

UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

DSS - Relatório: Fase 2
Grupo 24

Ana Filipa Ribeiro Murta (A93284)
Ana Paula Oliveira Henriques (A93268)
Leonardo Araújo Freitas (A93281)
Rui Pedro Gomes Coelho (A58898)
Tiago André Leça Carneiro (A93207)

novembro, 2021



Conteúdo

1	Enquadramento do projeto	6
2	Carregamento dos dados	7
2.1	Funcionários	7
2.2	Clientes	7
2.3	Equipamentos	8
2.4	Orçamentos	8
2.5	Planos de Trabalho	8
2.6	Serviços	9
3	Armazenamento dos dados	10
4	Modelo de Domínio	11
5	Diagrama de Componentes	13
6	Diagrama de Packages	15
7	Diagrama de Classes	16
8	Diagramas de Sequências	18
8.1	Notificar Cliente	18
8.2	Serviço Expresso	19
8.3	Gestão das Estatísticas	19
8.4	Registar Pedido de Orçamento	20
8.5	Login	20

8.6	Consultar Plano	21
8.7	Levantar Equipamento	22
9	Manual de Utilização	23
10	Análise Crítica	26
11	Conclusão	27

Lista de Figuras

4.1	Modelo de Domínio	11
5.1	Diagrama de Componentes	13
6.1	Diagrama de Packages	15
7.1	Diagrama de Classes	16
8.1	contactarCliente	18
8.2	servicoExpresso	19
8.3	gestaoDeEstatisticas	19
8.4	gestaoDeOrcamento	20
8.5	verificaLogin	20
8.6	consultarPlano	21
8.7	gestaoDeEquipamentos	22
9.1	Menu Principal do Técnico e Rececionista	23
9.2	Menu Principal do Gestor	23
9.3	Submenu Cliente	23
9.4	Submenu Orçamento do Técnico	24
9.5	Submenu Orçamento do Rececionista	24
9.6	Submenu Orçamento do Gestor	24
9.7	Submenu Equipamento do Técnico	24

9.8	Submenu Equipamento do Rececionista	24
9.9	Submenu Equipamento do Gestor	24
9.10	Submenu Serviço do Técnico	25
9.11	Submenu Serviço do Rececionista	25
9.12	Submenu Serviço do Gestor	25
9.13	Submenu Relatório de Estatísticas do Gestor	25
9.14	Exemplo de uma consulta de um plano	25
9.15	Exemplo de uma alteração de utilizador	25

Capítulo 1

Enquadramento do projeto

No âmbito da unidade curricular Desenvolvimento de Sistemas de Software, foi realizado este trabalho prático que consiste na implementação de um Sistema de Gestão para Centros de Reparação de equipamentos eletrónicos, devendo garantir a gestão de todo o processo de reparação de equipamentos, desde a orçamentação até à entrega do equipamento.

Na primeira fase, desenvolveu-se o **Modelo de Domínio** e o **Diagrama de Use Cases**, responsáveis por representar as interações existentes entre as diversas entidades, de modo a haver uma perceção de como o sistema funcionará.

Na segunda fase, desenvolveu-se o **Diagrama de Componentes**, **Diagrama de Packages**, **Diagrama de Classes** e os **Diagramas de Sequências**. Em seguida, foi desenvolvido o código em Java, implementando a aplicação que cumpra todos os requisitos especificados.

Ao longo do relatório é mostrado os diagramas elaborados, juntamente com uma breve explicação sobre a construção dos mesmos. Para além disto, é, ainda, esclarecido como foi feito o carregamento dos dados, imprescindíveis para o funcionamento do sistema, bem como o seu armazenamento. Finalmente, é apresentada uma análise crítica do resultado obtido.

Durante a realização da segunda fase do trabalho, foram feitas algumas mudanças relativamente às decisões efetuadas na primeira fase, no Modelo de Domínio.

Capítulo 2

Carregamento dos dados

Uma estratégia adotada para facilitar a implementação deste sistema foi o carregamento de dados a partir de ficheiros de texto. Para esse efeito, temos a classe **Parser**, responsável por realizar o parsing dos dados sobre os funcionários, os clientes, os equipamentos, os orçamentos, os planos de trabalhos e os serviços, após terem sido lidos os ficheiros previamente mencionados.

2.1 Funcionários

Para o carregamento de objetos do tipo **FuncionariosFacade**, foi construído o ficheiro *dadosFuncionarios.txt*, adotando a seguinte estrutura:

username	password	tipo funcionário	equip recebidos	equip devolvidos
jackinthebooks;	bdgals12;	Rececionista;	KIN121989,MCNSHRYSD;	KIN121989;

Assim que o método **parseLogin()** realizar o respetivo carregamento, serão armazenados, num Map, todos os funcionários deste sistema, sendo a chave associada a cada objeto **Funcionario** o seu *username*. Este Map é uma variável de instância da classe **FuncionariosFacade**.

2.2 Clientes

Para o carregamento de objetos do tipo **ClientesFacade**, foi construído o ficheiro *dadosClientes.txt*, adotando a seguinte estrutura:

cliente (ID)	nome	NIF	telefone	email	equip (ID)
1928Wwd8372;	Ana Murta;	211092831;	933094729;	a93284@uminho.com;	LI0283EEE

Assim que o método **parseCliente()** realizar o respetivo carregamento, serão armazenados, num Map, todos os clientes deste sistema, sendo a chave associada a cada objeto **Cliente** o seu ID. Este Map é uma variável de instância da classe **ClientesFacade**.

2.3 Equipamentos

Para o carregamento de objetos do tipo `EquipamentosFacade`, foi construído o ficheiro *dadosEquipamentos.txt*, adotando a seguinte estrutura:

NIF	equip (ID)	estado
298788212;	CXX1299WQ;	reparado

Assim que o método `parseEquip()` realizar o respetivo carregamento, serão armazenados, num Map, todos os equipamentos deste sistema, sendo a chave associada a cada objeto `Equipamento` o NIF do cliente a que pertence. Este Map é uma variável de instância da classe `EquipamentosFacade`.

2.4 Orçamentos

Para o carregamento de objetos do tipo `OrcamentosFacade`, foi construído o ficheiro *dadosOrcamentos.txt*, adotando a seguinte estrutura:

orçamento (ID)	equip (ID)	data	custo	prazo
7;	KIN121989;	11-03-2021 14:50:16;	39;	PT31M;

técnico	nota	plano (ID)	estado
ruipgcoelho;	Falta peça;	NJFVI54S;	true;

Assim que o método `parseOrcamento()` realizar o respetivo carregamento, serão armazenados, num Map, todos os orçamentos deste sistema, sendo a chave associada a cada objeto `Orcamento` o seu ID. Este Map é uma variável de instância da classe `OrcamentosFacade`.

2.5 Planos de Trabalho

Para o carregamento de objetos do tipo `PlanoFacade`, foi construído o ficheiro *dadosPlano.txt*, adotando a seguinte estrutura:

plano (ID)	orçamento (ID)	técnico	custo	prazo	passos
8;	8;	Leonardo1924;	72.0;	PT46M	testar,72.0,PT46M

Assim que o método `parsePlano()` realizar o respetivo carregamento, serão armazenados, num Map, todos os planos de trabalho deste sistema, sendo a chave associada a cada objeto `PlanoTrabalho` o seu ID. Este Map é uma variável de instância da classe `PlanoFacade`.

2.6 Serviços

Para o carregamento de objetos do tipo `ServicosFacade`, foi construído o ficheiro *dadosServico.txt*, adotando a seguinte estrutura:

<code>serviço (ID)</code>	<code>tipo serviço</code>	<code>plano (ID)</code>	<code>custo total</code>	<code>tempo total</code>
<code>1;</code>	<code>expresso;</code>	<code>1;</code>	<code>10.0;</code>	<code>PT10M;</code>

Assim que o método `parseServico()` realizar o respetivo carregamento, serão armazenados, num Map, todos os orçamentos deste sistema, sendo a chave associada a cada objeto `Servico` o seu ID. Este Map é uma variável de instância da classe `ServicosFacade`.

Capítulo 3

Armazenamento dos dados

Depois de efetuar o *login* e as suas credenciais serem autenticadas, o utilizador (um funcionário) poderá alterar dados já existentes no sistema ou, então, criar novos dados. Antes de efetuar o *logout*, estes dados terão de ser armazenados nos respetivos ficheiros de modo a não se perder informação necessária ao funcionamento do sistema. Para isso, temos a classe `Saver`, que é responsável por processar este armazenamento.

Nesta classe, encontra-se o método estático `gravar(IStoreLN store store)`, que, tal como o nome sugere, grava toda a informação da `Store` carregada e modificada pelo utilizador nos respetivos ficheiros a fim de serem futuramente carregados com as novas atualizações.

O código abaixo apresenta um exemplo desta gravação no ficheiro *dadosFuncionarios.txt*:

```
Map<String,Funcionario> func = store.getFuncionariosFacade().getFuncionarios();

try{
    File fileFunc = new File("input/dadosFuncionarios.txt");
    if(!(fileFunc.exists())) fileFunc.createNewFile();
    PrintWriter printFunc = new PrintWriter(fileFunc);
    for(Funcionario funcionario : func.values()){
        funcionario.gravar(printFunc);
    }
    printFunc.close();
}
catch(IOException e) {
    e.printStackTrace();
}
```

Em que o método `gravar(printFunc)`, implementado na classe `Funcionario`, imprime o objeto em questão no respetivo ficheiro com o auxílio do `PrintWriter`:

```
public void gravar(PrintWriter print){
    String recebidos = String.join(",",this.equipamentosRecebidos);
    String devolvidos = String.join(",",this.equipamentosDevolvidos);
    print.println(this.username + ";" + this.password + ";"
        + this.tipo + ";" + recebidos + ";" + devolvidos);
}
```

Modelo de Domínio

Com a inicialização da segunda fase deste projeto prático, decidiu-se realizar algumas alterações, que considerámos mais apropriadas, ao **Modelo de Domínio** elaborado na primeira fase de modo a descrever mais objetivamente as interações que ocorrem no sistema a implementar. Estas alterações foram, também, tomadas em consideração em sequência da análise dos comentários feitos relativamente às entregas da primeira fase do trabalho.

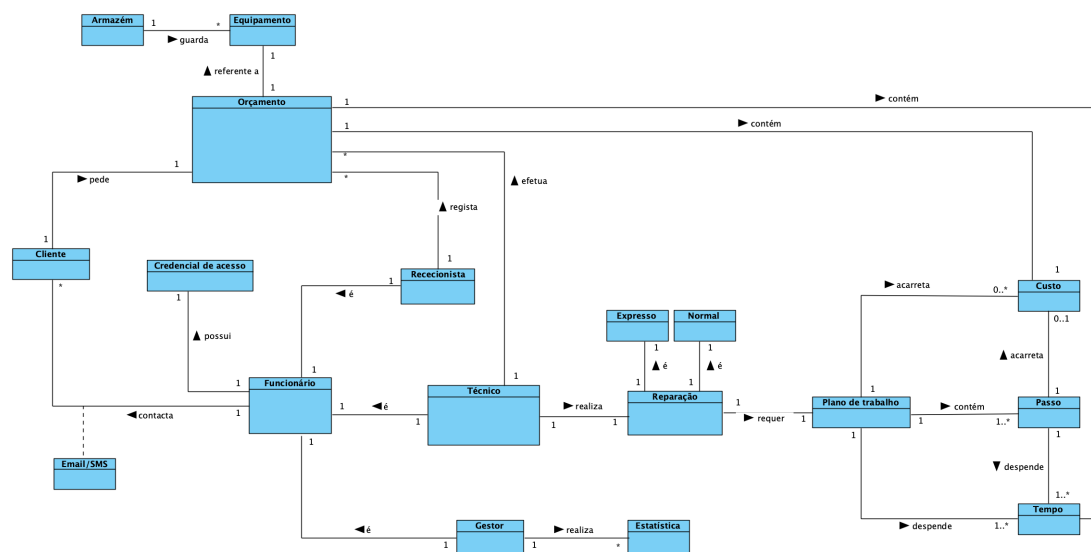


Figura 4.1: Modelo de Domínio

Por conseguinte, efetuou-se as seguintes mudanças:

- A entidade **Funcionário** foi adicionada, representando que um **Gestor**, um **Técnico** e um **Rececionista** são funcionários. Tomou-se esta decisão em função de que qualquer tipo de **Funcionário** pode entrar em contacto com o **Cliente** e, no modelo anterior, estava apenas representada essa interação entre o **Técnico** e o **Cliente**.
- As relações entre **Notificação** e **Email** e entre **Notificação** e **SMS** foram removidas porque uma **Notificação** não é um **Email** nem uma **SMS**, tal como estava antes descrito. Deste

modo, a entidade **Email/SMS** passou a ser uma classe de associação já que esta não é uma característica nem do **Cliente**, nem do **Funcionário**, mas sim da relação entre ambas.

- A entidade **Estatística** foi adicionada, sendo estabelecida uma relação entre esta e o **Gestor**, sendo que apenas este tipo de funcionário pode realizar estatísticas.
- A relação entre **Gestor** e **Rececionista** e a relação entre **Gestor** e **Técnico**, descritas como “avalia’”, foram removidas porque não há nenhuma funcionalidade no nosso sistema em que o **Gestor** avalia outro **Funcionário**.

Capítulo 5

Diagrama de Componentes

Após a análise do domínio do problema e dos seus requisitos, elaborada na primeira fase, procedeu-se ao desenvolvimento do **Diagrama de Componentes**, cujo objetivo é organizar o modelo em subsistemas com algum grau de complexidade. Cada subsistema desempenha uma função diferente.

Esta prática mostra o relacionamento estabelecido entre os diferentes componentes de um sistema, tornando-se bastante útil aquando da inicialização de um sistema de software já que conseguimos agrupar os métodos que melhor se adequam a cada um dos subsistemas.

Como cada subsistema implementa uma interface com os métodos a que lhe estão associados — cada subsistema funciona completamente encapsulada —, são implementadas **Facades**, que esconde as complexidades de um sistema maior e fornece uma interface mais simples ao cliente. Isto possibilita a interação com outros subsistemas através da API de cada Facade e, para tal, não precisa de conhecer a organização interna desse subsistema.

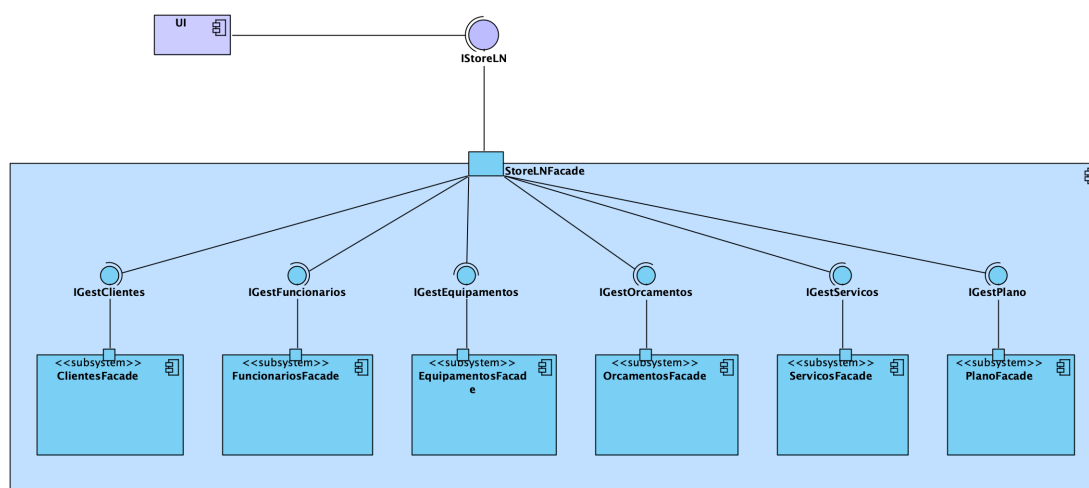


Figura 5.1: Diagrama de Componentes

Assim sendo, foram decididos os seguintes subsistemas:

- **EquipamentosFacade**: Trata de toda a gestão dos equipamentos e implementa a **IGestEquipamentos**.
- **FuncionariosFacade**: Trata de toda a gestão dos funcionários e implementa a **IGestFuncionarios**.
- **ClientesFacade**: Trata de toda a gestão dos clientes subsistema implementa a **IGestClientes**.
- **OrcamentosFacade**: Trata de toda a gestão dos orçamentos e implementa a **IGestOrcamentos**.
- **ServicosFacade**: Trata de toda a gestão de serviços e implementa a **IGestServicos**.
- **PlanoFacade**: Trata de toda a gestão dos planos de trabalho e implementa a **IGestPlano**.

Por fim, o *User Interface* (UI) da Store tem acesso às funcionalidades deste sistema de reparações através da interface **IStoreLN**, que, por sua vez, acede às interfaces implementadas pelos subsistemas apontados anteriormente.

Capítulo 6

Diagrama de Packages

O passo realizado a seguir foi o desenvolvimento do **Diagrama de Packages**, usado para mostrar, em forma de packages, a organização e disposição de vários elementos de modelos, juntamente com as dependências entre os vários packages. Isto proporcionou, por isso, uma maior facilidade em organizar visualmente a arquitetura do nosso sistema, sofrendo algumas atualizações conforme a evolução do projeto.

Deste modo, o package UI e o package InPutOutPut importam, separadamente, o package StoreLN, que, por sua vez, contém a classe StoreLNFacade e os packages dos subsistemas. Cada um destes subsistemas inclui as classes que lhe foram atribuídas durante a concretização do sistema.

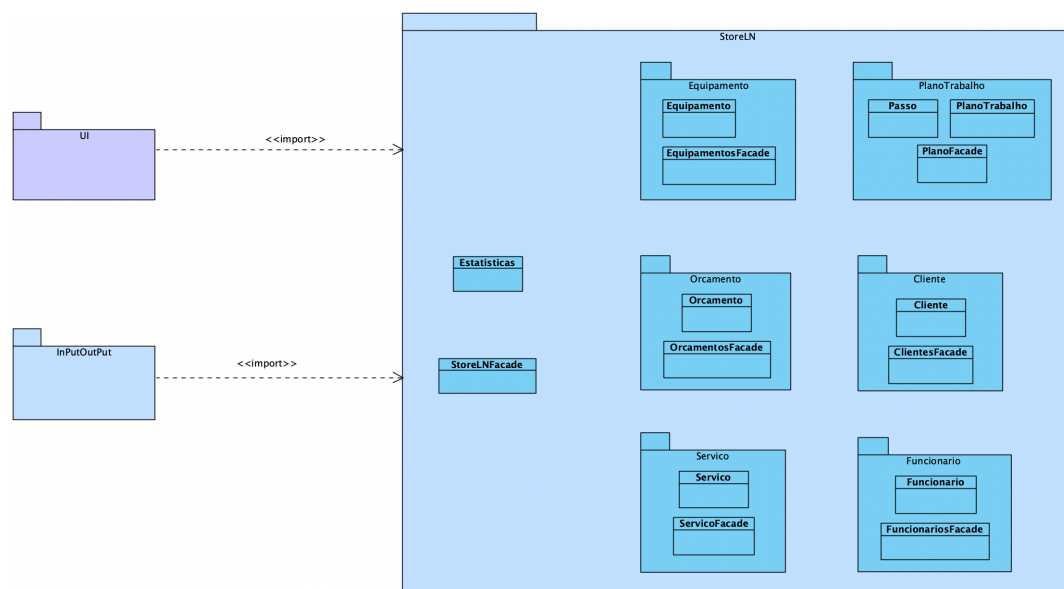


Figura 6.1: Diagrama de Packages

Capítulo 7

Diagrama de Classes

O fluxo do programa é representado a partir do **Diagrama de Classes** e da definição das interfaces que serão necessárias, que, também, permitem mostrar os métodos e as variáveis de instância de cada classe implementada no projeto.

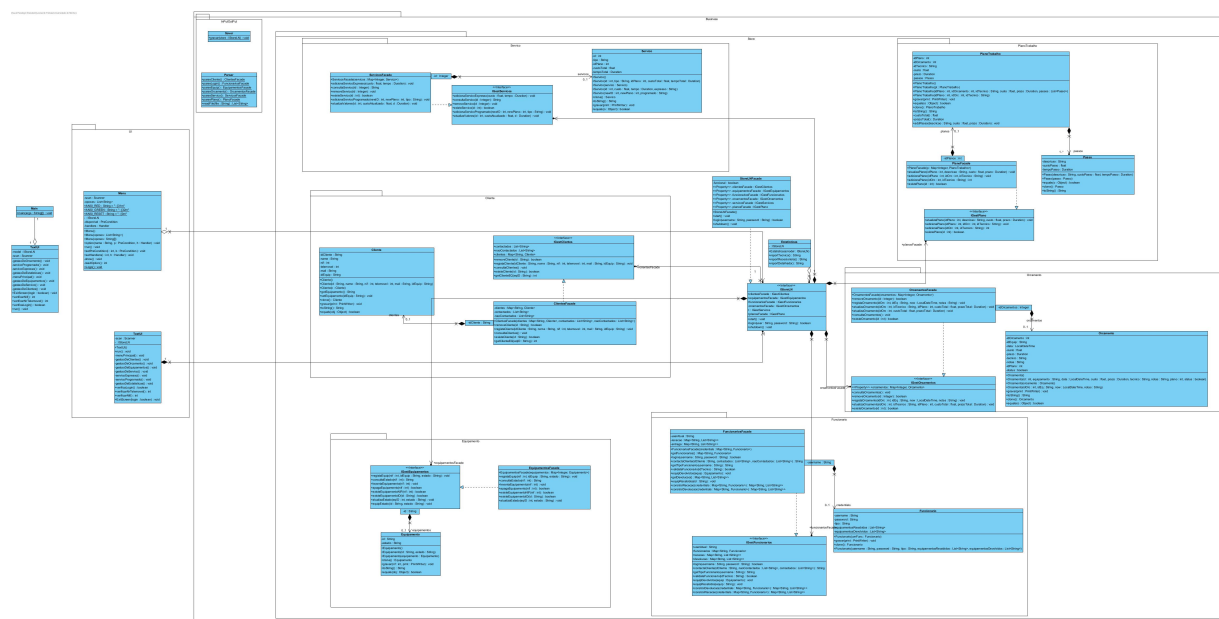


Figura 7.1: Diagrama de Classes

No diagrama desenvolvido, podemos observar algumas das classes mais fundamentais para o funcionamento do nosso sistema. No centro da representação, temos a função que unifica este programa, **StoreLNFacade**, que tem como objetivo guardar as informações lidas pela classe *Parse* em memória, para que possam ser utilizadas pelo sistema.

Para além disto, podemos observar as várias classes que são lidas por esta mesma classe: a classe **Funcionario**, **Equipamento**, **Cliente**, **Orcamento**, **Servico** e a classe **PlanoTrabalho**. Todas as classes anteriormente referidas têm uma correspondente classe Facade, que, por sua vez, incluem uma interface específica.

Depois de serem carregadas as informações destas classes, são apresentadas as várias opções a que um utilizador pode recorrer, através da classe **TextUI**. Esta classe recorre à memória carregada para poder fazer face aos pedidos do utilizador do sistema.

Capítulo 8

Diagramas de Sequências

Posteriormente, procedeu-se ao arranque da **fase de implementação** da aplicação pretendida, agora com uma perspetiva mais clara e objetiva das funcionalidades do sistema e de como as implementar, graças ao trabalho feito na primeira fase deste projeto e na fase de conceção.

Deste modo, através da análise de requisitos e de uma boa conceção da arquitetura e do comportamento do sistema, foi possível programar o software com uma maior facilidade visto que já havia sido construído uma ideia mais consistente das interações de todos os seus componentes. Além disto, sendo este um projeto com um grau de complexidade considerável, a divisão da sua implementação em blocos modulares a fim de simplificar a utilização de um subsistema complexo e, assim, conceber a solução mais eficiente exequível.

8.1 Notificar Cliente

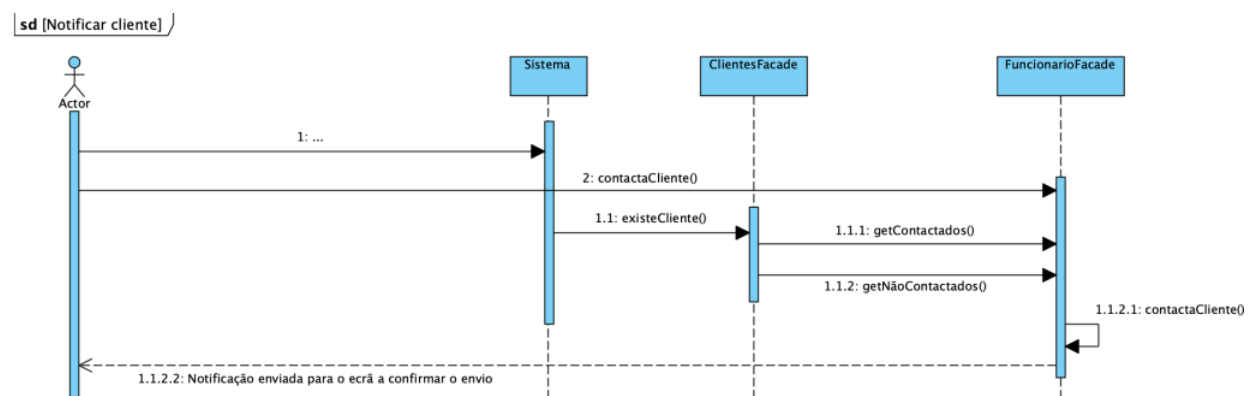


Figura 8.1: contactarCliente

8.2 Serviço Expresso

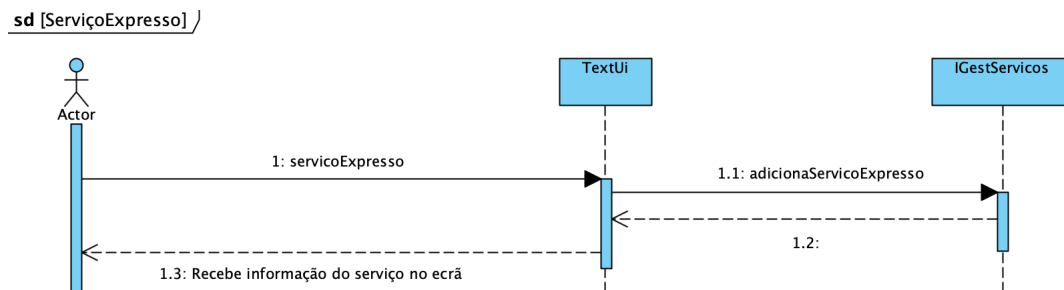


Figura 8.2: servicoExpresso

8.3 Gestão das Estatísticas

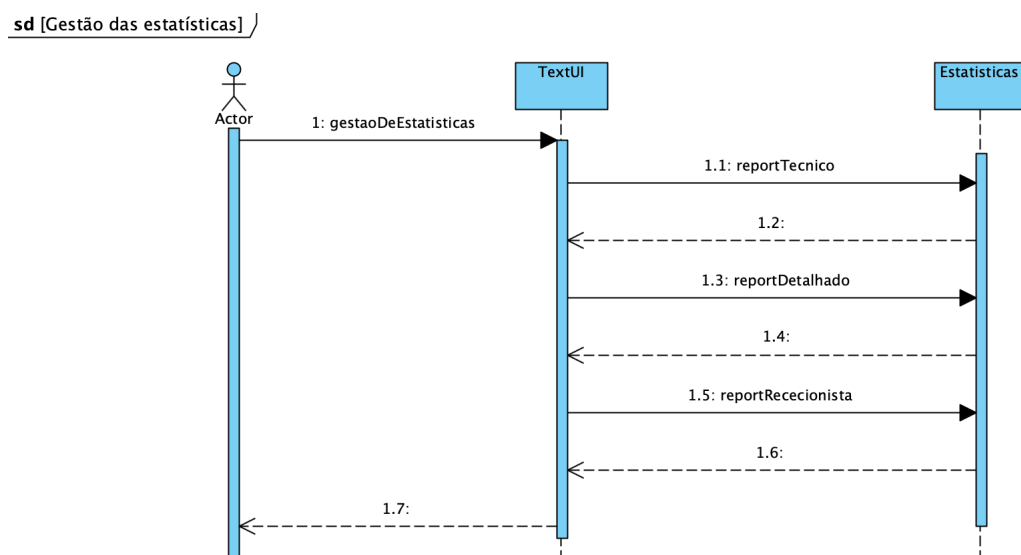


Figura 8.3: gestaoDeEstatisticas

8.4 Registrar Pedido de Orçamento

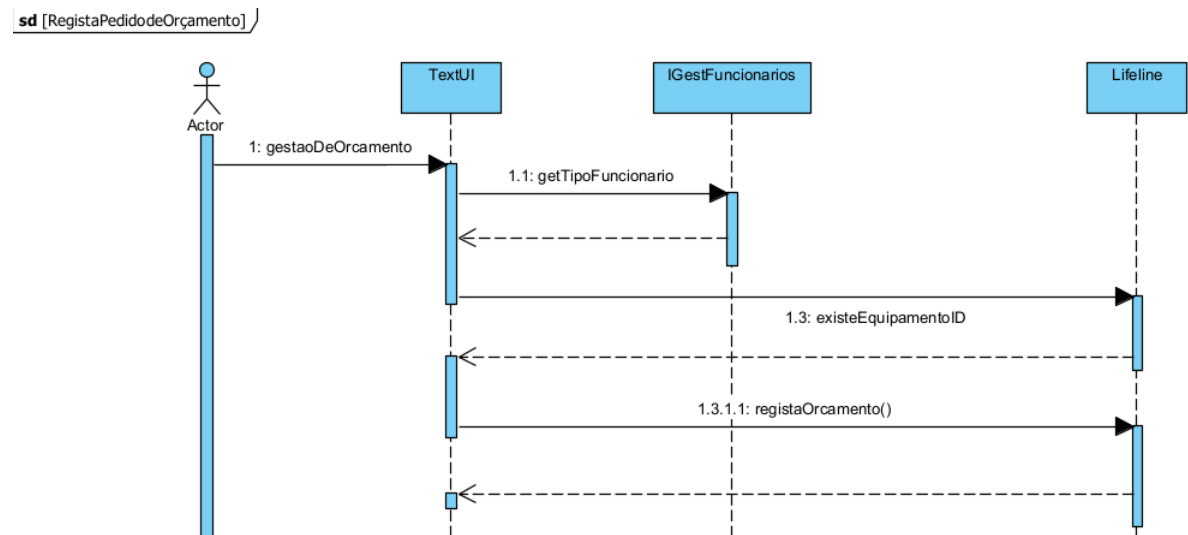


Figura 8.4: gestaoDeOrcamento

8.5 Login

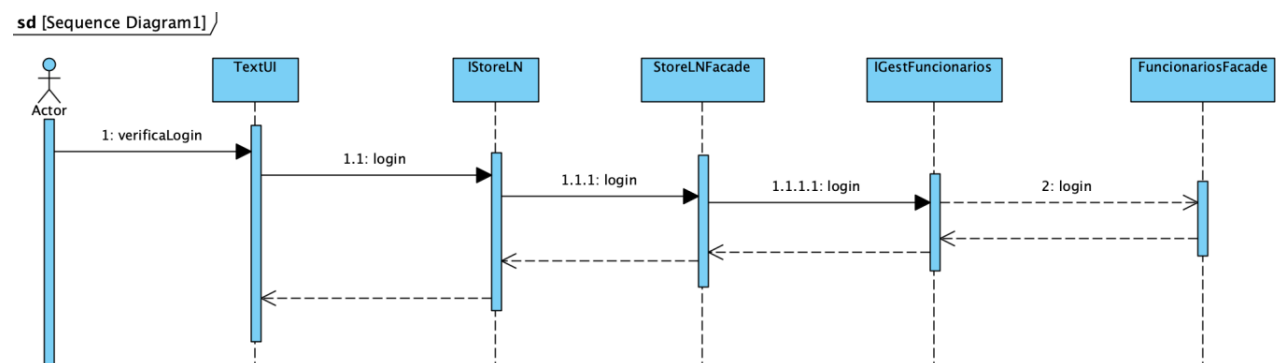


Figura 8.5: verificaLogin

8.6 Consultar Plano

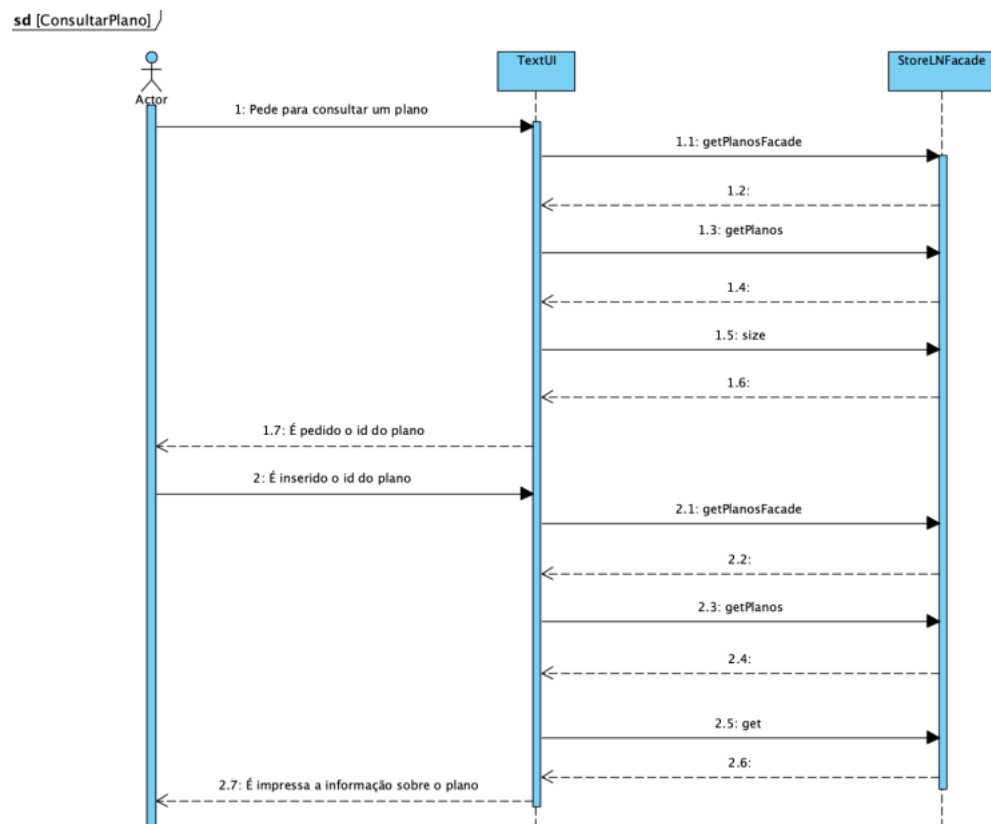


Figura 8.6: consultarPlano

8.7 Levantar Equipamento

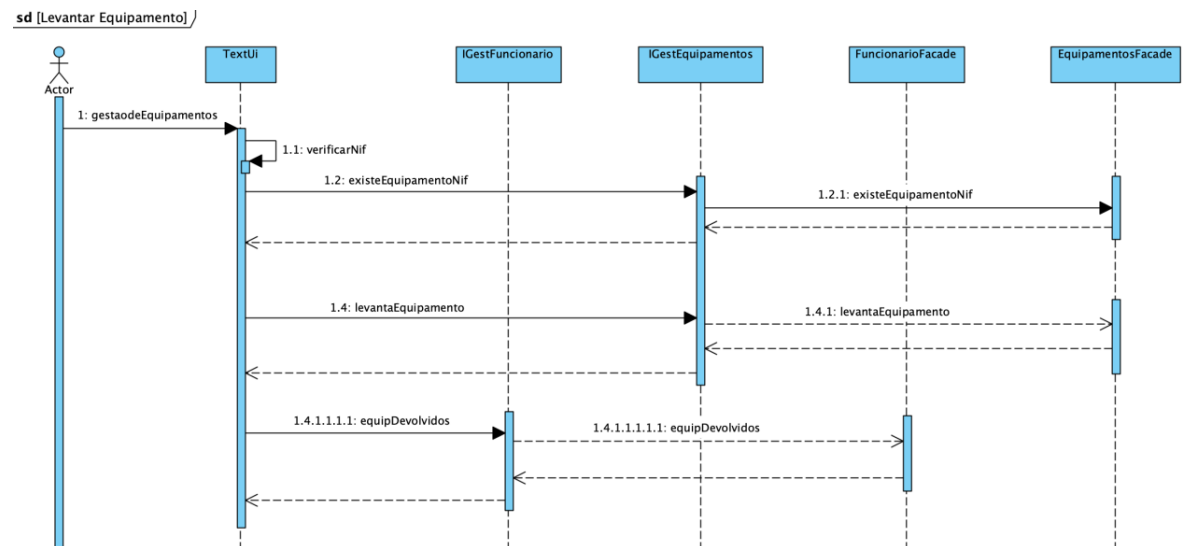


Figura 8.7: gestaoDeEquipamentos

Capítulo 9

Manual de Utilização

Inicialmente, o utilizador encontra um menu principal consoante o tipo de funcionário que é, ou seja, o menu principal será diferente caso seja um técnico, um rececionista ou um gestor.

Assim sendo, se o utilizador for um rececionista ou um técnico, o menu principal permitir-lhe-á aceder às funcionalidades aplicadas aos clientes, aos orçamentos, aos equipamentos e aos serviços. Para além disto, poderá consultar o plano de trabalho e, ainda, alterar o utilizador atual. Na situação do utilizador ser um gestor, para além das funcionalidades referidas anteriormente, também lhe é possibilitado aceder ao relatório de estatísticas.

```
Bem vindo ao Sistema da Loja!

***** UMRepair *****
1 - Cliente
2 - Orçamento
3 - Equipamento
4 - Serviço
5 - Indisponível || Sem credenciais de acesso
6 - Consultar plano de trabalho
7 - Alterar utilizador
*****
0 - Logout || Back
```

Figura 9.1: Menu Principal do Técnico e Rececionista

```
Bem vindo ao Sistema da Loja!

***** UMRepair *****
1 - Cliente
2 - Orçamento
3 - Equipamento
4 - Serviço
5 - Relatório de estatísticas
6 - Consultar plano de trabalho
7 - Alterar utilizador
*****
0 - Logout || Back
```

Figura 9.2: Menu Principal do Gestor

Em seguida, é apresentado o submenu conforme a escolha do utilizador. Se o mesmo, independentemente do seu tipo, escolher a opção 1, ser-lhe-á mostrado o submenu dos clientes, permitindo-o registar, remover e notificar um cliente e consultar a lista de clientes.

```
Opção: 1

***** UMRepair *****
1 - Registrar novo cliente
2 - Remover um cliente
3 - Notificar o cliente
4 - Consultar a lista de clientes
*****
0 - Logout || Back
```

Figura 9.3: Submenu Cliente

Se o usuário, sendo um técnico, escolher a opção 2, ser-lhe-á mostrado o submenu dos orçamentos, possibilitando-lhe editar o orçamento, consultar a lista de orçamentos e apagar um orçamento. Se o usuário for um gestor, este terá apenas a possibilidade de consultar a lista de orçamentos. Finalmente, se o usuário for um rececionista, este pode, ainda, fazer um pedido de orçamento.

```
Opção: 2

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Editar Orçamento
3 - Consultar Lista de Orçamentos
4 - Apagar Orçamento
*****
0 - Logout || Back
```

Figura 9.4:
Submenu
Orçamento
do
Técnico

```
Opção: 2

***** UMRepair *****
1 - Pedido de Orçamento
2 - Indisponível || Sem credenciais de acesso
3 - Consultar Lista de Orçamentos
4 - Indisponível || Sem credenciais de acesso
*****
0 - Logout || Back
```

Figura 9.5:
Submenu
Orçamento
do
Rececionista

```
Opção: 3

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Indisponível || Sem credenciais de acesso
3 - Consultar Lista de Orçamentos
4 - Indisponível || Sem credenciais de acesso
*****
0 - Logout || Back
```

Figura 9.6:
Submenu
Orçamento
do
Gestor

Se o utilizador, sendo um rececionista, escolher a opção 3, ser-lhe-á apresentado o submenu dos equipamentos, permitindo-o registar, levantar e apagar um equipamento, bem como consultar o estado do equipamento, ou seja, se este se encontra reparado, em espera, no armazém ou em processo. Se o utilizador for um técnico ou gestor, apenas lhe é possibilitado levantar e apagar um equipamento, bem como consultar o estado do equipamento.

```
Opção: 3

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Consultar Estado
3 - Indisponível || Sem credenciais de acesso
4 - Apagar Equipamento
*****
0 - Logout || Back
```

Figura 9.7:
Submenu
Equipamento
do
Técnico

```
Opção: 3

***** UMRepair *****
1 - Registar Equipamento
2 - Consultar Estado
3 - Levantar Equipamento
4 - Apagar Equipamento
*****
0 - Logout || Back
```

Figura 9.8:
Submenu
Equipamento
do
Rececionista

```
Opção: 3

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Consultar Estado
3 - Indisponível || Sem credenciais de acesso
4 - Apagar Equipamento
*****
0 - Logout || Back
```

Figura 9.9:
Submenu
Equipamento
do
Gestor

Se o utilizador, sendo um técnico, escolher a opção 4, ser-lhe-á mostrado o submenu dos serviços. Assim, este poderá “definir” o serviço como expresso ou programado, consultar e apagar o serviço e ainda consultar a lista de serviços. No caso do utilizador ser um rececionista ou um gestor, não lhe será autorizado “definir” o serviço como expresso ou programado.

Se o utilizador, sendo um gestor, escolher a opção 5, este terá acesso ao submenu dos relatórios de estatísticas, podendo consultar o relatório de serviços dos técnicos, o relatório detalhado dos técnicos e, por fim, o relatório dos rececionistas.


```
Opção: 1

***** UMRepair *****
1 - Serviço Expresso
2 - Serviço Programado
3 - Consultar Serviço
4 - Consultar Lista de Serviços
5 - Apagar Serviço
*****
0 - Logout || Back
```

Figura 9.10:
Submenu
Serviço
do
Técnico

```
Opção: 1

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Indisponível || Sem credenciais de acesso
3 - Consultar Serviço
4 - Consultar Lista de Serviços
5 - Apagar Serviço
*****
0 - Logout || Back
```

Figura 9.11:
Submenu
Serviço
do
Rececionista

```
Opção: 1

***** UMRepair *****
1 - Indisponível || Sem credenciais de acesso
2 - Indisponível || Sem credenciais de acesso
3 - Consultar Serviço
4 - Consultar Lista de Serviços
5 - Apagar Serviço
*****
0 - Logout || Back
```

Figura 9.12:
Submenu
Serviço
do
Gestor

```
Opção: 3

***** UMRepair *****
1 - Relatório de serviços dos Técnicos
2 - Relatório detalhado dos Técnicos
3 - Relatório de Rececionistas
*****
0 - Logout || Back
```

Figura 9.13: Submenu Relatório de Estatísticas do Gestor

Se o utilizador, independentemente do seu tipo, escolher a opção 6, poderá consultar um plano de trabalho, fornecendo o ID do plano que solicita.

```
Opção: 6

Indique o ID do plano a consultar: 2
ID plano: 2 ID orçamento: 2 Técnico: parkercvst Custo: 90.0 Prazo: PT12M Passos: [Descrição: testar Custo: 90.0 Tempo: PT12M]
```

Figura 9.14: Exemplo de uma consulta de um plano

Se o utilizador, independentemente do seu tipo, escolher a opção 7, ser-lhe-á possibilitado alterar o utilizador atual, fornecendo o username e a respetiva password deste novo utilizador.

```
Opção: 7

Inserir username: filipamurta_10
Inserir password: xxxHappierThanEverxxx

Utilizador alterado.
```

Figura 9.15: Exemplo de uma alteração de utilizador

Finalmente, nos menus, quando é ilustrado a vermelho "Indisponível || Sem credenciais de acesso", isto significa que o utilizador em questão não tem acesso a essa funcionalidade.

Capítulo 10

Análise Crítica

Ao longo da implementação do projeto, o grupo tentou sempre fazer algo eficiente e simples de compreender. O planeamento, análise e conceção adotados foram estabelecidos desde o início, ou seja, o grupo concordou que, antes de tudo, primeiro seriam desenvolvidos os diagramas de modo a se obter uma visão geral e estruturada do trabalho. Em seguida, desenvolveu-se uma solução que, sempre que necessário, seria modificada e melhorada uma vez que, à medida que se ia construindo código, o grupo denotava que certas ideias tomadas previamente poderiam não ser a melhor solução. Assim sendo, algumas destas ideias foram retiradas, outras trabalhadas novamente. Aliás, durante a implementação do código, isto foi comprovado: certas ideias que pensávamos serem funcionais, por vezes, complicavam o descomplicado. Apesar de tudo, esta estratégia mostrou-se consideravelmente trabalhosa, pelo o facto de ser necessário ter sempre em atenção atualizar os diagramas.

Adicionalmente, foi possível conciliar as várias ideias dos elementos do grupo através de reuniões frequentes. Estas reuniões facultaram discussões sobre as abordagens que iriam ser tomadas para cada etapa do projeto, tornando o código acessível face o pretendido.

Nesta segunda fase, o grupo desejava ter consigo implementar a classe `SubPasso` de forma a especificar melhor o **Plano de Trabalho** já que, para além desta, o `SubPasso` influenciava o cálculo do custo e do tempo especificados no orçamento. Contudo, sentimos alguma dificuldade a executar esta parte do projeto. Para concluir, o grupo também não implementou o desvio padrão relativo às estatísticas e sentimos, ainda, que o relatório detalhado dos técnicos poderia apresentar mais informação.

Capítulo 11

Conclusão

A metodologia de **planeamento, análise e conceção** nunca tinha sido antes aplicada por nenhum dos elementos do grupo; porém, mostrou-se bastante útil na medida em que permitiu desenvolver uma solução sólida e fácil de modificar sempre que necessário. Todo este processo antes da implementação demonstrou-se muito vantajoso, pois, caso contrário, se o código fosse implementado sem nenhuma análise prévia do sistema complexo em mãos, encontrar-se-iam imensas incoerências trabalhosas de corrigir. Assim sendo, a partir deste processo, produzimos uma solução sólida, que garante a satisfação dos requisitos do problema.

Concluindo a realização deste projeto, o grupo acredita ter apresentado bons resultados à resolução deste sistema de reparações, devido ao cuidado constante de verificar a coerência do projeto que. Para tal, sempre que se iniciava uma nova fase, reavaliava-se as anteriormente elaboradas.