



TRABALHO PRÁTICO

Desenvolvimento de Sistemas de Software

GRUPO 20



André Nunes A85635



Inês Bastos A89522



Joana Sousa A83614



João Freitas A83782



Tiago Gomes A78141

23 de Dezembro de 2020

Conteúdo

1	Breve Descrição do Enunciado	2
2	Lógica de Negócio	3
2.1	Use Cases	3
2.1.1	Consultar listagem de localizações	3
2.1.2	Registo dos produtos através de QR-code	3
2.1.3	Receção da informação para transportar a palete	4
2.1.4	Notifica que recolheu a palete	4
2.1.5	Notifica que descarregou a palete	4
2.1.6	Iniciar sessão do gestor	5
2.1.7	Terminar Sessão do Gestor	5
2.1.8	Registo do Gestor	5
2.2	Diagrama de Classes	6
2.3	Diagrama de Packages	7
2.4	Diagramas de Sequência	9
2.4.1	Listar a localização das Paletes	9
2.4.2	Iniciar Sessão (Gestor)	10
2.4.3	Terminar Sessão (Gestor)	10
2.4.4	Registar um Gestor	11
2.4.5	Comunicar QR-Code	11
2.4.6	Comunicar ordem de transporte	12
2.4.7	Notificar Recolha	13
2.4.8	Notificar Entrega	14
3	Acesso aos dados	15
3.1	Modelo Lógico	16
4	Interface	17
5	Conclusão	24

Capítulo 1

Breve Descrição do Enunciado

O presente relatório foi realizado no âmbito da terceira fase do projeto da unidade curricular de *Desenvolvimento de Sistemas de Software*, tendo sempre em consideração as etapas realizadas precedentemente. Durante este documento iremos demonstrar o processo, através da interface, de transporte das paletes simulando a comunicação com o leitor e os robots.

Assim, foi necessário **rever** os objetivos desta fase final e **reestruturar** alguns dos nossos **procedimentos anteriormente formulados**.

Para essa finalidade, inicialmente, **o nosso diagrama de classes**, elaborado na fase anterior, **foi reformulado**. Uma vez que um dos objetivos desta fase é o de ter os dados num sistema externo (uma base de dados neste caso), foi necessário **substituir as estruturas de dados convenientes**. Deste modo, foi criado um **modelo lógico da base dados**, que iremos apresentar posteriormente no presente relatório. Também, devido a esta mudança, foi necessário **atualizar os nossos diagramas de sequência** para implementarem as novas estruturas.

Por fim, foi implementada uma ***interface* para um utilizador**, de forma a demonstrar todas as funcionalidades implementadas neste sistema.

Capítulo 2

Lógica de Negócio

2.1 Use Cases

De seguida, iremos confrontar os *Use Cases* implementados no nosso sistema, apresentando a tabela respetiva de cada um, com as devidas atualizações realizadas:

2.1.1 Consultar listagem de localizações

Use Case:	Consultar listagem de localizações
Ator:	Gestor.
Pré-Condição:	Existir Paletes no sistema
Pós-Condição:	Localizações consultadas
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor solicita a listagem.2. Sistema indica a lista das localizações das paletes.

2.1.2 Registo dos produtos através de QR-code

Use Case:	Registo dos produtos através de QR-code.
Ator:	Scanner.
Pré-Condição:	True.
Pós-Condição:	O Sistema fica com o registo dos produtos a partir de QR-code.
Fluxo Normal:	<ol style="list-style-type: none">1. Scanner envia o QR-code.2. Sistema regista os produtos.

2.1.3 Receção da informação para transportar a paleta

Use Case:	Receção da informação para transportar a paleta.
Ator:	Robot.
Pré-Condição:	Existe pelo menos um robot disponível.
Pós-Condição:	O Robot fica com um registo do trajeto a realizar.
Fluxo Normal:	<ol style="list-style-type: none">1. O Sistema verifica a disponibilidade dos robots.2. O Robot é eleito.3. O Sistema calcula o percurso a cumprir.4. O Sistema indica o trajeto da paleta a transportar.
Fluxo Alternativo 1: [Todos os Robots estão ocupados](passo 1)	<ol style="list-style-type: none">1.1 Sistema aguarda a disponibilidade de um Robot.1.2 Retorna ao passo 2.

2.1.4 Notifica que recolheu a paleta

Use Case:	Notifica que recolheu a paleta.
Ator:	Robot
Pré-Condição:	O Robot está em <i>Estado</i> "Recolheu"
Pós-Condição:	Atualização do Estado e Localização do Robot e Paleta
Fluxo Normal:	<ol style="list-style-type: none">1. O Robot avisa o Sistema que recolheu a paleta com sucesso.2. O Sistema recolhe informação da paleta.3. O Sistema altera o estado do Robot.4. O Sistema atualiza a posição do Robot e da paleta.
Fluxo Alternativo 1: [A paleta está no ponto de recolha (tapete rolante)](passo 2)	<ol style="list-style-type: none">2.1 Sistema Remove paleta da lista de paletes no ponto de recolha2.2 Retorna ao passo 3.

2.1.5 Notifica que descarregou a paleta

Use Case:	Notifica que descarregou a paleta.
Ator:	Robot
Pré-Condição:	O Robot está em <i>Estado</i> "Entregou"
Pós-Condição:	Atualização do Estado e Localização do Robot e Paleta
Fluxo Normal:	<ol style="list-style-type: none">1. O Robot avisa o Sistema que descarregou a paleta com sucesso.2. O Sistema recolhe informação da paleta.3. O Sistema altera o estado do Robot.4. O Sistema atualiza a posição do Robot.5. O Sistema atualiza a posição da paleta.
Fluxo Exceção 1: [A paleta é entregue no ponto de entrega (saída do armazém)](passo 5)	<ol style="list-style-type: none">5.1 Sistema Remove paleta do sistema.

2.1.6 Iniciar sessão do gestor

Use Case:	Iniciar sessão do gestor
Ator:	Gestor.
Pré-Condição:	Existirem gestores registados no login
Pós-Condição:	Gestor tem acesso à gestão do armazem
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende iniciar sessão.2. Sistema solicita login e password.3. Gestor insere login e password.4. Sistema valida credenciais.5. Sistema confirma acesso.
Fluxo Exceção 1: [Credenciais não existem no sistema](passo 4)	4.1 Sistema notifica que as credenciais estão erradas.

2.1.7 Terminar Sessão do Gestor

Use Case:	Terminar Sessão.
Ator:	Gestor.
Pré-Condição:	Gestor deve ter sessão iniciada no sistema
Pós-Condição:	Gestor termina sessão
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende terminar sessão.2. Sistema termina a sessão.

2.1.8 Registo do Gestor

Use Case:	Registo.
Ator:	Gestor.
Pré-Condição:	True
Pós-Condição:	Gestor é registado no sistema
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende registar uma nova conta.2. Sistema solicita login, código identificador e password.3. Sistema faz novo registo.
Fluxo Exceção 1: [O login já existe](passo 2)	2.1 Sistema indica que o login já existe.

2.2 Diagrama de Classes

Após a conclusão da implementação do nosso sistema, foi necessário atualizar o nosso diagrama de classes para a lógica de negócio.

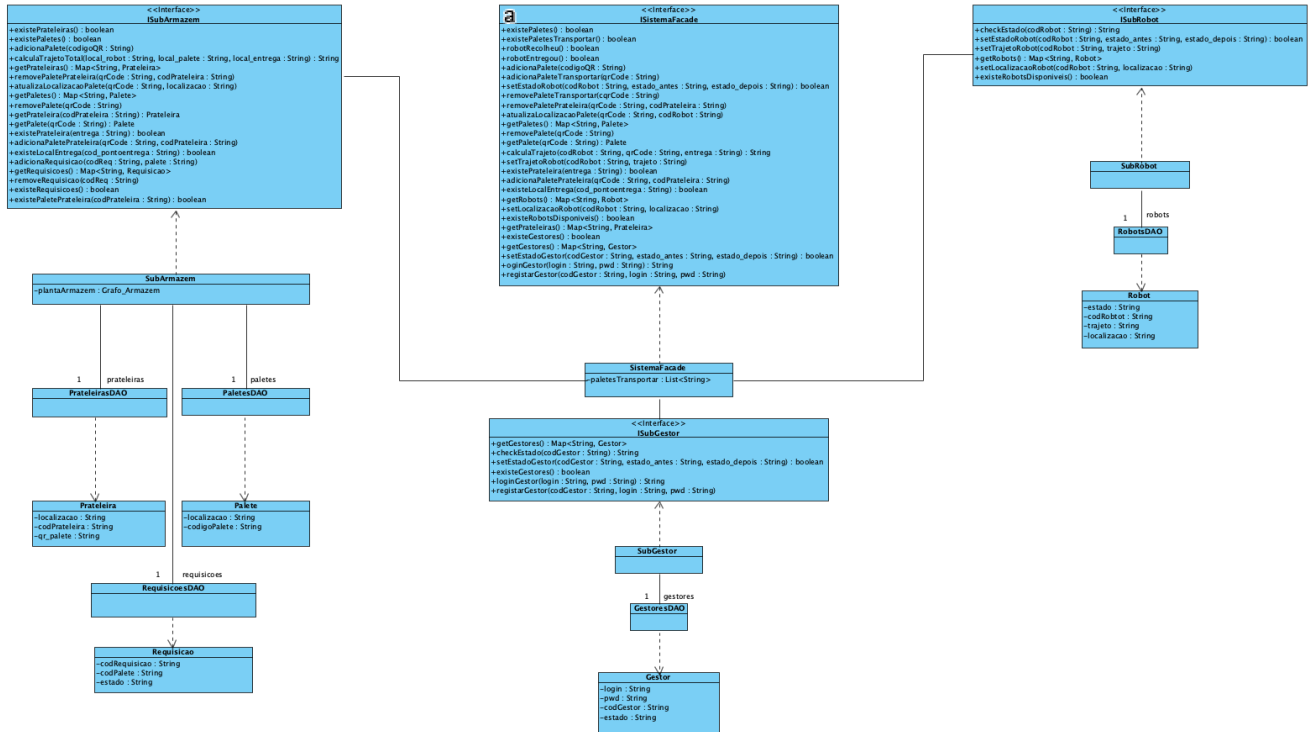


Figura 2.1: Diagrama de Classes

2.3 Diagrama de Packages

Uma vez concluída a implementação do nosso sistema de gestão do armazém em questão, foi necessário reestruturar o nosso **diagrama de packages**, de modo a incluir a interface do utilizador, o acesso aos dados e, também, um *package* auxiliar para a implementação da planta do armazém permitindo, assim, utilizar o algoritmo de **Dijkstra** para calcular o trajeto mais curto dos nossos robots integrantes no sistema.

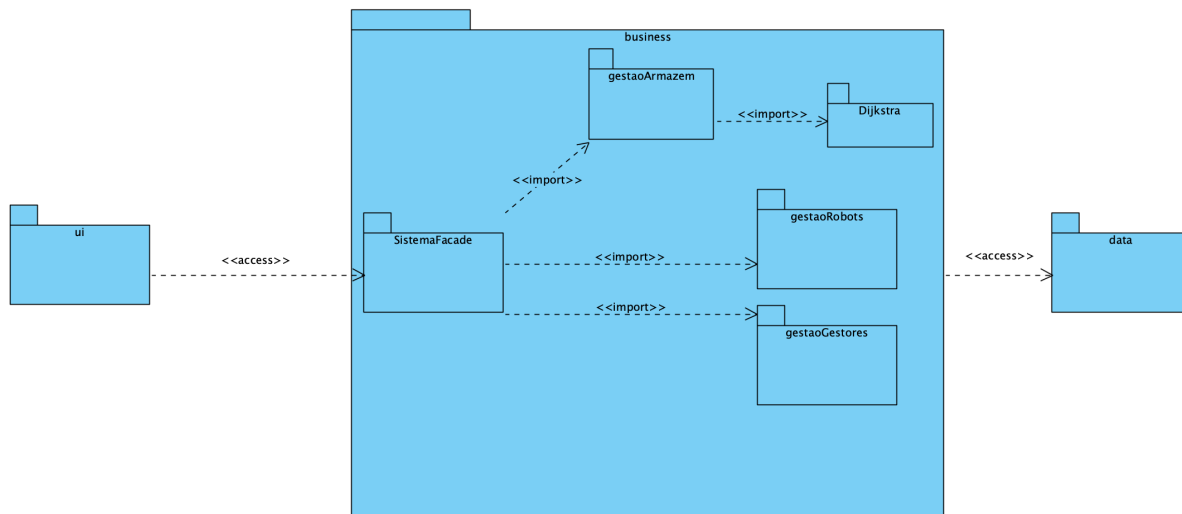


Figura 2.2: Diagrama de Packages

Na nossa camada de lógico, subdividimos em cinco packages mais pequenos.

- O principal papel do package **SistemaFacade** será o de fazer a ligação da interface com a camada lógica.
- Dentro do package **gestaoArmazem** teremos todo o tratamento necessário relativamente à gestão das paletes, prateleiras e requisições no sistema.
- O package **Dijkstra** é um *sub-package* importado pelo **gestaoArmazem**. Neste package é implementado toda a estrutura de dados e métodos necessários para configurar uma planta do armazém (implementado num grafo), tal como calcular o trajeto mais curto entre dois pontos.

Para a estrutura do armazém produzida nesta fase, utilizando este package, foi empregue um que define a posição dos corredores, tal como os respetivos tamanhos, e cruzamentos.

Nota: Para descarregar num determinado corredor, considerou-se que existe apenas um ponto de descarga nesse corredor, existindo o número de prateleiras que se definam. Isto permite uma **solução mais flexível** para ser **possível definir quantas prateleiras se deseja colocar em cada corredor**. Assim, apresentamos a seguinte representação do grafo:

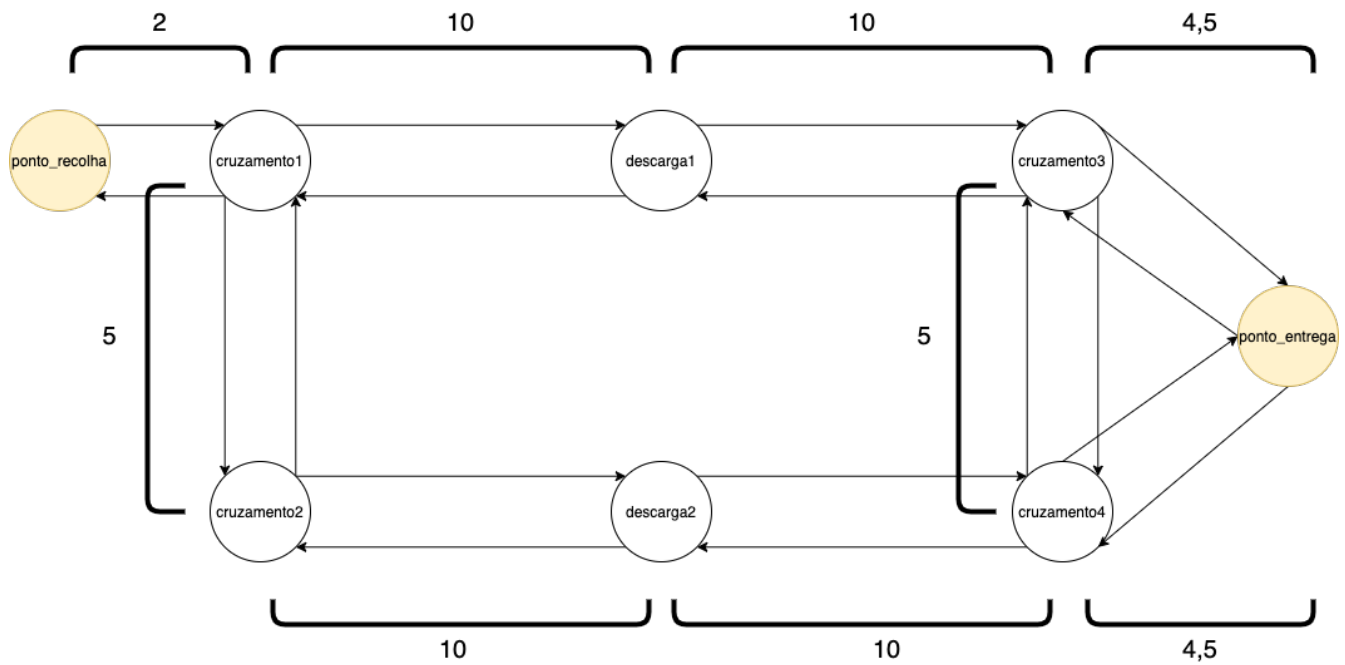


Figura 2.3: Grafo da planta do armazém

Como podemos observar, foi necessário criar um grafo orientado, definindo todos os corredores com as respectivas direções.

- Dentro do package **gestaoRobots** teremos todo o tratamento necessário relativo à gestão de todos os robots integrantes no armazém.
- Dentro do package **gestaoGestores** teremos todo o tratamento necessário relativo à gestão de todos os gestores integrantes no armazém.

A nossa **camada de acesso aos dados** integra todas as classes e métodos necessários para o acesso aos dados de um sistema externo que contém a informação do mesmo. A nossa **camada de interface de utilizador** integra todas as classes e métodos necessários para que seja possível um utilizador obter toda a informação implementada na nossa lógica de negócio.

2.4 Diagramas de Sequência

Com o intuito de melhor explicitar o desenvolvimento deste *sistema de software*, foram realizados novos **diagramas de sequência**. Cada diagrama expõe o processo tomado pelo grupo em cada *Use Case* anteriormente definido.

2.4.1 Listar a localização das Paletes

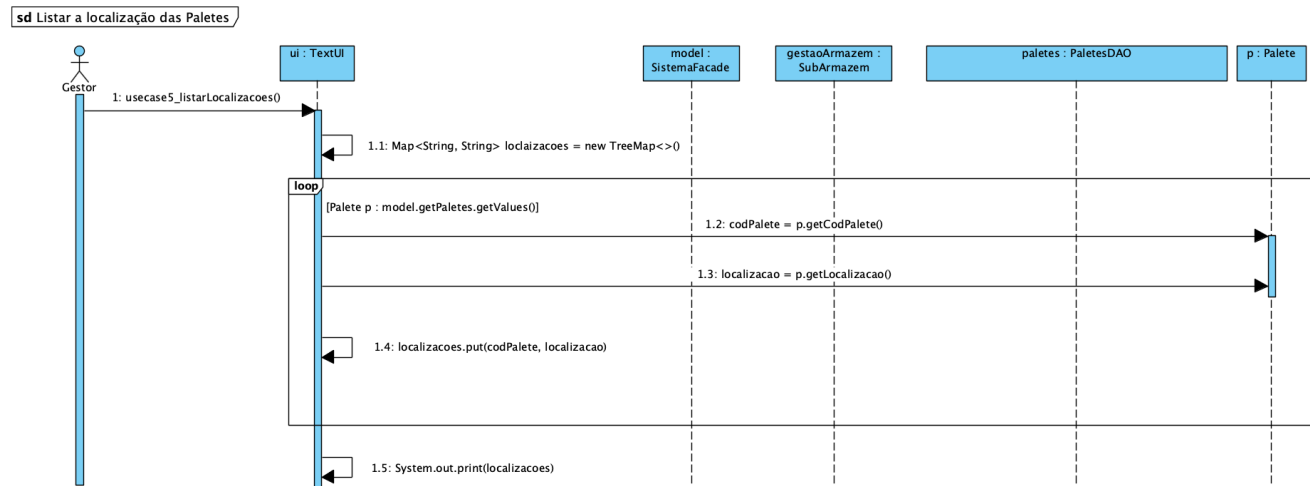


Figura 2.4: Diagrama da listagem das localizações das Paletes

2.4.2 Iniciar Sessão (Gestor)

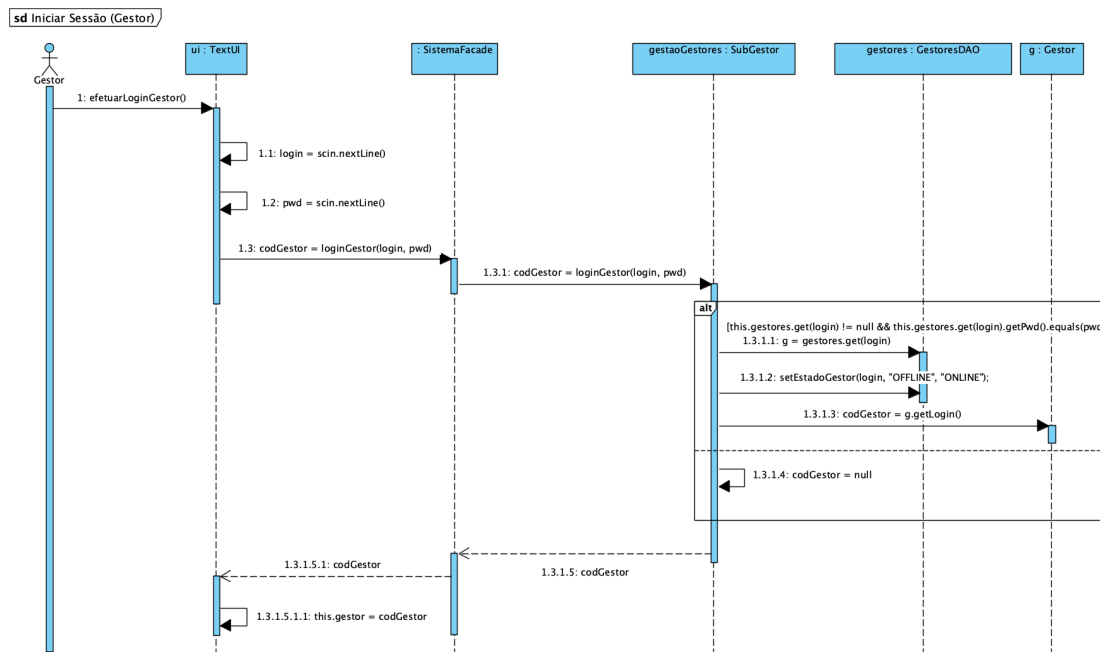


Figura 2.5: Diagrama de iniciar sessão numa conta existente de um Gestor

2.4.3 Terminar Sessão (Gestor)

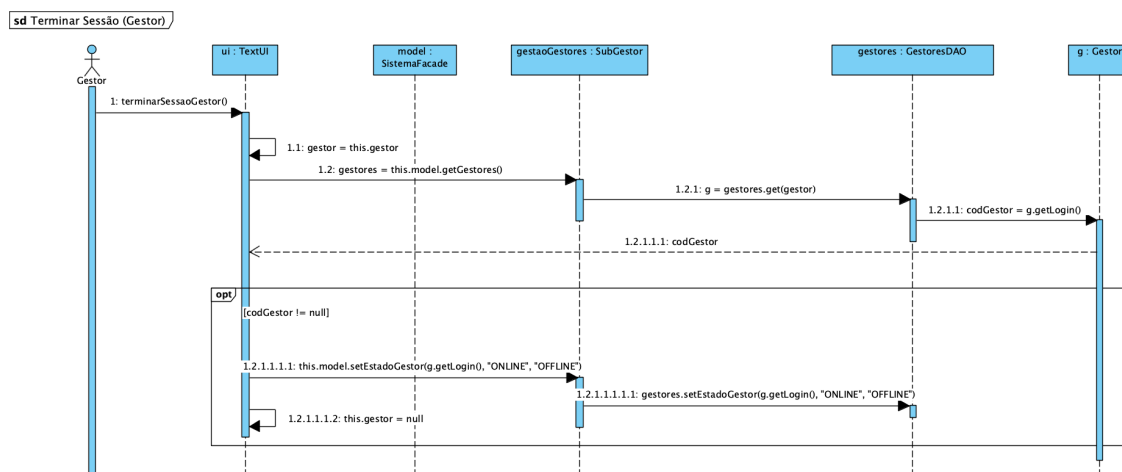


Figura 2.6: Diagrama de terminar sessão numa conta de Gestor "online"

2.4.4 Registrar um Gestor

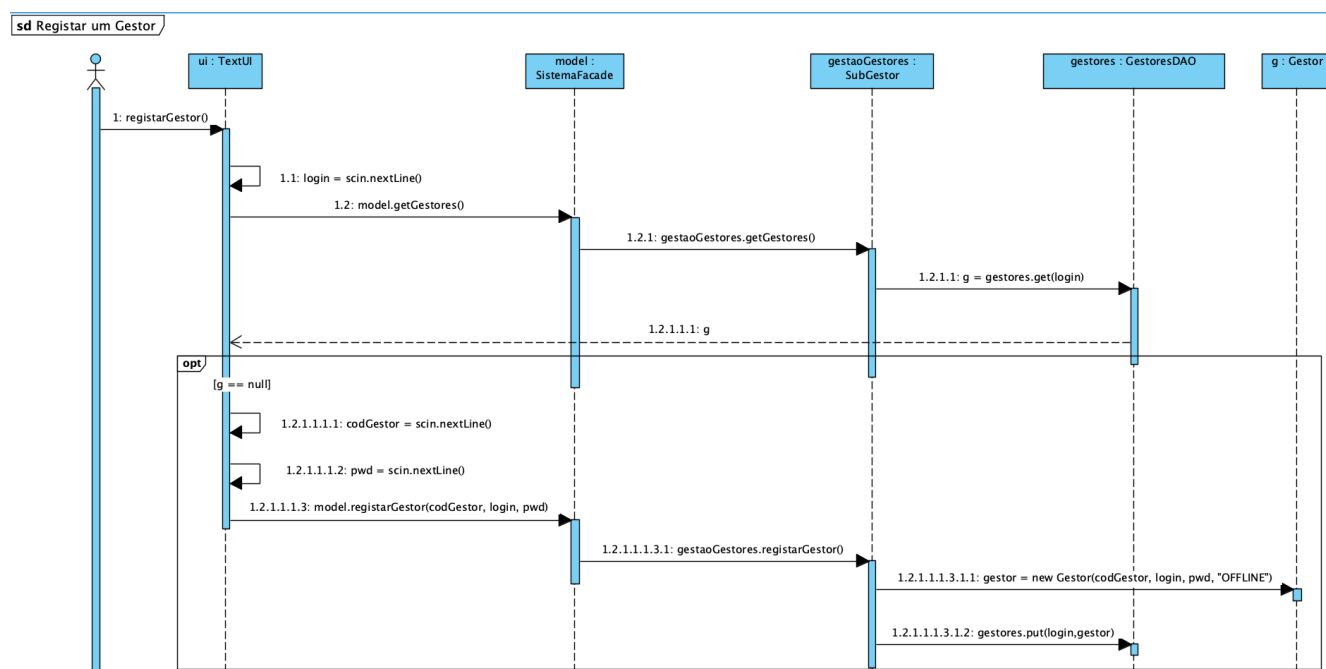


Figura 2.7: Diagrama do processo de Registrar uma nova conta de Gestor

2.4.5 Comunicar QR-Code

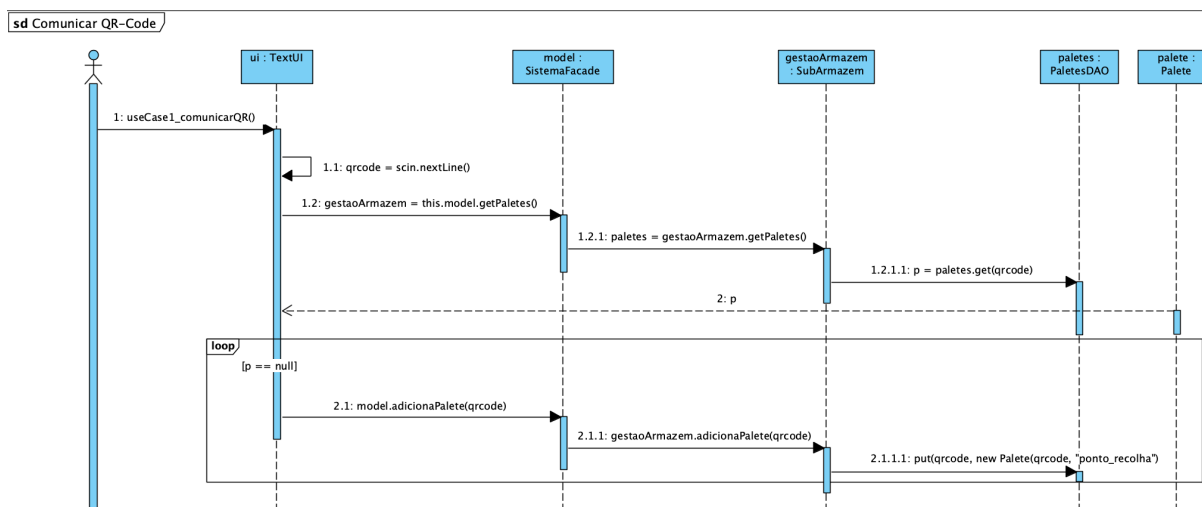


Figura 2.8: Diagrama do registo de paletes no sistema

2.4.6 Comunicar ordem de transporte

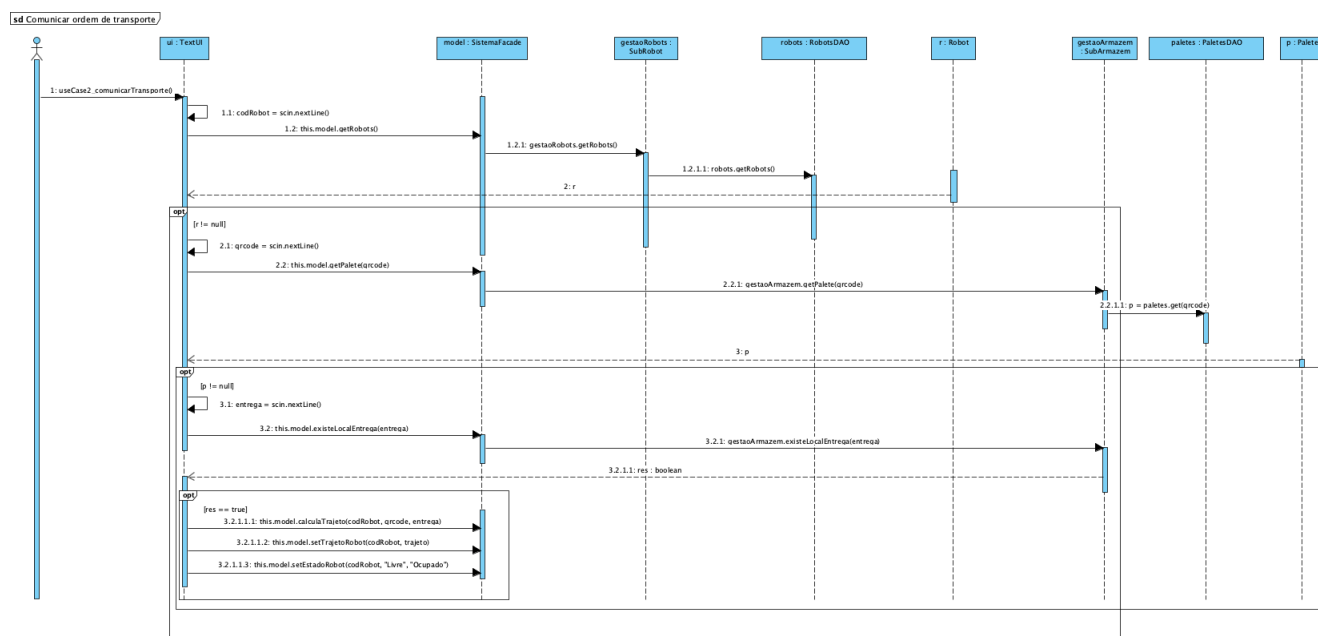


Figura 2.9: Diagrama de calcular e enviar um trajeto a um Robot

Nota: Neste diagrama não está exposto o processo dos últimos métodos chamados ao *Sistema-Facade* devido à extensão deste *Use Case*. Porém, o respetivo nome do método é auto-explicativo do processo que realiza.

2.4.7 Notificar Recolha

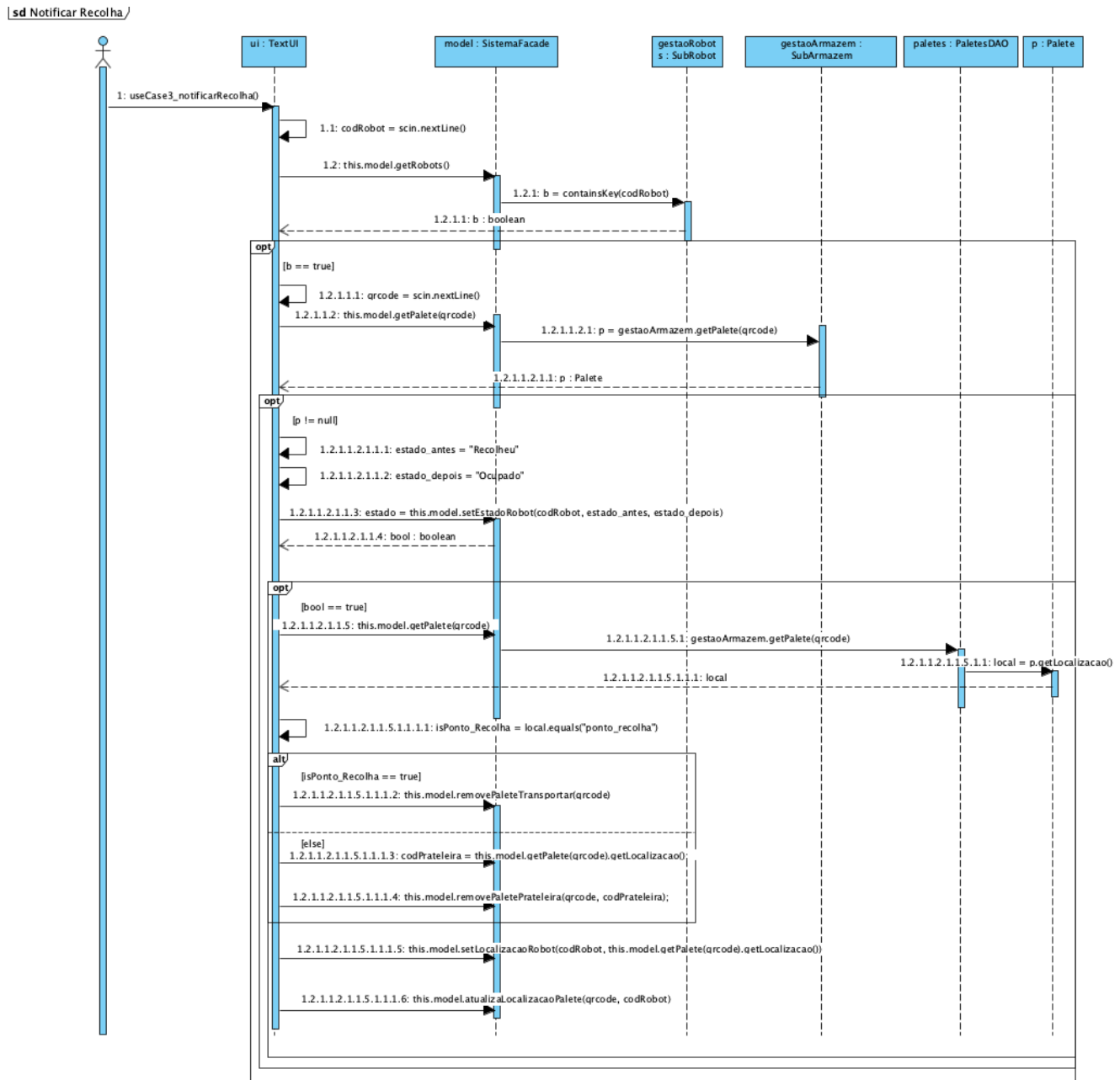


Figura 2.10: Diagrama do processo de notificação de uma recolha de paleta

Nota: Neste diagrama não está exposto o processo dos métodos mais complexos chamados ao *SistemaFacade* devido à extensão deste *Use Case*. Porém, o respetivo nome do método é auto-explicativo do processo que realiza.

2.4.8 Notificar Entrega

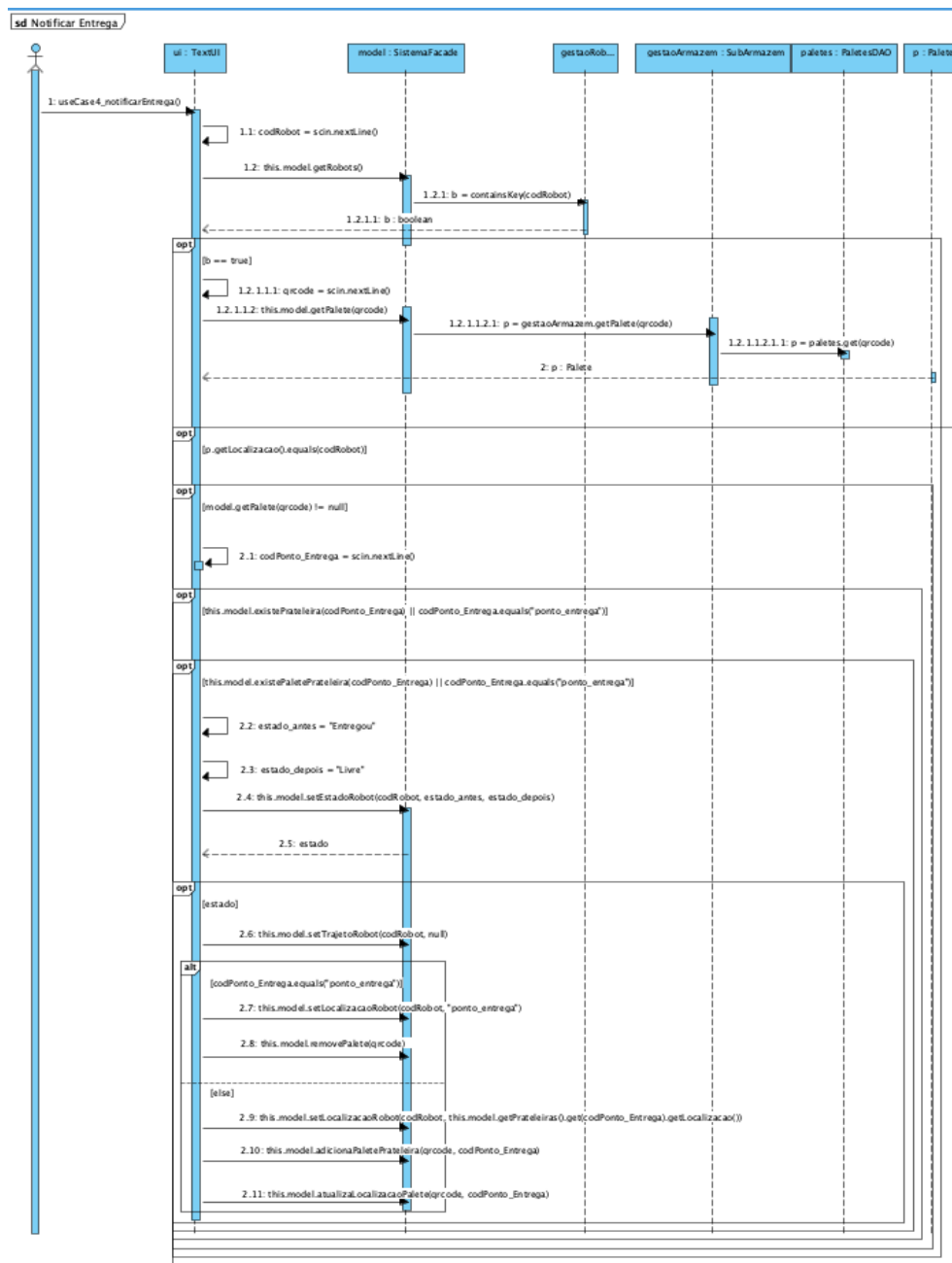


Figura 2.11: Diagrama do processo de notificação de uma entrega de paleta

Nota: Neste diagrama não está exposto o processo dos métodos mais complexos chamados ao *SistemaFacade* devido à extensão deste *Use Case*. Porém, o respetivo nome do método é auto-explicativo do processo que realiza.

Capítulo 3

Acesso aos dados

De forma a implementar a lógica de dados do sistema, foi criada uma base de dados com o nome *gestaoArmazem*. Como se trata de um sistema externo ao da lógica de negócio, foi necessário criar classes do tipo **DAO** (*Data Access Object*). Estas classes acedem diretamente à base de dados para executar os comandos necessários e, consequentemente, aceder à informação.

```
CREATE database gestaoArmazem;  
Use gestaoArmazem;  
DROP database gestaoArmazem;  
  
public class DAOconfig {  
    static final String USERNAME = "----"; // Actualizar  
    static final String PASSWORD = "----"; // Actualizar  
    private static final String DATABASE = "gestaoArmazem"; // Actualizar  
    // private static final String DRIVER = "jdbc:mariadb"; // Usar para MariaDB  
    private static final String DRIVER = "jdbc:mysql"; // Usar para MySQL  
    static final String URL = DRIVER + "://localhost:3306/" + DATABASE;  
}
```

Figura 3.1: Configuração dos DAO's e criação da base de dados no *software MySQL Workbench*

Para a implementação deste nosso sistema, foi utilizado o *software MySQL Workbench*, de modo a ser possível testar o nosso sistema. Na entrega desta fase, enviamos, também, um ficheiro do tipo *.sql* com todas as *queries* para uma verificação cómoda da informação presente na base de dados, permitindo o teste dos *DAO's* implementados.

3.1 Modelo Lógico

Como o nosso sistema foi implementado em linguagem **JAVA** e o objetivo dos *DAO's* ser o de aceder aos dados da BD criada, foi necessário criar um modelo lógico que é criado quando os *DAO's* acedem ao sistema externo.

Assim, de seguida demonstramos a nossa implementação do nosso modelo lógico, em linguagem **JAVA** e **SQL**:

```
private RobotsDAO() {
    try (Connection conn = DriverManager.getConnection(DAOconfig.URL, DAOconfig.USERNAME, DAOconfig.PASSWORD);
        Statement stm = conn.createStatement()) {

        String sql = "CREATE TABLE IF NOT EXISTS Robots (" + "CodRobot varchar(45) NOT NULL PRIMARY KEY,"
            + "Estado varchar(45) DEFAULT NULL," + "Trajeto varchar(" + max_char + ") DEFAULT NULL,"
            + "Localizacao varchar(45) DEFAULT NULL)";
        stm.executeUpdate(sql);
    } catch (SQLException e) {
        // Erro a criar tabela...
        e.printStackTrace();
        throw new NullPointerException(e.getMessage());
    }
}

private GestoresDAO() {
    try (Connection conn = DriverManager.getConnection(DAOconfig.URL, DAOconfig.USERNAME, DAOconfig.PASSWORD);
        Statement stm = conn.createStatement()) {

        // trajeto é string de 500 chars, aumentar se armazém for de grande dimensão
        String sql = "CREATE TABLE IF NOT EXISTS Gestores (" + "CodGestor varchar(45) NOT NULL PRIMARY KEY,"
            + "Login varchar(45) DEFAULT NULL," + "Pwd varchar(45) DEFAULT NULL,"
            + "Estado varchar(45) DEFAULT NULL)";
        stm.executeUpdate(sql);
    } catch (SQLException e) {
        // Erro a criar tabela...
        e.printStackTrace();
        throw new NullPointerException(e.getMessage());
    }
}

private PaletesDAO() {
    try (Connection conn = DriverManager.getConnection(DAOconfig.URL, DAOconfig.USERNAME, DAOconfig.PASSWORD);
        Statement stm = conn.createStatement()) {

        String sql = "CREATE TABLE IF NOT EXISTS paletes (" + "CodPaleta varchar(45) NOT NULL PRIMARY KEY,"
            + "Localizacao varchar(45) DEFAULT NULL)";
        stm.executeUpdate(sql);
    } catch (SQLException e) {
        // Erro a criar tabela...
        e.printStackTrace();
        throw new NullPointerException(e.getMessage());
    }
}

private PrateleirasDAO() {
    try (Connection conn = DriverManager.getConnection(DAOconfig.URL, DAOconfig.USERNAME, DAOconfig.PASSWORD);
        Statement stm = conn.createStatement()) {

        String sql = "CREATE TABLE IF NOT EXISTS prateleiras (" + "CodPrateleira varchar(45) NOT NULL PRIMARY KEY,"
            + "Localizacao varchar(45) DEFAULT NULL," + "CodPaleta varchar(45) DEFAULT NULL)";
        // + "codPaleta varchar(45) DEFAULT NULL)";
        stm.executeUpdate(sql);
    } catch (SQLException e) {
        // Erro a criar tabela...
        e.printStackTrace();
        throw new NullPointerException(e.getMessage());
    }
}
```

Figura 3.2: Criação do modelo lógico

Capítulo 4

Interface

Quando introduzimos o programa, é-nos apresentado o Menu Inicial com uma mensagem de Boas Vindas ao utilizador. Inicialmente, o utilizador poderá escolher entre uma das três opções 1-Gestor-, 2-Requisições-, 0-Sair-. Selecionada a opção 1, o Gestor tem a hipótese de se registar no sistema ou, se já estivesse registado anteriormente, poderá efetuar o seu login (tendo de inserir a sua password). Além disso, existe ainda mais uma opção referente ao Gestor terminar a sua sessão. Porém, essa seleção não é exibida na imagem abaixo pois, para isso, o login terá de ser efetuado. Logo após o gestor ter inserido os seus dados, poderá selecionar o primeiro Use Case referente a comunicar códigos QR que serão explicados posteriormente.

```
Bem vindo ao Sistema de Gestão do Armazém!

GRUPO 20

>>>>>>>| André Nunes - A85635 |<<<<<<<<
>>>>>>>| Inês Bastos - A89522 |<<<<<<<<
>>>>>>>| Joana Sousa - A83614 |<<<<<<<<
>>>>>>>| João André - A83782 |<<<<<<<<
>>>>>>>| Tiago Gomes - A78141 |<<<<<<<<

1ºSemestre DSS
ANO LETIVO 2020/2021

Menu
● 1 - OPÇÃO: Gestor
● 2 - OPÇÃO: Requisições
● 0 - Sair
>>> Opção: 1

Menu
● 1 - GESTOR: Efetuar Login
● 2 - ---
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 1
Inserir Login:
gestor1
Inserir Password:
pass1

SEJA BEM-VINDO <gestor1>

Menu
● 1 - Use Case 1 - Comunicar código QR
● 2 - ---
● 3 - ---
● 4 - ---
● 5 - ---
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 
```

Figura 4.1: Menu do Gestor

Como podemos observar pela imagem, depois do gestor ter efetuado o seu login, é apresentado o primeiro Use Case, em que este trata de comunicar o código QR, ou seja, temos de colocar o código QR da paleta desejada e, posteriormente, este código será adicionado à base de dados e notificado que está no local de recolha.

Depois de finalizado o primeiro use case, é-nos apresentados mais um, no qual se comunica a ordem de transporte. Este resulta de inserirmos o código de robot associado, o código QR da paleta e o local de entrega desejado. Posteriormente, consoante estes dados, temos a informação do caminho mais curto.

```

Menu
1 - Use Case 1 - Comunicar código QR
2 - ---
3 - ---
4 - ---
5 - ---
6 - Estado Robot: Recolheu
7 - Estado Robot: Entregou
8 - SIMULAÇÃO DO PROCESSO
0 - Sair
>>> Opção: 1
Inserir código QR:
paleta1
Paleta com QR-Code paleta1 adicionada à base de dados! Está no local de recolha!

Menu
1 - Use Case 1 - Comunicar código QR
2 - Use Case 2 - Comunicar ordem de transporte
3 - ---
4 - ---
5 - Use Case 5 - Consultar listagem de localizações
6 - Estado Robot: Recolheu
7 - Estado Robot: Entregou
8 - SIMULAÇÃO DO PROCESSO
0 - Sair
>>> Opção: 2
Inserir código do Robot:
RobotCode_1
Inserir código QR:
paleta1
Inserir local de entrega:
descarga2

Não existe um caminho entre ponto_recolha e ponto_recolha!

O caminho mais curto entre ponto_recolha e descarga2 é:
ponto_recolha cruzamento1 cruzamento2 descarga2

O trajeto custa: 17.0
---->Trajeto calculado e enviado para o robot: ponto_recolha cruzamento1 cruzamento2 descarga2

Menu
1 - Use Case 1 - Comunicar código QR
2 - Use Case 2 - Comunicar ordem de transporte
3 - ---
4 - ---
5 - Use Case 5 - Consultar listagem de localizações
6 - Estado Robot: Recolheu
7 - Estado Robot: Entregou
8 - SIMULAÇÃO DO PROCESSO
0 - Sair
>>> Opção: 
```

Figura 4.2: Use Case 1 e 2

Para utilizar o Use Case 3 - Notificar recolha de paletes - o estado do robot terá que ser "Recolheu". Ao escolher esta opção (6), o gestor deverá inserir o código do robot associado. Posteriormente ao estado do robot ter sido alterado poderemos então notificar a recolha de paletes, seleccionando a opção três. Para isso, precisamos introduzir o código do robot e o código QR da paleta a que nos queremos referir. No final desta execução o sistema informa o gestor que o robot seleccionado recolheu a paleta pretendida.

```

███ Menu ███
● 1 - Use Case 1 - Comunicar código QR
● 2 - Use Case 2 - Comunicar ordem de transporte
● 3 - ---
● 4 - ---
● 5 - Use Case 5 - Consultar listagem de localizações
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 6
Inserir código do Robot:
RobotCode_1

███ Menu ███
● 1 - Use Case 1 - Comunicar código QR
● 2 - Use Case 2 - Comunicar ordem de transporte
● 3 - Use Case 3 - Notificar recolha de paletes
● 4 - ---
● 5 - Use Case 5 - Consultar listagem de localizações
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 3
Inserir código do Robot:
RobotCode_1
Código QR da paleta:
paleta1
RobotCode_1 recolheu!
```

Figura 4.3: Use Case 3

Após a utilização do Use Case 3, ou seja, seguido a notificar a recolha das paletes precisamos de proceder para a entrega das mesmas. Para isso, o estado do robot precisa de ser alterado para "Entregou", de forma a informar o sistema de que avançou para a entrega das paletes, precisando para isso, de inserir o código do robot escolhido.

Após esta inserção procedemos a seleção da opção 4, identificando o código do robot, o código QR da palete e ainda o local de entrega planeado. É ainda importante salientar que caso a palete seja depositada num local de entrega, automaticamente será removida da base de dados. Tal como acima referido, o sistema informa o gestor que o robot seleccionado entregou a palete pretendida.

```

███ Menu ███
● 1 - Use Case 1 - Comunicar código QR
● 2 - ---
● 3 - ---
● 4 - ---
● 5 - Use Case 5 - Consultar listagem de localizações
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 7
Inserir código do Robot:
RobotCode_1

███ Menu ███
● 1 - Use Case 1 - Comunicar código QR
● 2 - ---
● 3 - ---
● 4 - Use Case 4 - Notificar entrega de paletes
● 5 - Use Case 5 - Consultar listagem de localizações
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 4
Inserir código do Robot:
RobotCode_1
Código QR da palete:
palete1
Local de Entrega:
PrateleiraDescarga1_3
RobotCode_1 entregou!
```

Figura 4.4: Use Case 4

Após os use cases de notificar recolha e entrega de paletes, podemos, como desde início, consultar a listagem de localizações que é referente ao use case 5. Ao eleger esta opção, o sistema revela as diferentes paletes guardadas na base de dados e as suas localizações.

```

Menu
● 1 - Use Case 1 - Comunicar código QR
● 2 - Use Case 2 - Comunicar ordem de transporte
● 3 - ---
● 4 - ---
● 5 - Use Case 5 - Consultar listagem de localizações
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 5
{palete1=ponto_recolha, palete11=RobotCode_1, palete2=ponto_recolha, palete3=ponto_recolha, palete33=RobotCode_3, palete4=ponto_recolha}

```

Figura 4.5: Use Case 5

No caso de o Gestor ter selecionado a hipótese de registar uma nova conta de acesso ao sistema, é necessário que indique os dados a inserir na nova conta de acesso ao sistema de base de dados e, em caso de sucesso, o sistema indica os dados da conta registada. Na eventualidade de inserir um login já existente, é notório salientar que não será possível fazer o registo dessa conta.

```

Menu
● 1 - GESTOR: Efetuar Login
● 2 - ---
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 3
Inserir o seu login:
grupo20
Inserir o seu código de Gestor:
grupo20
Inserir a sua password:
grupo20

>>>>>> Código de Gestor <grupo20>
>>>>>> Login: <grupo20>
>>>>>> Password: <grupo20>

Menu
● 1 - GESTOR: Efetuar Login
● 2 - ---
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 

```

Figura 4.6: Registo Gestor

Como já referido anteriormente, o Gestor tem a opção de terminar sessão, seleccionando a opção 2. Assumindo que ele efetuou todas as funcionalidades pretendidas, o Sistema mostra uma mensagem de despedida voltando ao menu do Gestor. Neste ponto, conseguimos reparar que, como não existe qualquer login associado, esta opção deixou de ser possível de ser seleccionada.

```

Menu
● 1 - GESTOR: Efetuar Login
● 2 - ---
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 1
Inserir Login:
gestor1
Inserir Password:
pass1

SEJA BEM-VINDO <gestor1>

Menu
● 1 - Use Case 1 - Comunicar código QR
● 2 - ---
● 3 - ---
● 4 - ---
● 5 - ---
● 6 - Estado Robot: Recolheu
● 7 - Estado Robot: Entregou
● 8 - SIMULAÇÃO DO PROCESSO
● 0 - Sair
>>> Opção: 0

Menu
● 1 - GESTOR: Efetuar Login
● 2 - GESTOR: Terminar Sessão
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 2

Até à próxima <gestor1>

Menu
● 1 - GESTOR: Efetuar Login
● 2 - ---
● 3 - GESTOR: Registrar
● 0 - Sair
>>> Opção: 
```

Figura 4.7: Terminar Sessão Gestor

Voltando ao menu inicial e escolhendo a opção 2, dirigimo-nos ao menu das requisições. Este menu tem apenas duas funcionalidades como registrar e remover uma requisição. Selecionando a opção de Registo deve ser inserido, tanto o código da requisição como o código QR de palete, porém para remover uma requisição é apenas necessário o código da requisição. É importante referir que o sistema não aceita a remoção de uma requisição sem um registo inicial.

```

███ ██████████ Menu ██████████
● 1 - OPÇÃO: Gestor
● 2 - OPÇÃO: Requisições
● 0 - Sair
>>> Opção: 2

███ ██████████ Menu ██████████
● 1 - REQUISIÇÃO: Registrar uma requisição
● 2 - ---
● 0 - Sair
>>> Opção: 1
Inserir Código da Requisição:
Requisicao_1
Inserir Código QR da Palete:
palete1

Requisição registrada no sistema!

███ ██████████ Menu ██████████
● 1 - REQUISIÇÃO: Registrar uma requisição
● 2 - REQUISIÇÃO: Remover uma requisição
● 0 - Sair
>>> Opção: 2
Inserir Código da Requisição:
Requisicao_1

Requisição removida do sistema!

███ ██████████ Menu ██████████
● 1 - REQUISIÇÃO: Registrar uma requisição
● 2 - ---
● 0 - Sair
>>> Opção: █
```

Figura 4.8: Funcionalidades de Requisição

Capítulo 5

Conclusão

Nesta terceira fase do trabalho e dando seguimento ao que foi realizado na primeira e segunda fase, continuamos com as etapas sequenciais da linguagem UML, mais especificamente, os diagramas sequenciais sobre os Use Cases fornecidos pelos docentes.

Esta terceira fase teve como principal objetivo a demonstração do processo de transporte das paletes isto é, desde que o QR code é lido até que as paletes são colocadas nas prateleiras. Com isto, no nosso trabalho é possível criar e cancelar registos de **pedidos de requisição** mas, no entanto, **não foi implementado o tratamento das mesmas**. Ainda assim, tivemos que implementar um diagrama de classes onde são colocados todas as operações necessárias, tais como os relacionamentos entre o sistema, as interfaces dos subsistemas, e os próprios subsistemas.

No seguimento do diagrama de classes, também foi atualizado o diagrama de packages, de modo a complementar a fase pretendida.

Para finalizar, apresentou-se o modelo lógico da nossa lógica de acesso aos dados, onde demonstramos como é que a informação é depositada num sistema externo de base de dados.

Posto isto, demos então por terminada esta terceira fase do trabalho prático da Unidade Curricular de Desenvolvimento de Sistemas de Software.