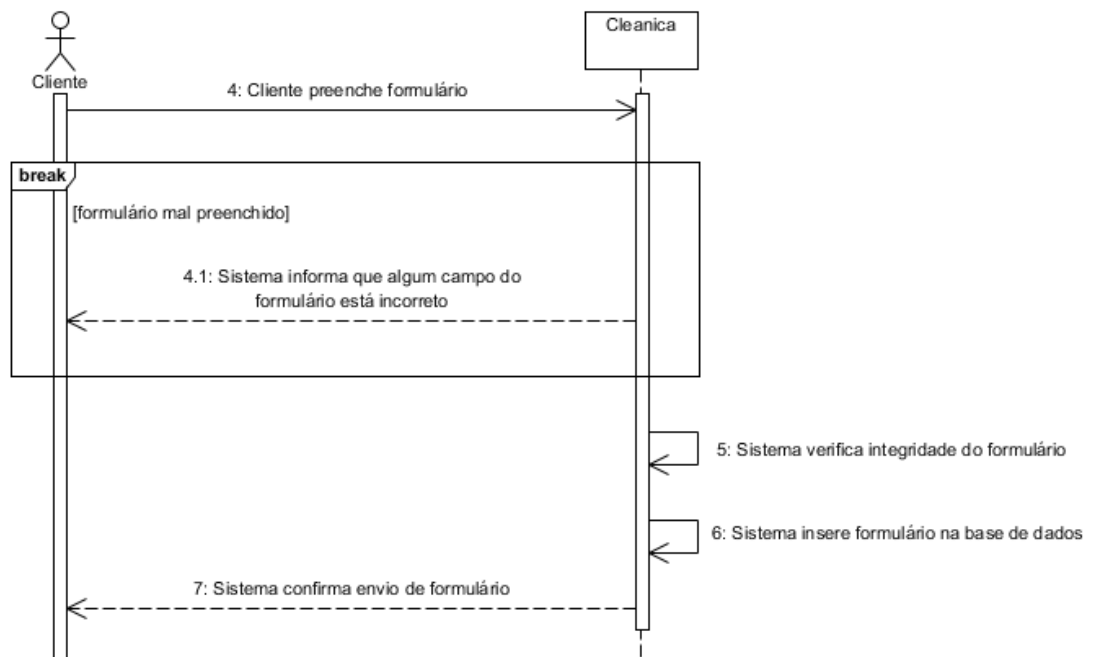


Figura 31 - Diagrama de sequência: preencher formulário de satisfação

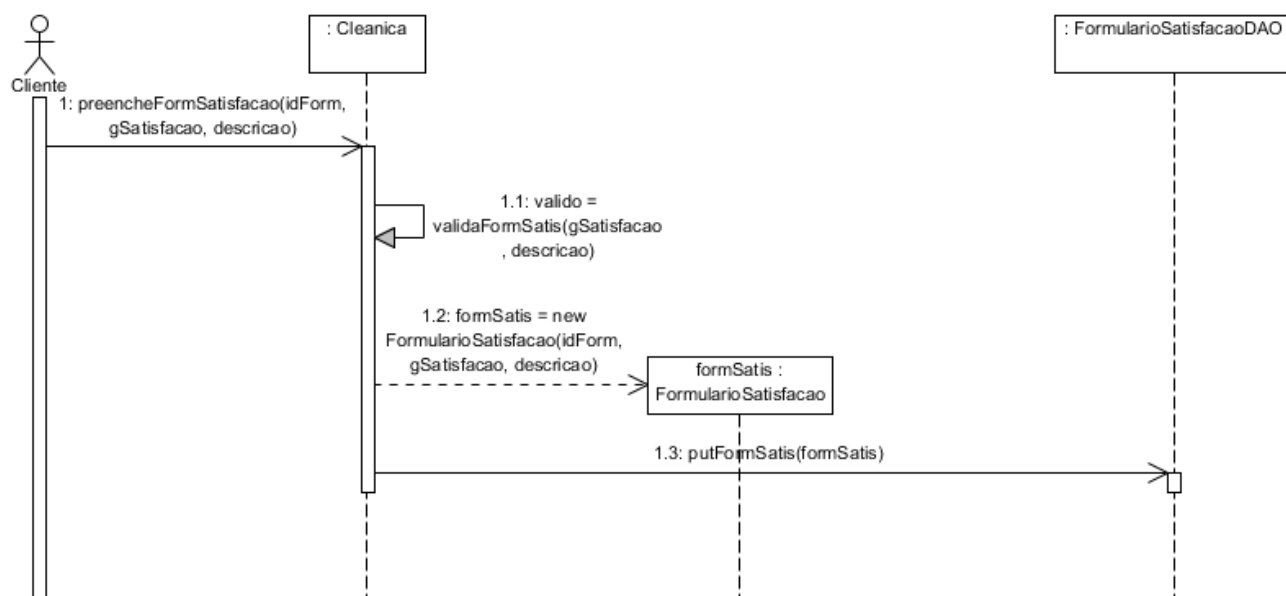


Figura 32 - Especificação do diagrama de sequência: preencher formulário de satisfação

Continua a ser importante referir que apesar de terem sido modelados todos os Use Case anteriormente apresentados, nesta secção apenas incluímos algumas dessas modelações. Sendo assim, “Preencher formulário de satisfação” é um dos muitos DSS desenvolvidos. Assim, a figura 32 apresenta um estado final da especificação do diagrama de sequência em que cada classe já se encontra substituída pelos respetivos DAO.

2.4.4. Requisitar serviços

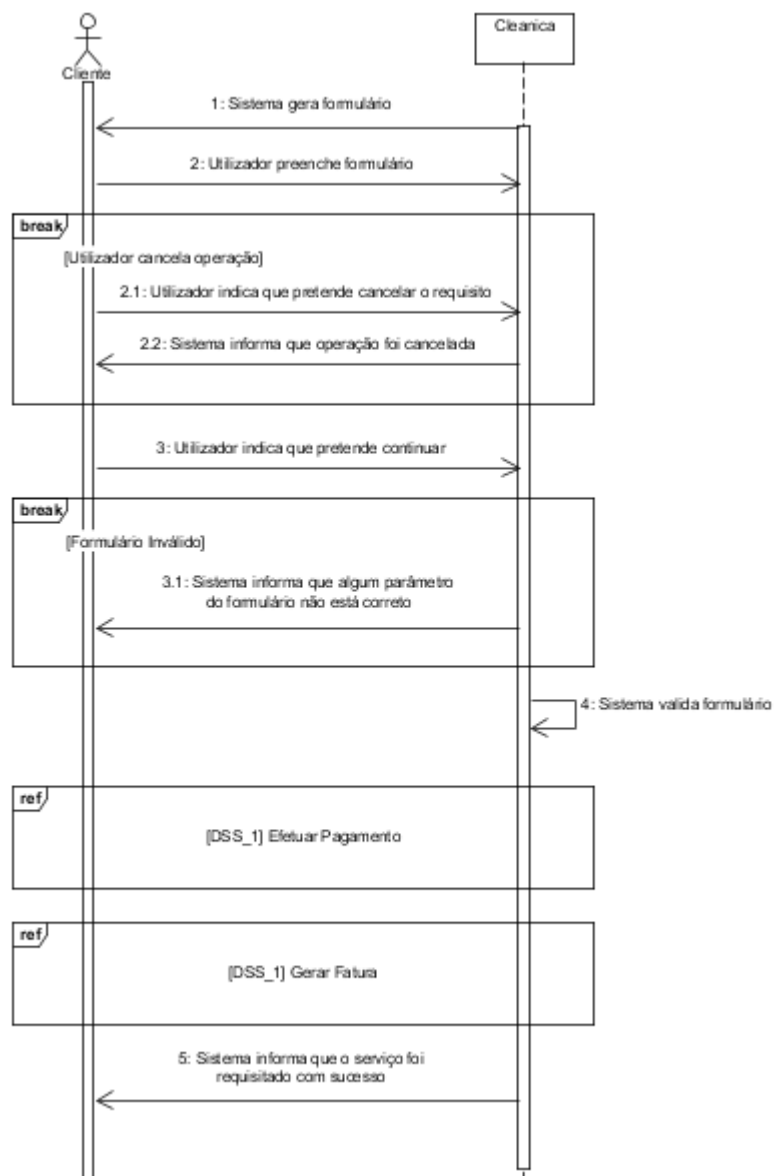


Figura 33 - Diagrama de sequência: requisitar serviço

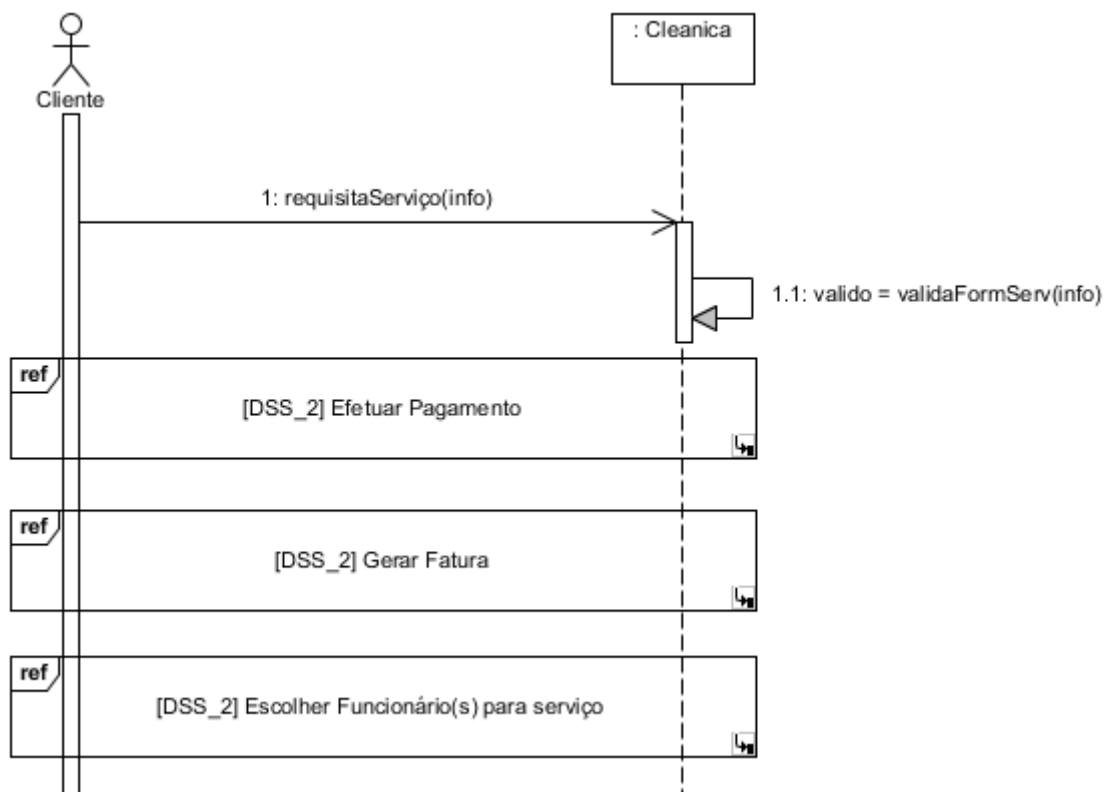


Figura 34 - Especificação do diagrama de sequências: requisitar serviço

➤ Funcionário

2.4.5. Confirmar realização de serviço

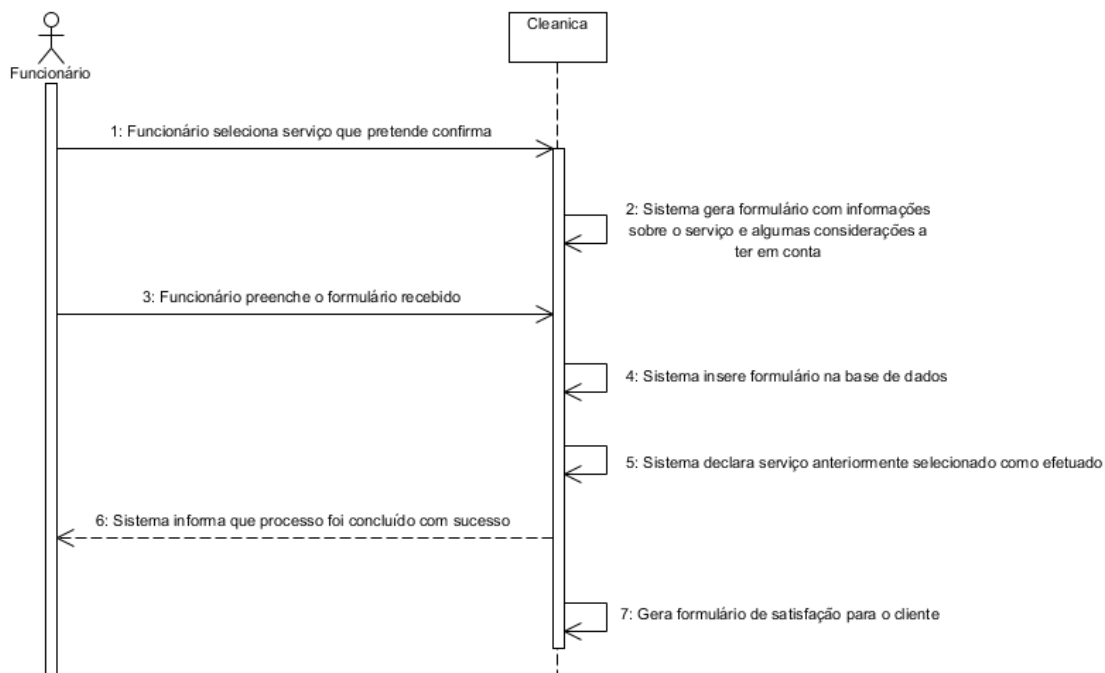


Figura 35 - Diagrama de sequência: confirmar realização de serviço

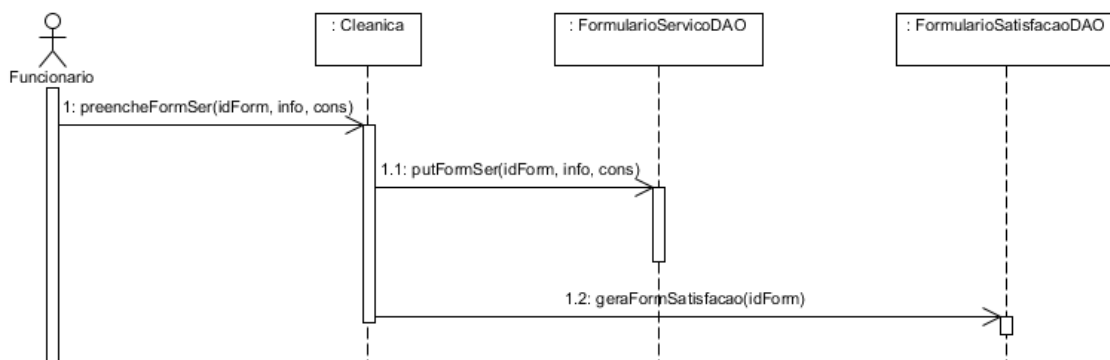


Figura 36 - Especificação do diagrama de sequência: confirmar serviço

2.4.6. Consultar calendário de serviços

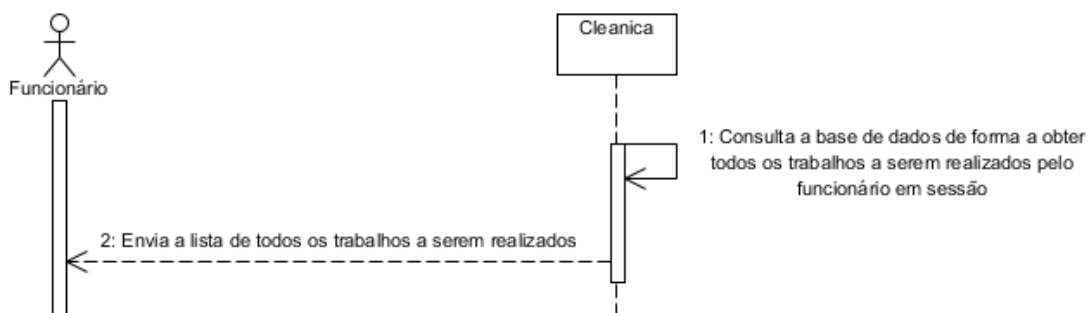


Figura 37 - Diagrama de sequência: consultar calendário de serviços

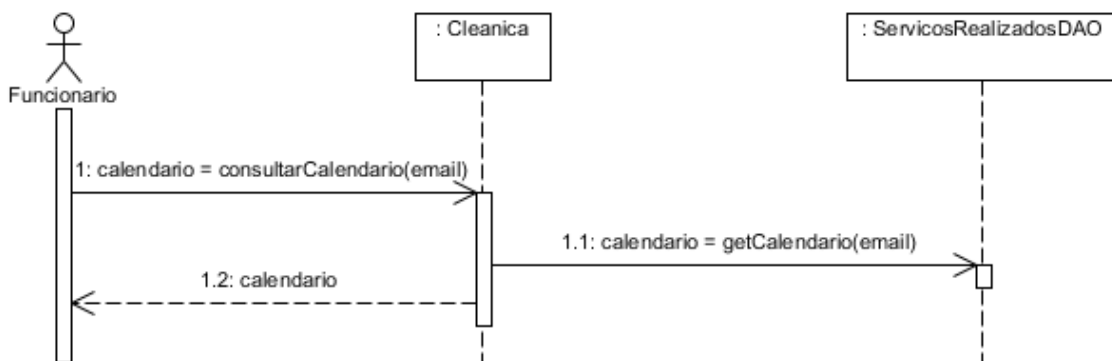


Figura 38 - Especificação do diagrama de sequência: consultar calendário de serviços

2.5. Diagrama de classes

Decidimos implementar diagrama de classes 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. Apenas realçar a existência da classe utilizador, usada para permitir fácil acesso aos dados de cada um deste.

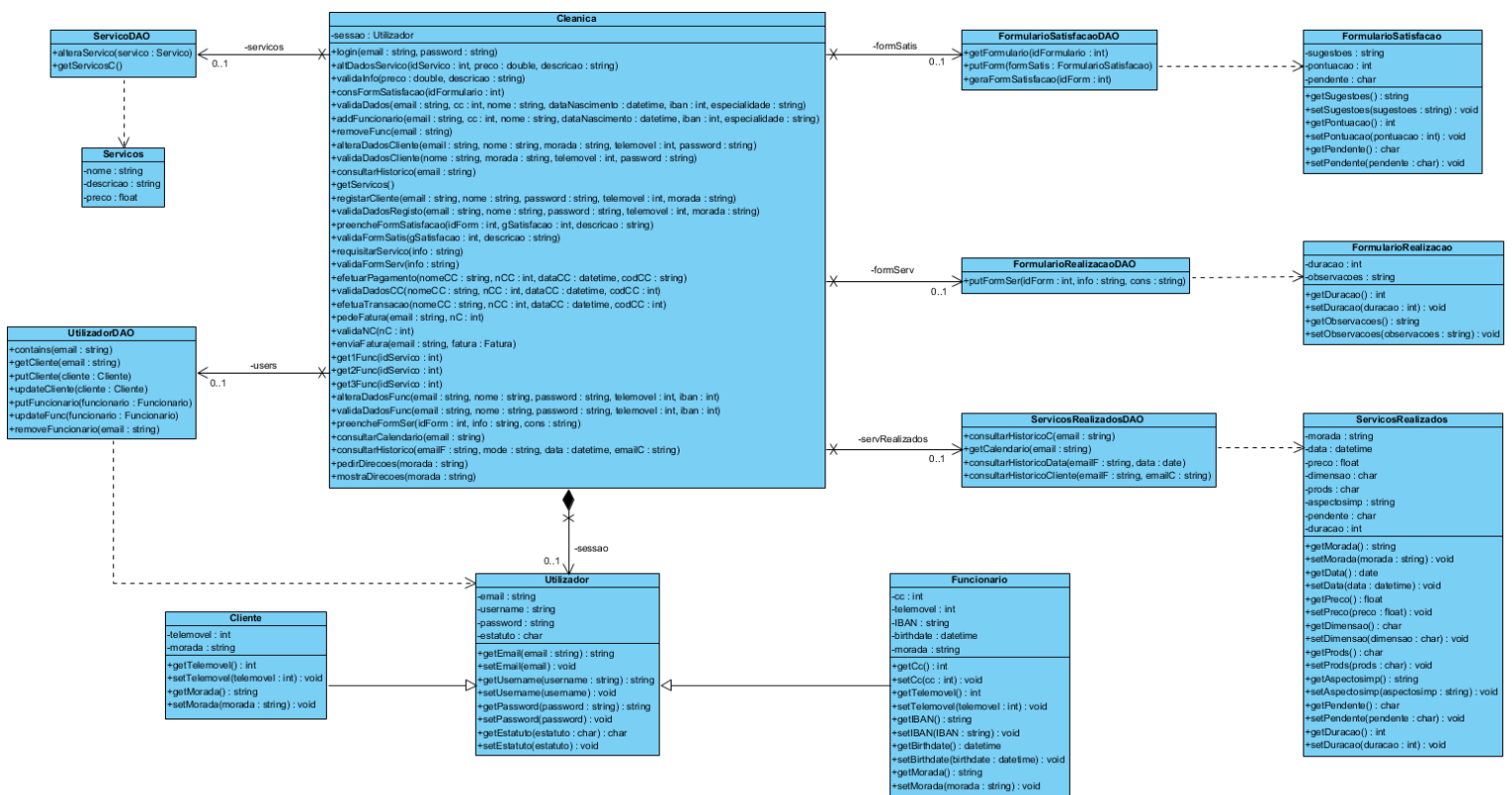


Figura 39 - Diagrama de classes

2.6. Diagrama de atividades

Sendo que se trata de uma aplicação consideramos que seria útil especificar o comportamento dos diferentes utilizadores. Assim, baseamo-nos nos diagramas de atividades, uma vez que, estes baseiam-se em modelos de fluxo de dados. Os diagramas que a seguir se apresentam, dizem respeito ao que resulta da interação dos utilizadores com 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.

2.6.1. Diagrama de atividade Administrador

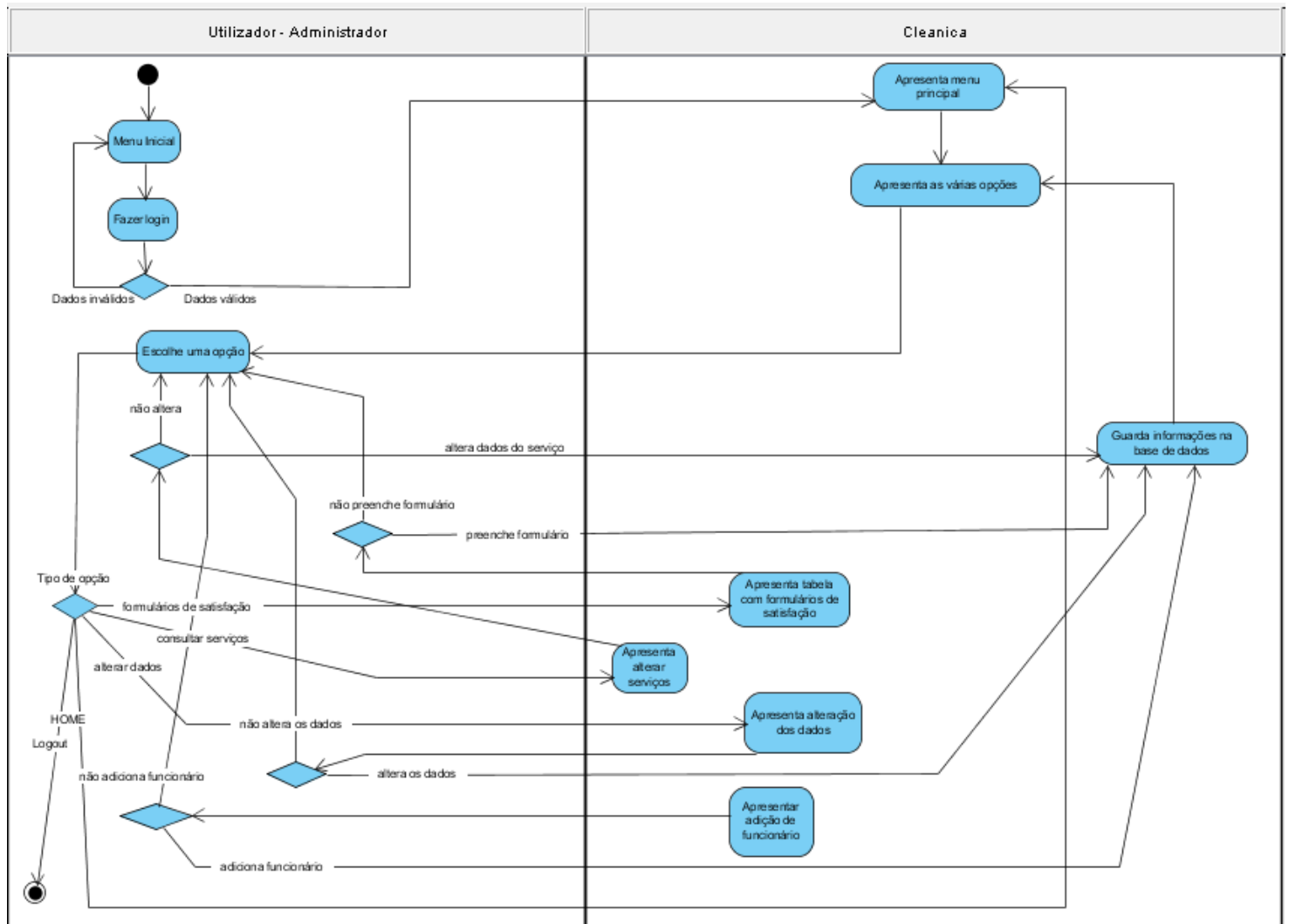


Figura 40 - Diagrama de Atividade Administrador

A figura 27 representa o desfecho das ações tomadas pelo administrador. Este possui um conjunto de opções que tem de ser registadas pela 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.

2.6.2. Diagrama de atividade Cliente

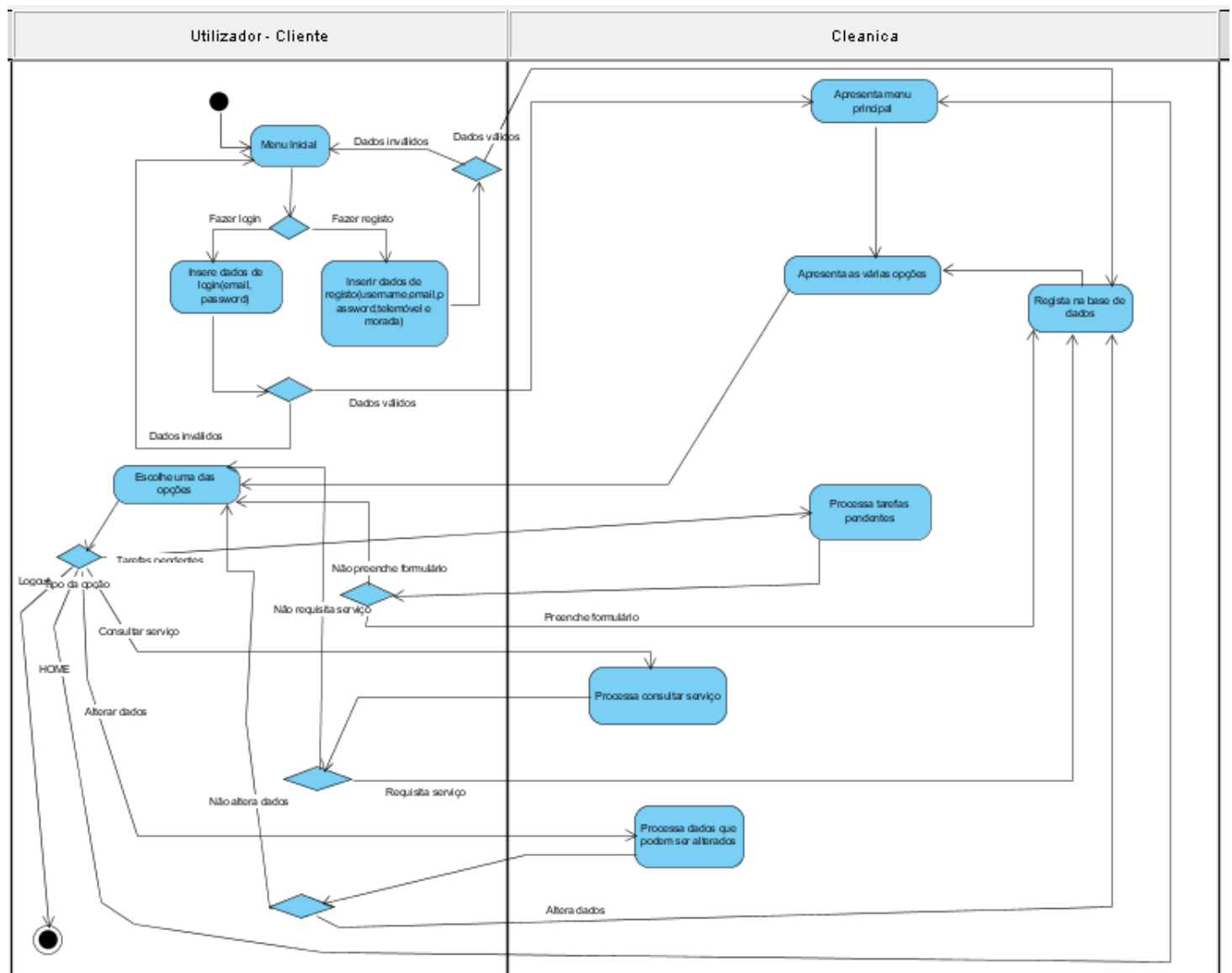


Figura 41 - Diagrama de Atividade Cliente

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 obvio do lado da aplicação.

2.6.3. Diagrama de atividade Funcionário

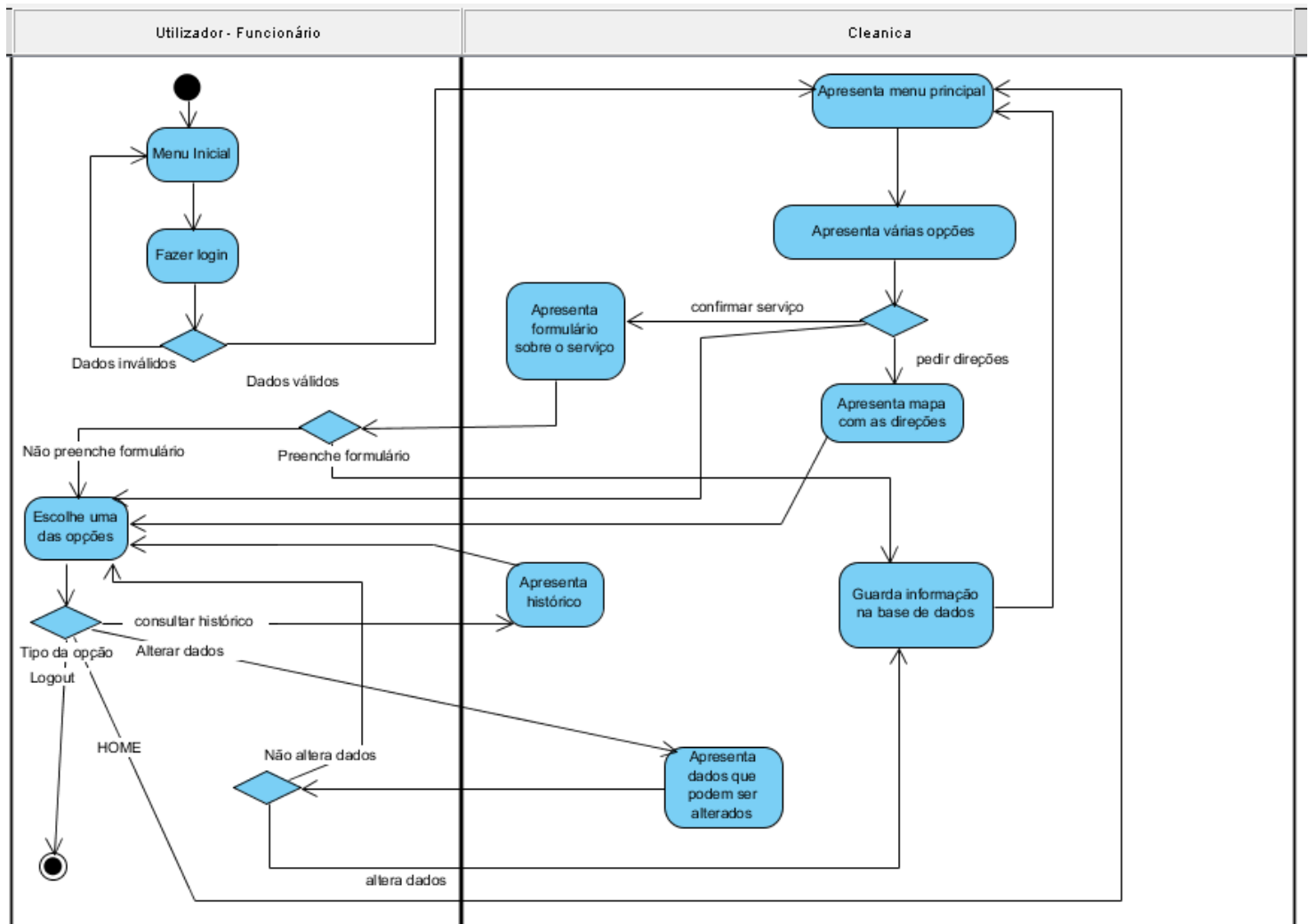


Figura 42 - Diagrama de Atividade Funcionário

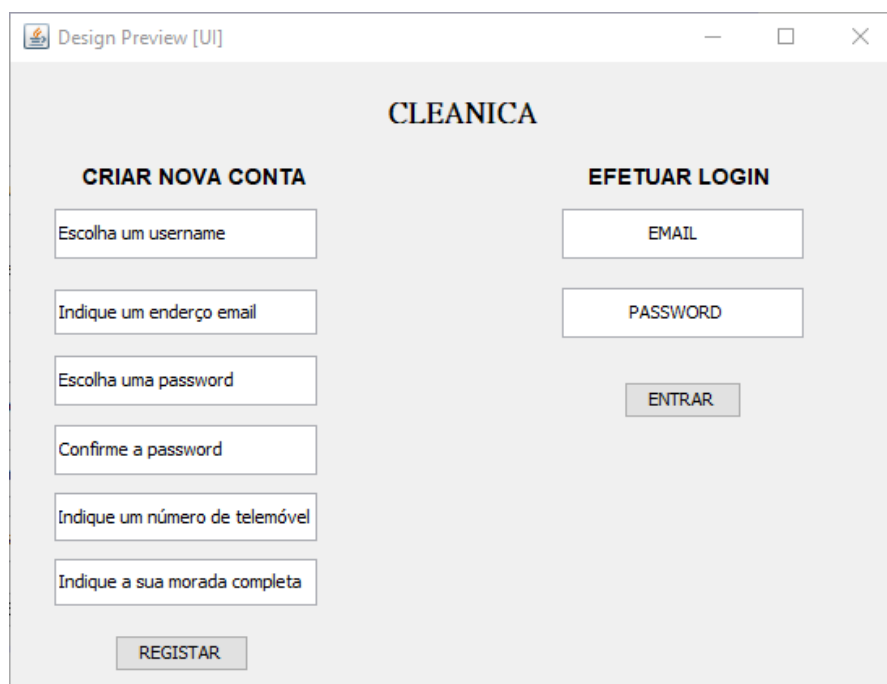
Por último, surge o diagrama de atividades relativo ao funcionário. A particularidade neste diagrama é que no menu “Home” deste, já se encontram algumas opções. Este pode usufruir destas, ou simplesmente, pode utilizar outros serviços tal como os restantes utilizadores.

3. 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 as todas as opções. Assim, de seguida especificamos individualmente cada uma das interfaces. É importante referir que sendo uma mockup, trata-se apenas de um protótipo, não sendo de todo, o modelo final.

➤ Menu de login e de registo



A mockup da interface de login e registo da Cleanica, apresentada numa janela de "Design Preview [UI]". O layout é dividido em duas colunas principais sob o título "CLEANICA".

Coluna da Esquerda: CRIAR NOVA CONTA

- Escolha um username
- Indique um endereço email
- Escolha uma password
- Confirme a password
- Indique um número de telemóvel
- Indique a sua morada completa
- Botão: REGISTRAR

Coluna da Direita: EFETUAR LOGIN

- EMAIL
- PASSWORD
- Botão: ENTRAR

Figura 43 - Mockup Login

3.1. Administrador

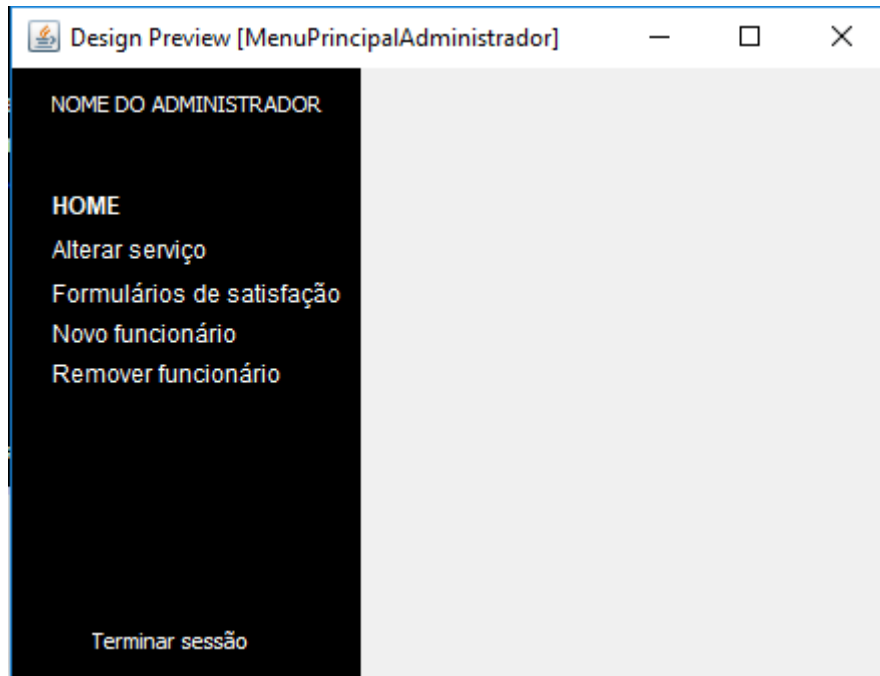


Figura 44 - Mockup Menu principal Administrador

No caso do administrador, o ecrã 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 um destas opções contém o seu próprio menu, abaixo, representado.

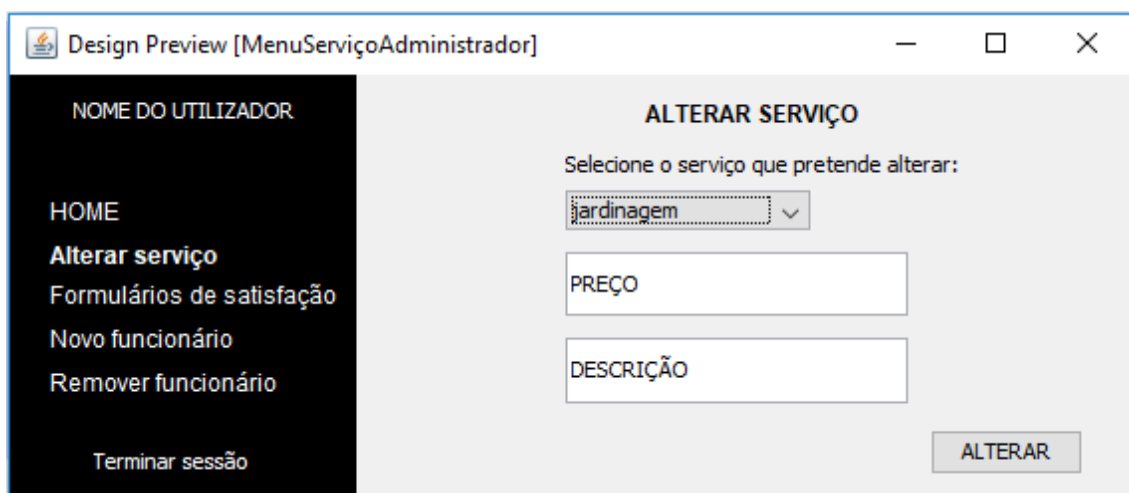


Figura 45 - Mockup Alterar serviço

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 46 - 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 47 - Mockup Novo Funcionário

De realçar que na figura 31 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.

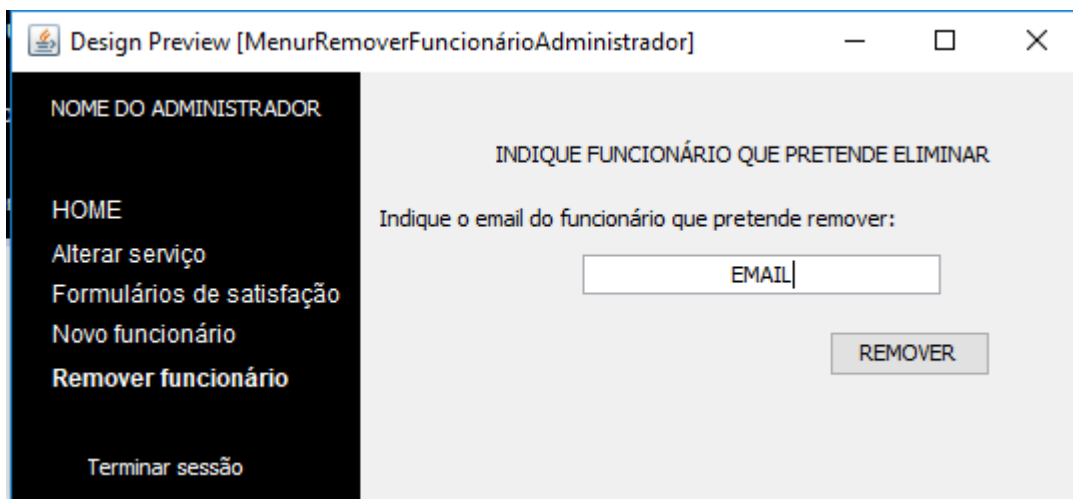


Figura 48 - Mockup Remover funcionário

3.2. Cliente

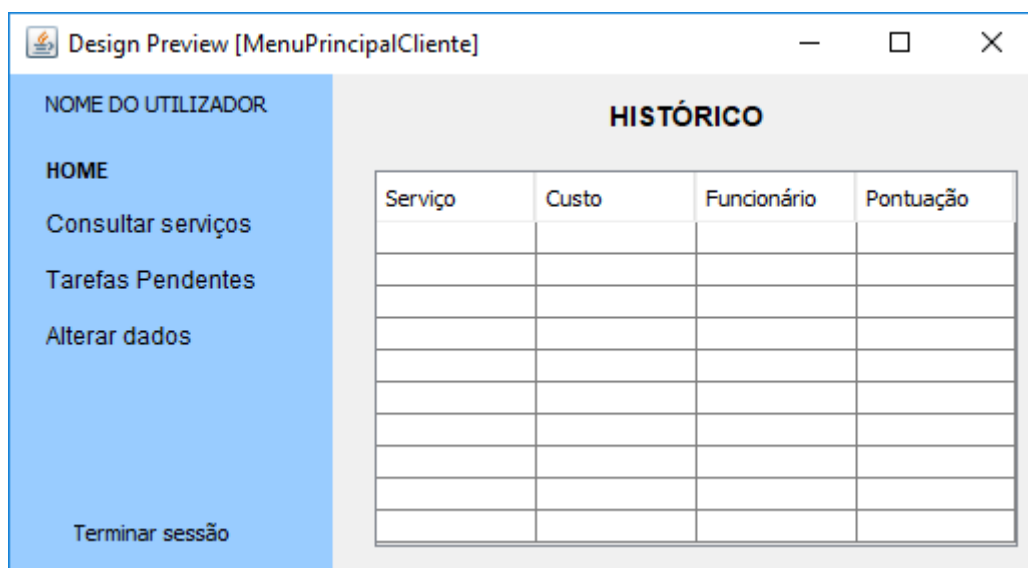


Figura 49 - Mockup Menu principal Cliente

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 50 - Mockup Consultar serviço

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

Figura 51 - Mockup Requisitar serviço lavagem automóvel

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

Figura 52 - Mockup Requisitar serviço, interiores, exteriores e piscina

A morada é indicada no registo para que o cliente não necessite constantemente de indica-la 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 53 - Mockup Alterar dados (cliente)

3.3. Funcionário

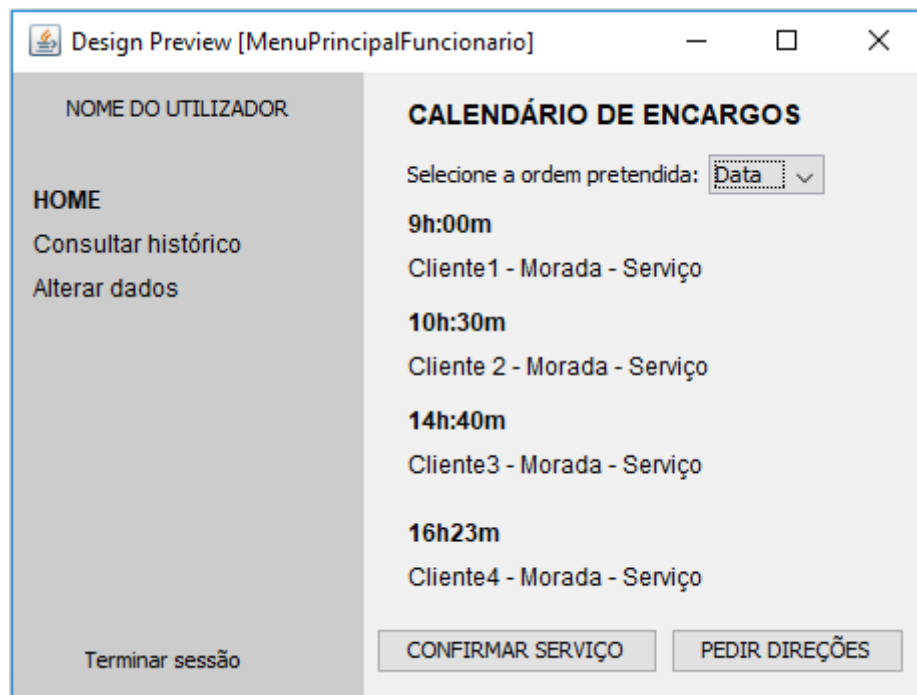


Figura 54 - Mockup Menu principal funcionário

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 formulário um formulário simples de confirmação do serviço e ainda, caso este ainda não tenha sido confirmado o funcionário, poderá ainda “Pedir direções” sobre a morada sobre a qual 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.

4. 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.

4.1. 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

4.2. 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

4.3. 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	Não	Não

			caracteres		
	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	Inteiro entre 0 e MAXINT	Não	Não

Serviços Realizados		realizar			
	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

Formulário de Realização	Id_FormulárioR	Identifica unicamente um Formulário de Realização	Inteiro entre 0 e MAXINT	Não	Não
	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

4.4. 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 lógico¹.

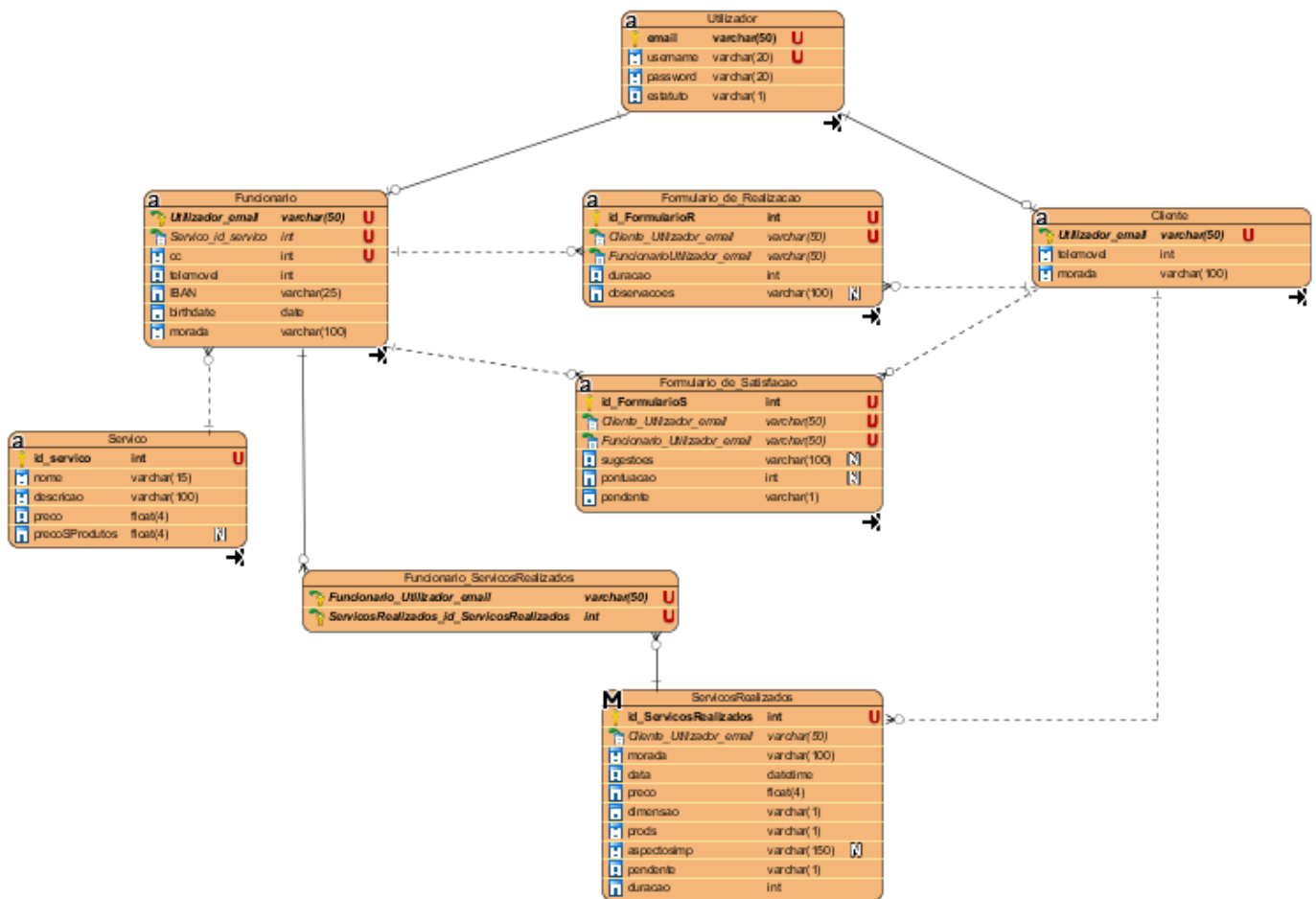


Figura 55 - Modelo lógico

¹ 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, se destaca 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*.

4.5. 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.

Referências

- Thomas Connolly, TC, Carolyn Begg, CB, (2005), *Database System – A Pratictal Approach to Designm Implementation, and Managements*, 4ªedição, Harlow: Pearson Education Limited

Lista de Siglas e Acrónimos

DSS *Diagrama de Sequência de Sistemas*

Anexos

I. Diagramas de Sequência

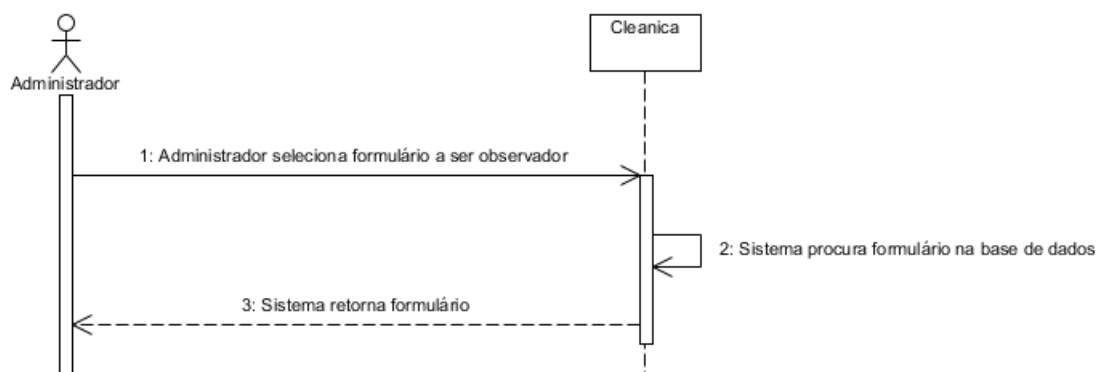


Figura 56 - Diagrama de sequência (Administrador): consultar formulário de satisfação

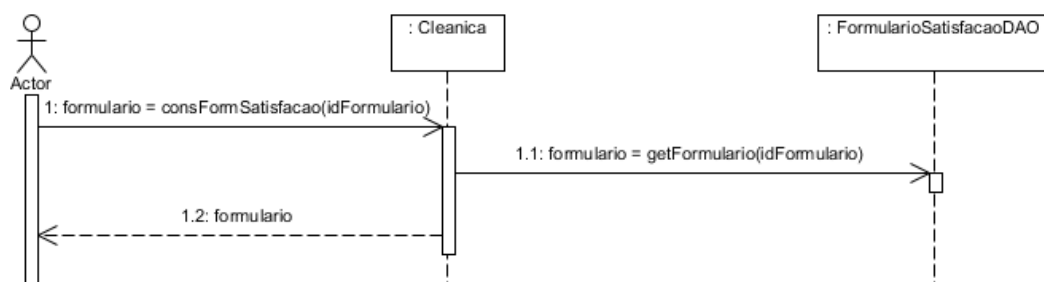


Figura 57 - Especificação do diagrama de sequência (Administrador): consultar formulário de satisfação

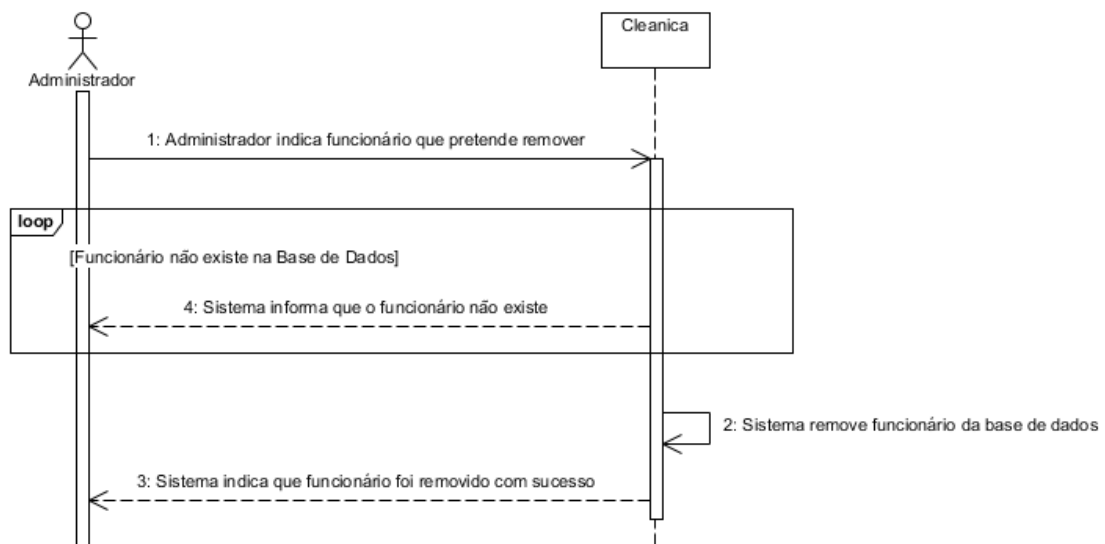


Figura 58 - Diagrama de sequência (Administrador): remover funcionário

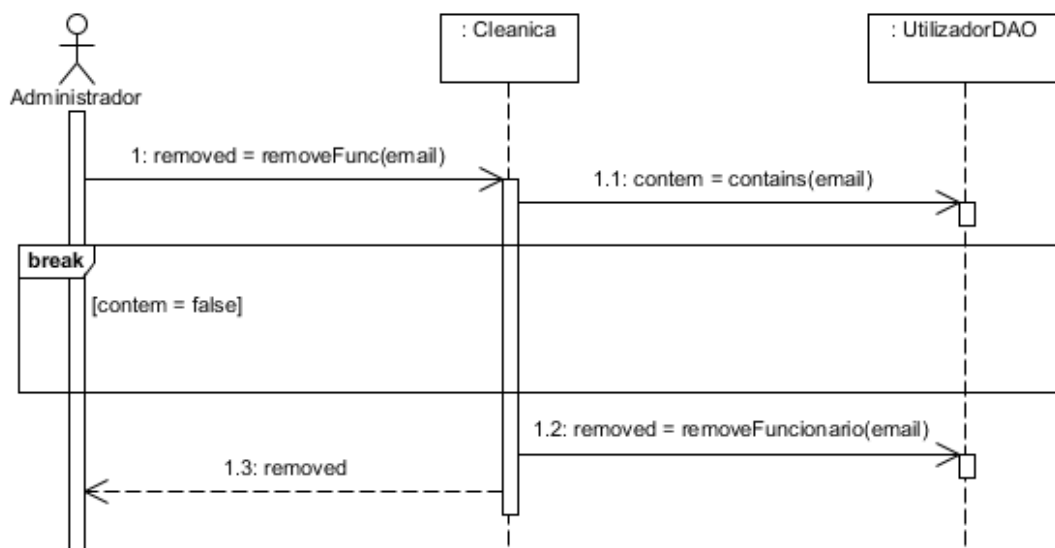


Figura 59 - Especificação do diagrama de sequência (Administrador): remover funcionário

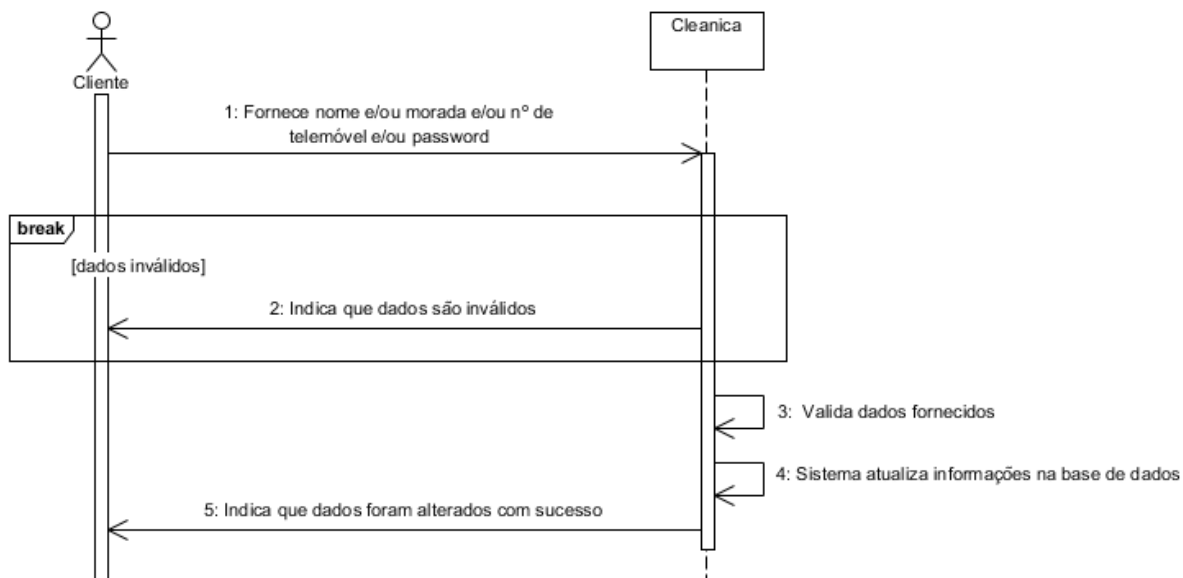


Figura 60 - Diagrama de sequência (cliente): alterar dados

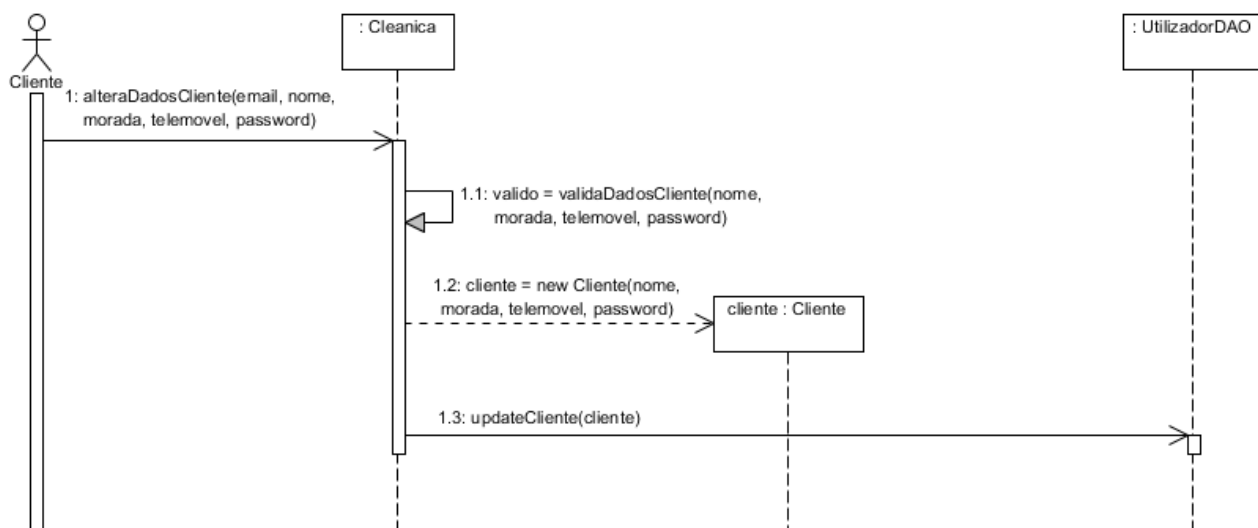


Figura 61 - Especificação do diagrama de sequência (cliente): alterar dados

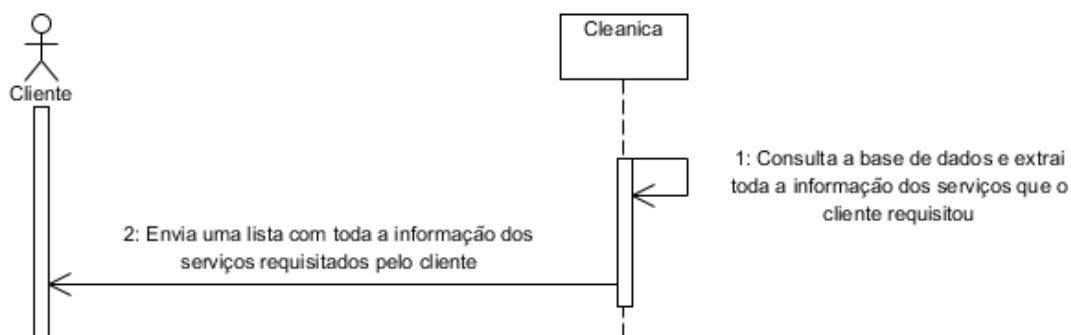


Figura 62 - Diagrama de sequência (cliente): consultar histórico

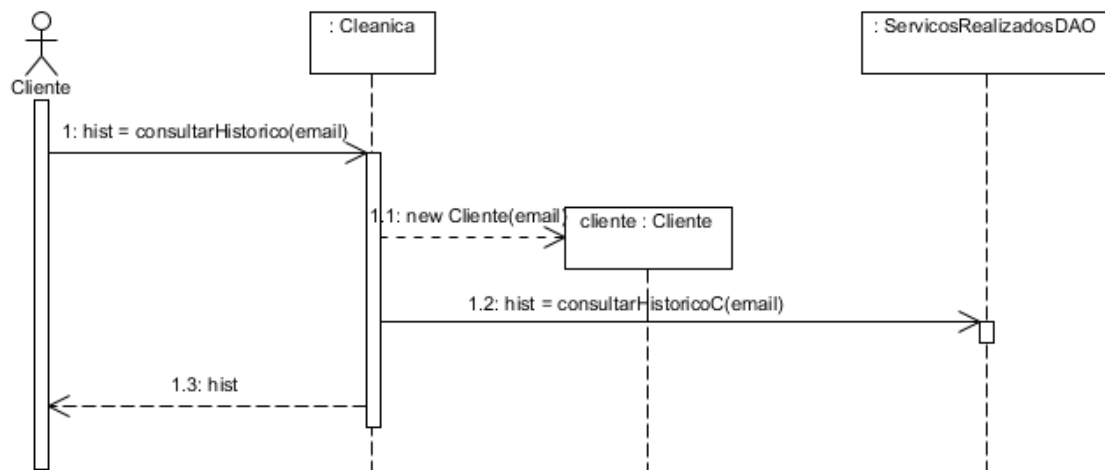


Figura 63 - Especificação do diagrama de sequência (cliente): consultar histórico

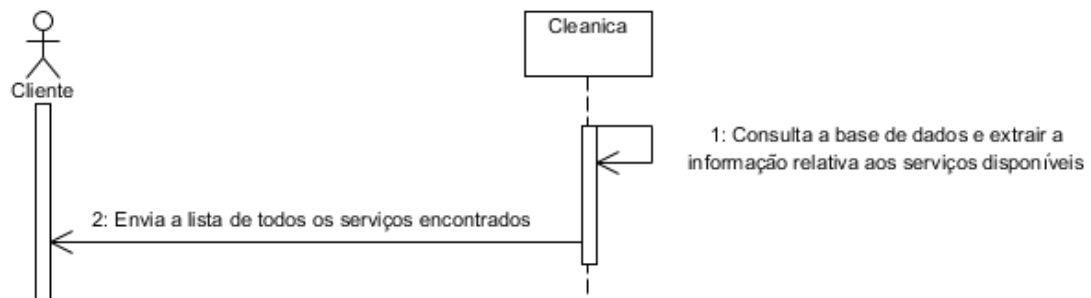


Figura 64 - Diagrama de sequência (cliente): consultar serviço

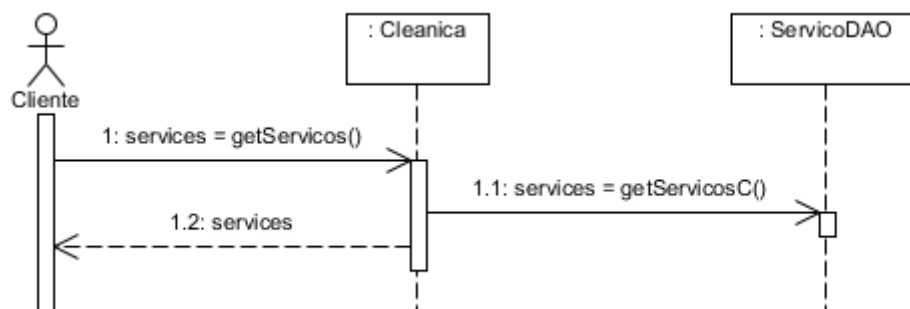


Figura 65 - Especificação do diagrama de sequência (cliente): consultar serviço

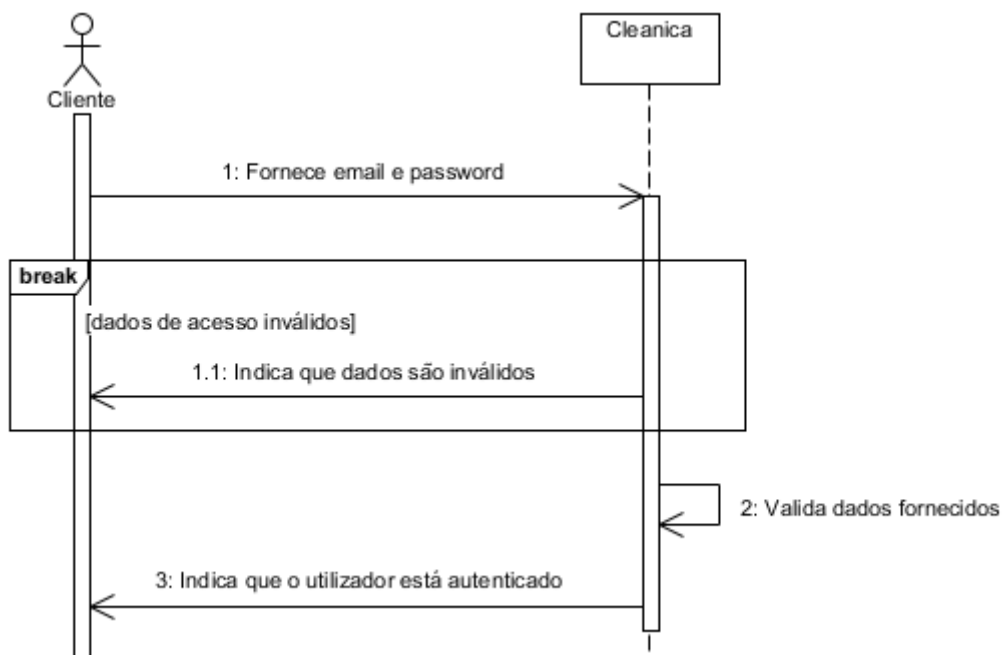


Figura 66 - Diagrama de sequência: efetuar Login

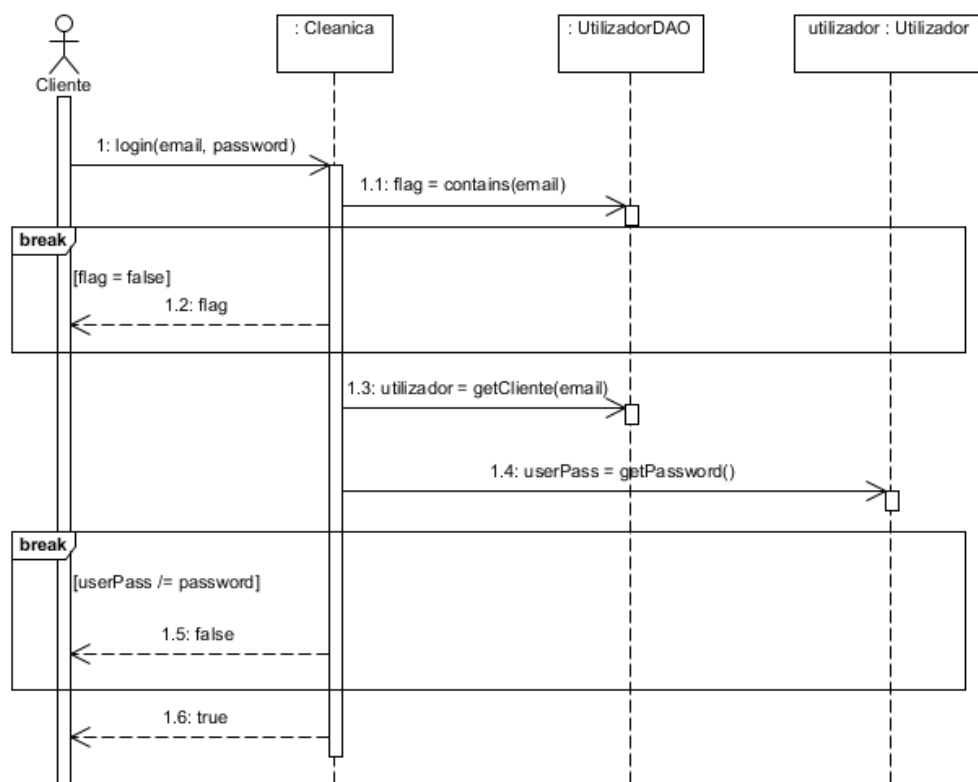


Figura 67 - Especificação do diagrama de sequência: efetuar Login

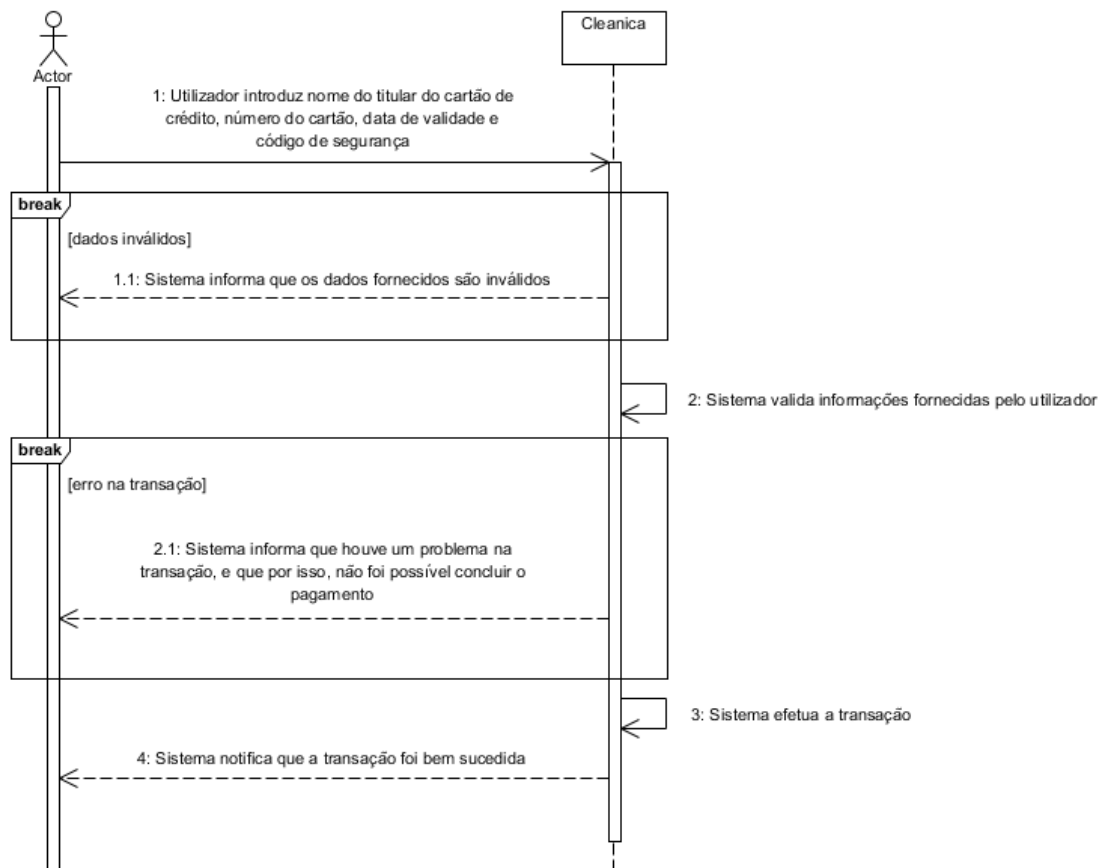


Figura 68 - Diagrama de sequência: efetuar pagamento

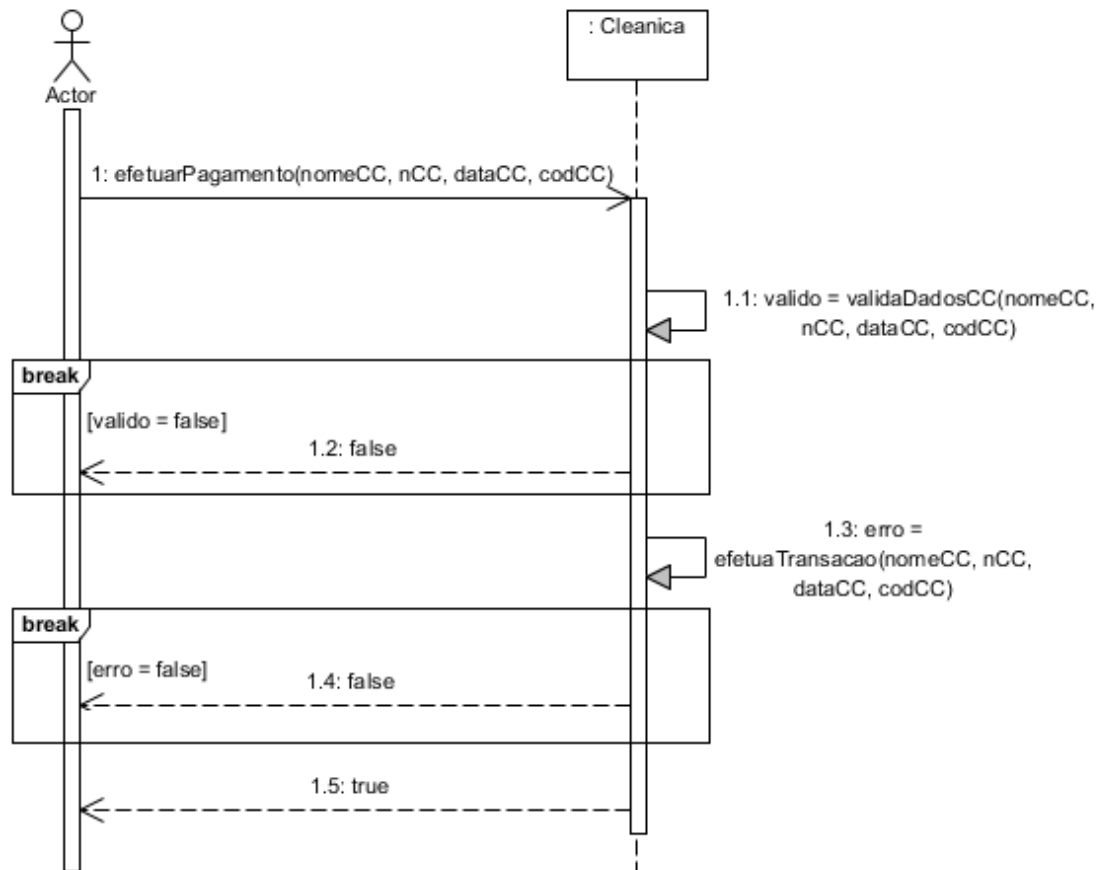


Figura 69 - Especificação do diagrama de sequência: efetuar pagamento

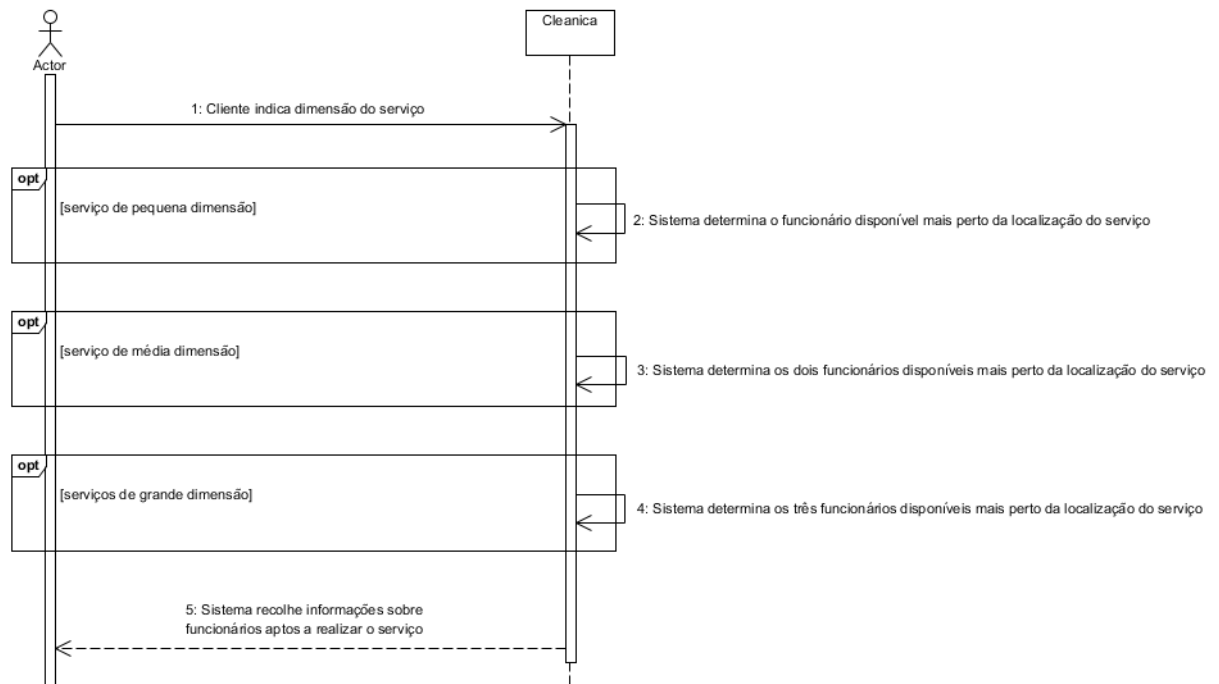


Figura 70 - Diagrama de sequência: escolher funcionário(s)

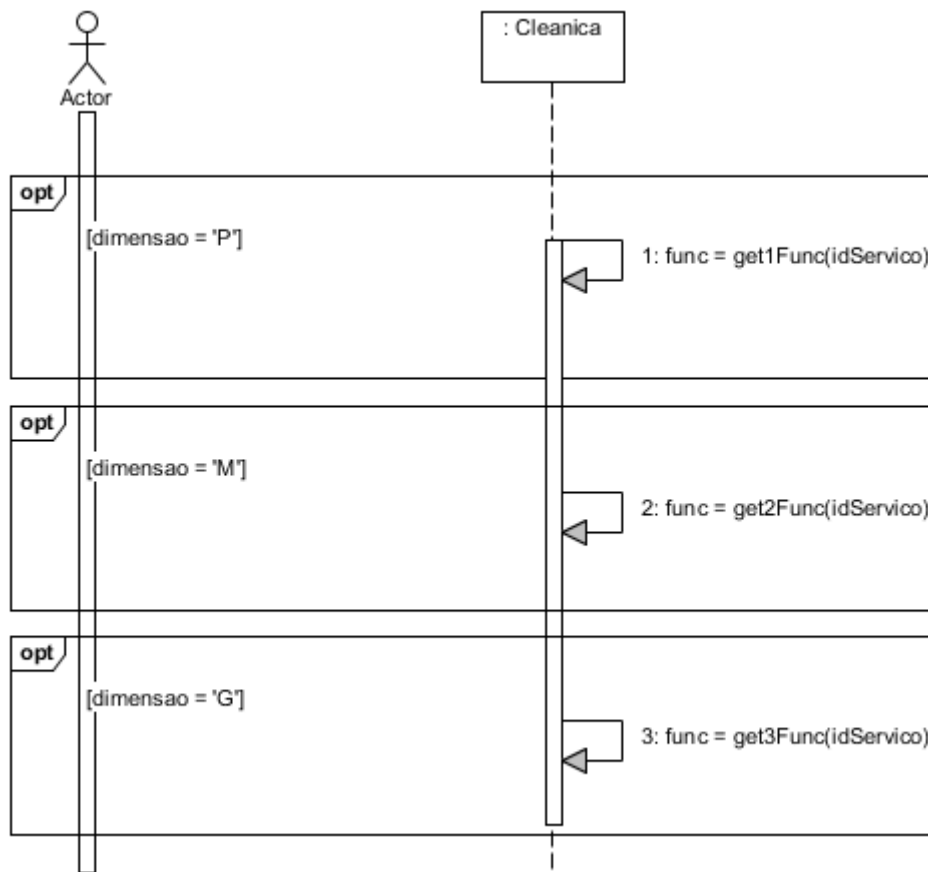


Figura 71 - Especificação do diagrama de sequência: escolher funcionário(s)

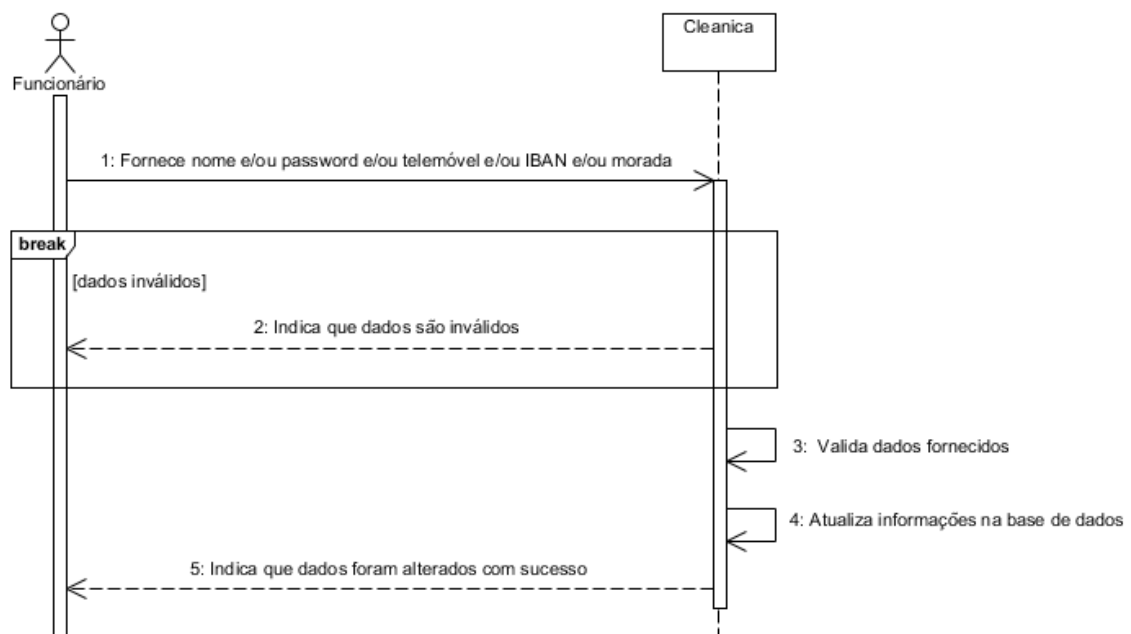


Figura 72 - Diagrama de sequência (funcionário): alterar dados

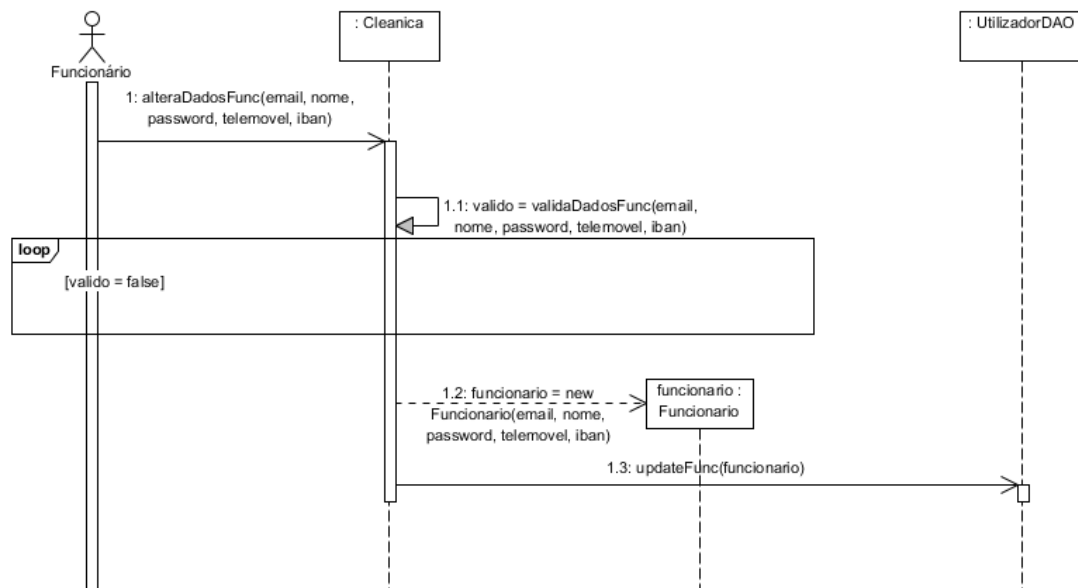


Figura 73 - Especificação do diagrama de sequência (funcionário): consultar serviço

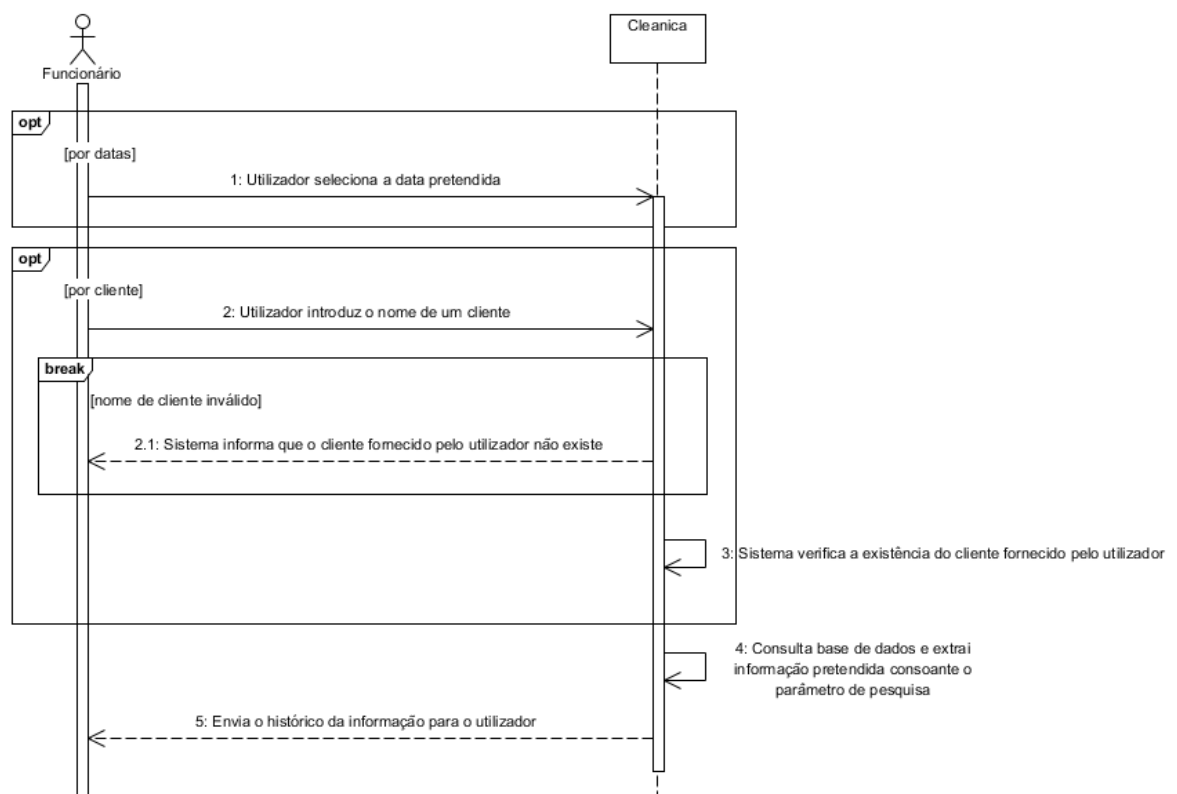


Figura 74 - Diagrama de sequência (funcionário): consultar histórico de encargos

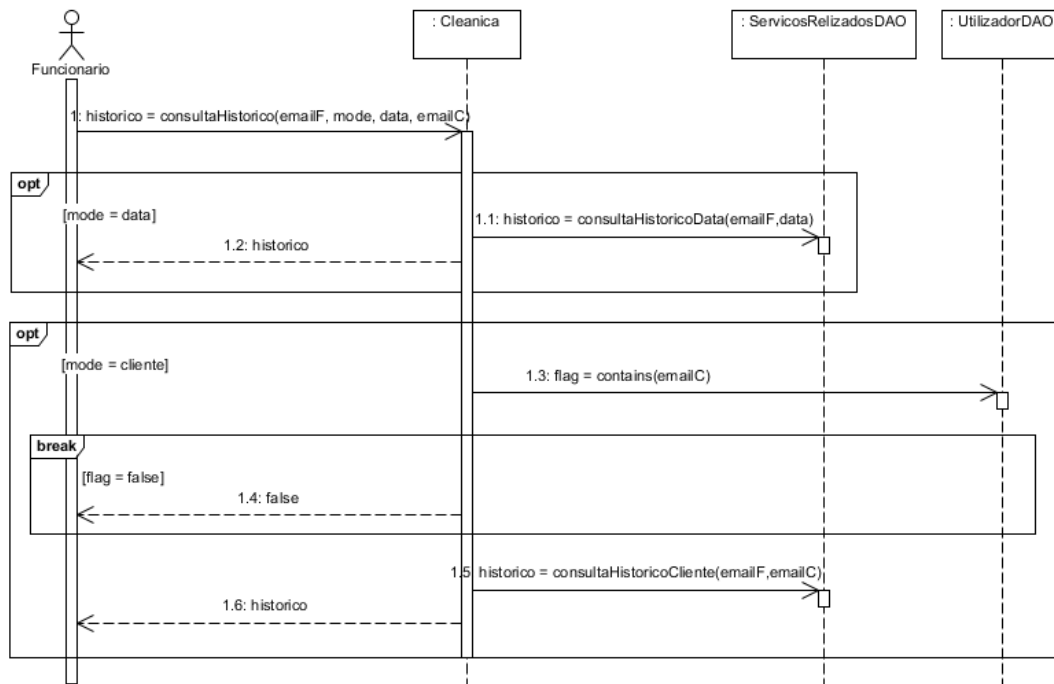


Figura 75 - Especificação do diagrama de sequência (funcionário): consultar histórico de encargos

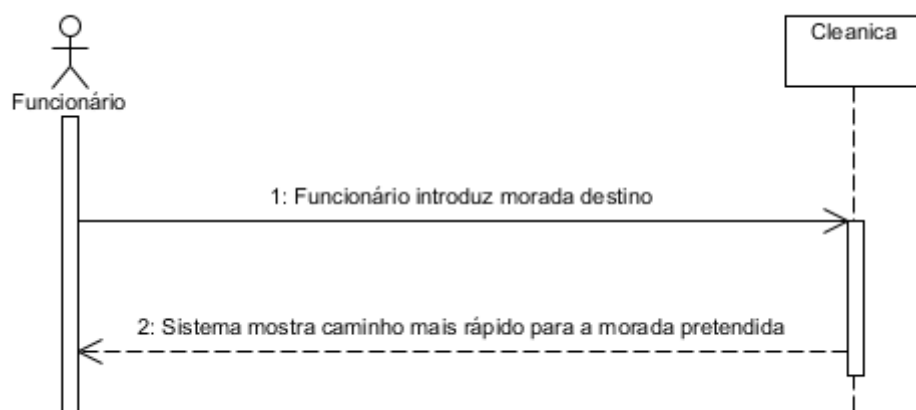


Figura 76 - Diagrama de sequência (funcionário): pedir direções

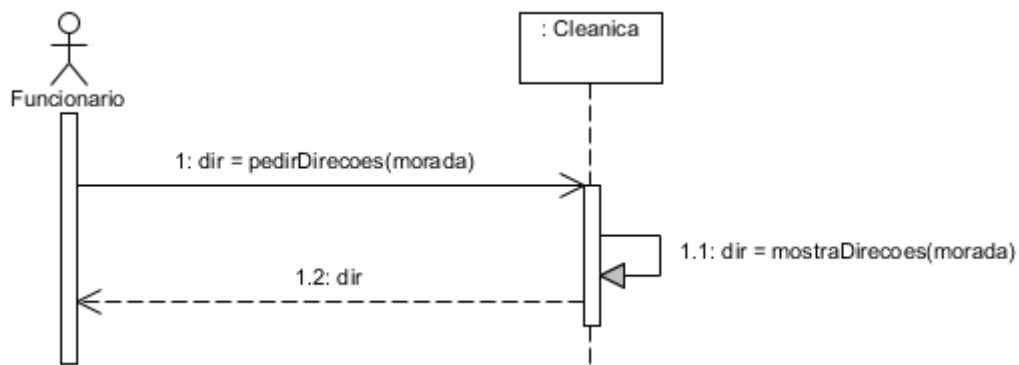


Figura 77 - Especificação do diagrama de sequência (funcionário): pedir direções

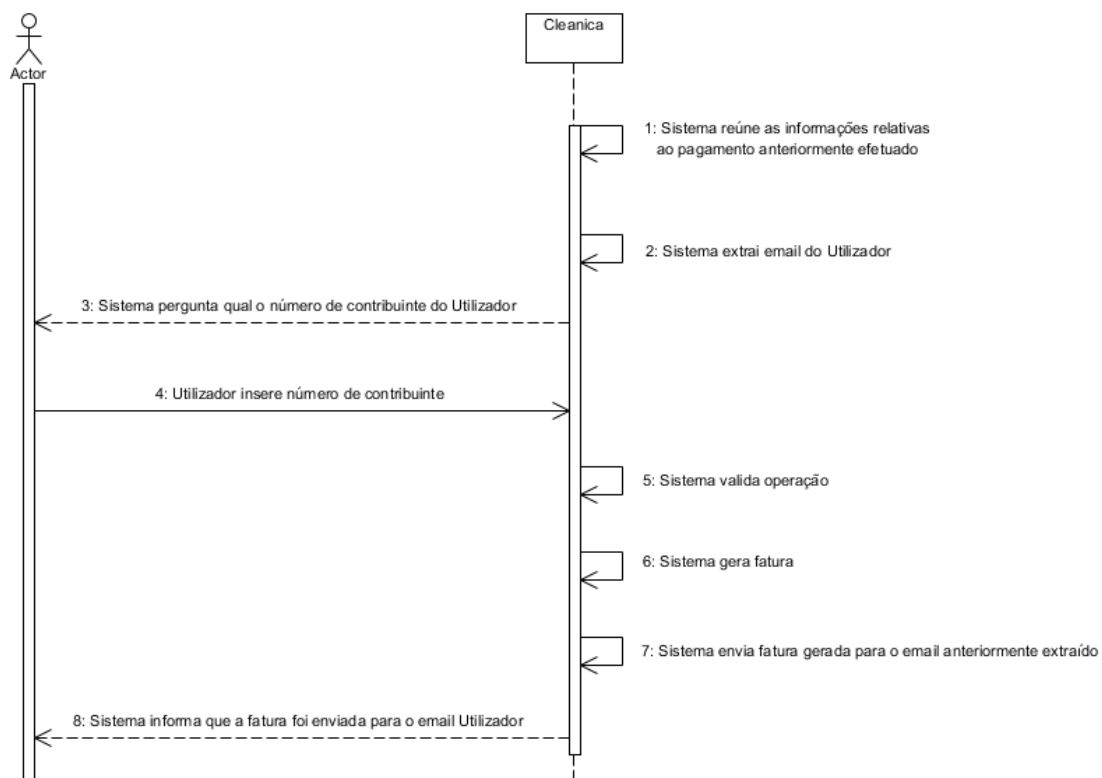


Figura 78 - Diagrama de sequência (funcionário): gerar fatura

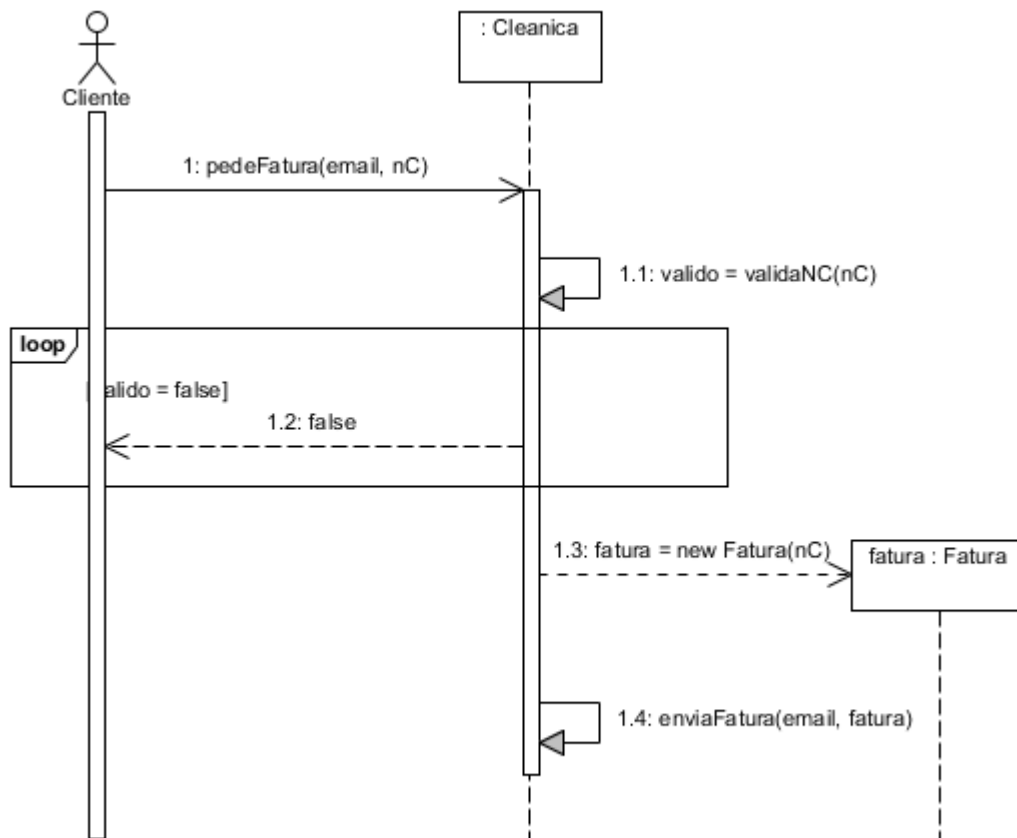


Figura 79 - Especificação do diagrama de sequência (funcionário): gerar fatura

II. Modelo Concetual

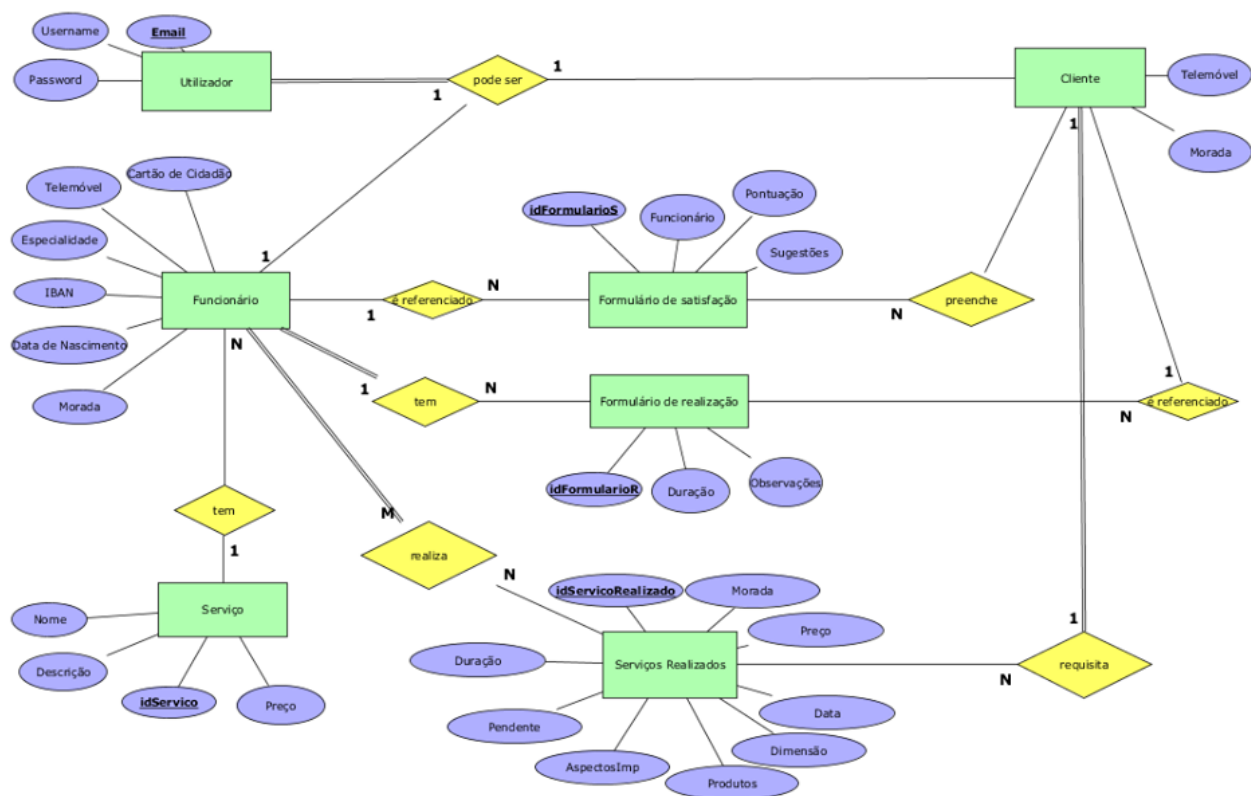


Figura 80 - Modelo Concetual

III. 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;

```