



TRABALHO PRÁTICO

Desenvolvimento de Sistemas de Software

GRUPO 20



André Nunes A85635



Inês Bastos A89522



Joana Sousa A83614



João Freitass A83782



Tiago Gomes A78141

28 de Novembro de 2020

Conteúdo

1	Breve Descrição do Enunciado	2
1.1	Modelo de Dominio	2
1.2	Modelo de Use Case	3
1.2.1	Consultar listagem de localizações	3
1.2.2	Iniciar Sessão do Gestor	4
1.2.3	Terminar Sessão do Gestor	4
1.2.4	Registo do Gestor	4
1.2.5	Iniciar Sessão do Encarregado	5
1.2.6	Terminar Sessão do Encarregado	5
2	Conclusão	19

Capítulo 1

Breve Descrição do Enunciado

O presente relatório foi realizado no âmbito da segunda fase do projeto da unidade curricular de Desenvolvimento de Sistemas de Software. Após a entrega da primeira fase, foi-nos proposta a criação de uma arquitetura conceptual do sistema, capaz de suportar o conjunto de Use Cases definidos na submissão anterior do mesmo trabalho, conjuntamente com os Use Cases definidos pela equipa docente.

Para esta segunda fase, de forma a atingir o objetivo pretendido, começamos por analisar os Use Cases de forma a registar todos os métodos necessários ao funcionamento proposto. Seguidamente, procedemos à construção dos Diagramas de Sequência para cada responsabilidade do sistema que assim o exigisse, a formulação do Diagrama de Classe, o Diagrama de Componentes e, por fim, o Diagrama de Packages.

1.1 Modelo de Dominio

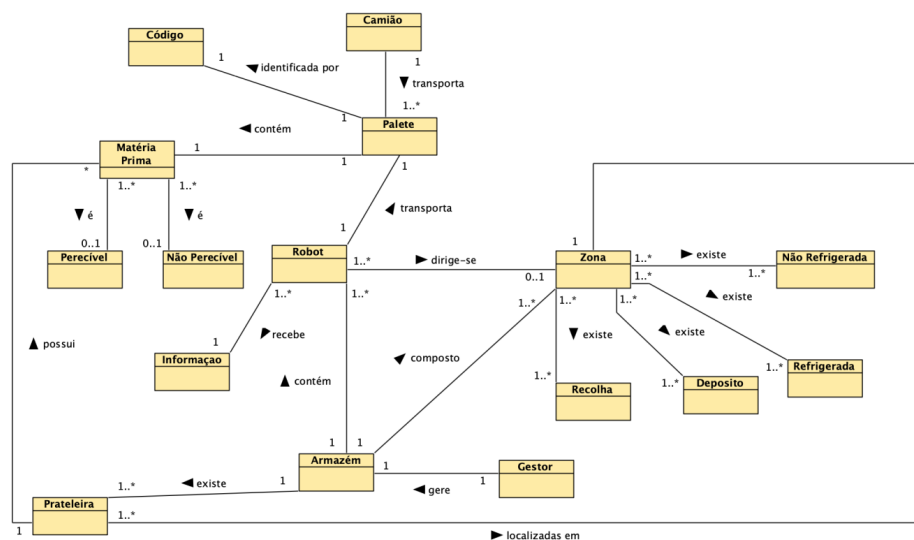


Figura 1.1: Modelo de Dominio

1.2 Modelo de Use Case

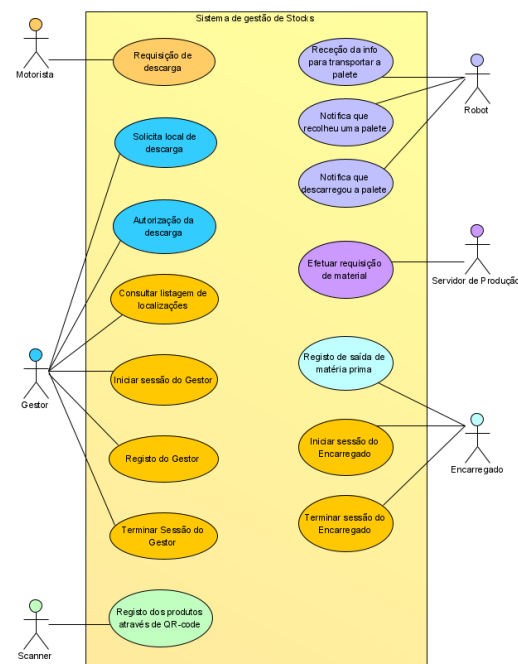


Figura 1.2: Modelo de Use Case

Ao nos confrontarmos com a 2ª Fase deste projeto, achamos, por bem, alterar alguns Use Cases usados na Fase 1. Adicionamos 5 *Use Cases* fundamentais nesta etapa como o *Iniciar Sessão* e *Terminar Sessão*, tanto para o Gestor, como para o Encarregado e ainda *Consultar listagem de localizações* no Ator Gestor e, por fim o *Registo do Gestor*. Estes novos Use Cases estão identificados na imagem acima com a cor amarela para uma melhor compreensão.

A análise é demonstrada em seguida:

1.2.1 Consultar listagem de localizações

Use Case:	Consultar listagem de localizações
Ator:	Gestor.
Pré-Condição:	True
Pós-Condição:	Localizacoes consultadas
Fluxo Normal:	<div>1. Gestor solicita a listagem.</div> <div>2. Sistema indica a lista das localizações das paletes.</div>

1.2.2 Iniciar Sessão do Gestor

Use Case:	Iniciar Sessão.
Ator:	Gestor.
Pré-Condição:	Pretende iniciar sessão.
Pós-Condição:	Sessão iniciada.
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende iniciar sessão.2. Sistema solicita email e password.3. Gestor insere email e password.4. Sistema verifica credenciais.5. Sistema valida credenciais.6. Sistema confirma acesso.
Fluxo Exceção 1: [Gestor não autoriza a descarga](passo 4)	4.1 Sistema informa que as credenciais não são válidas

1.2.3 Terminar Sessão do Gestor

Use Case:	Terminar Sessão.
Ator:	Gestor.
Pré-Condição:	Pretende terminar sessão.
Pós-Condição:	Sessão Terminada.
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende terminar sessão2. Sistema termina a sessão

1.2.4 Registo do Gestor

Use Case:	Registo.
Ator:	Gestor.
Pré-Condição:	Pretende registar-se.
Pós-Condição:	Novo Gestor registado.
Fluxo Normal:	<ol style="list-style-type: none">1. Gestor indica que pretende registar uma nova conta.2. Sistema solicita email e password.3. Sistema verifica se as informações já existem.4. Sistema faz novo registo.

1.2.5 Iniciar Sessão do Encarregado

Use Case:	Iniciar Sessão.
Ator:	Encarregado.
Pré-Condição:	Pretende iniciar Sessão.
Pós-Condição:	Sessão Iniciada.
Fluxo Normal:	<ol style="list-style-type: none">1. Encarregado indica que pretende iniciar sessão.2. Sistema solicita email e password.3. Encarregado insere email e password.4. Sistema verifica credenciais.5. Sistema valida credenciais.6. Sistema confirma acesso.
Fluxo Exceção 1: [Encarregado não autoriza a descarga](passo 4)	4.1 Sistema informa que as credenciais não são válidas

1.2.6 Terminar Sessão do Encarregado

Use Case:	Terminar Sessão.
Ator:	Encarregado.
Pré-Condição:	Pretende terminar sessão.
Pós-Condição:	Sessão Terminada.
Fluxo Normal:	<ol style="list-style-type: none">1. Encarregado indica que pretende terminar sessão2. Sistema termina a sessão

Lista Localizacao Paletes

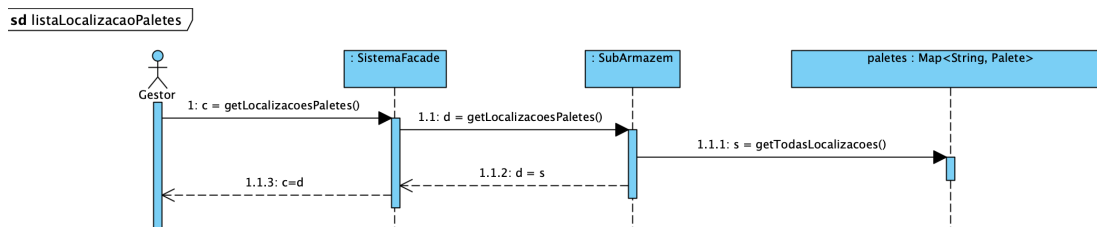


Figura 1.3: ListaLocalizacaoPaletes

O Gestor interage com o SistemaFacade para obter a lista de localização das paletes. Em seguida, este comunica com o subsistema do armazém e será esse que calcula todas as localizações de todas as paletes.

Iniciar Sessão Gestor

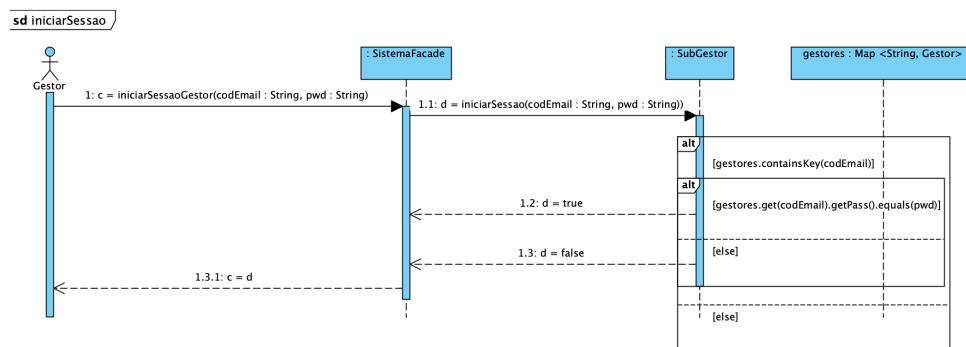


Figura 1.4: IniciarSessao

O Gestor interage com o Sistema Facade para iniciar sessão. Posteriormente, este comunica com o subsistema dos Gestores onde irá fazer a verificação do email e da password.

Terminar Sessão Gestor

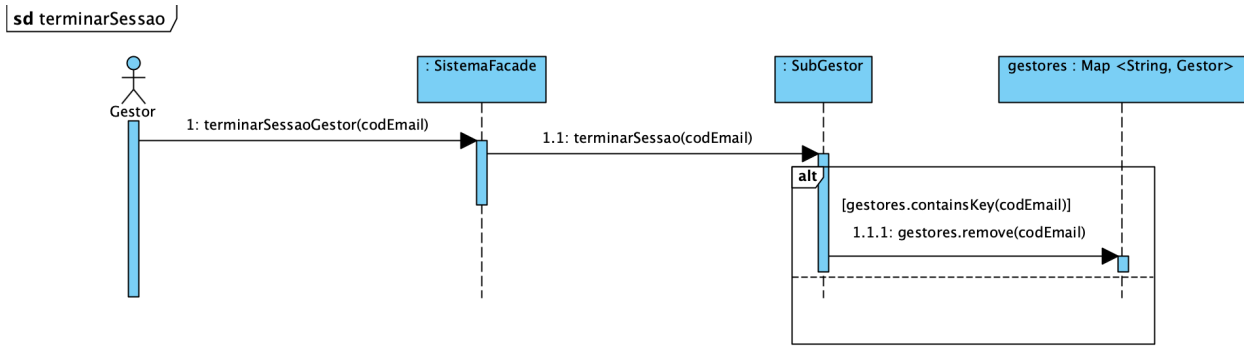


Figura 1.5: terminarSessao

O Gestor interage com o Sistema Facade para terminar sessão. Posteriormente, este comunica com o subsistema dos Gestores onde irá verificar se a sua conta existe.

Check Conta Gestores

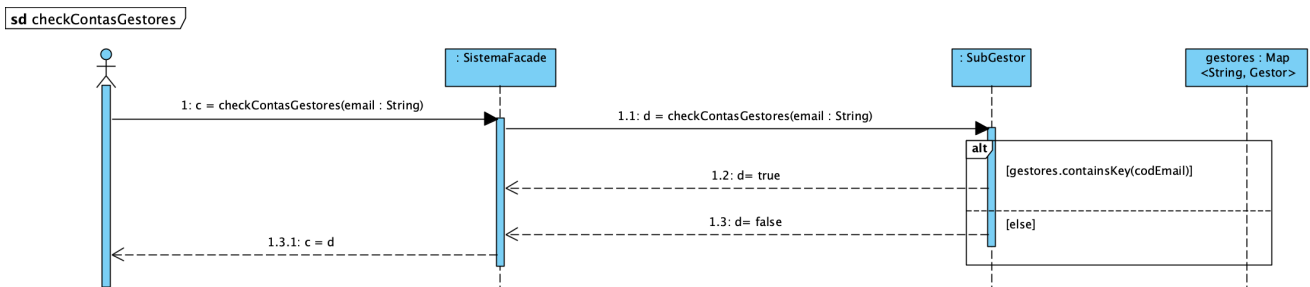


Figura 1.6: checkContaGestores

O ator interage com o Sistema Facade para verificar a existência de um Gestor.

Registrar Gestor

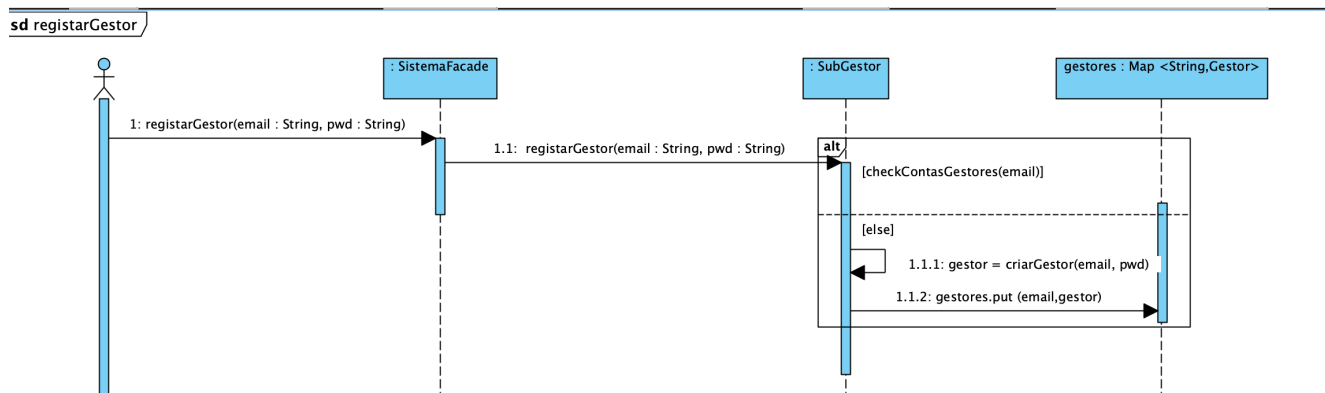


Figura 1.7: registrarGestor

O ator interage com o Sistema Facade para registrar um Gestor. Este comunica com o subsistema de Gestores onde será necessário o email e password do mesmo.

Check QRCode

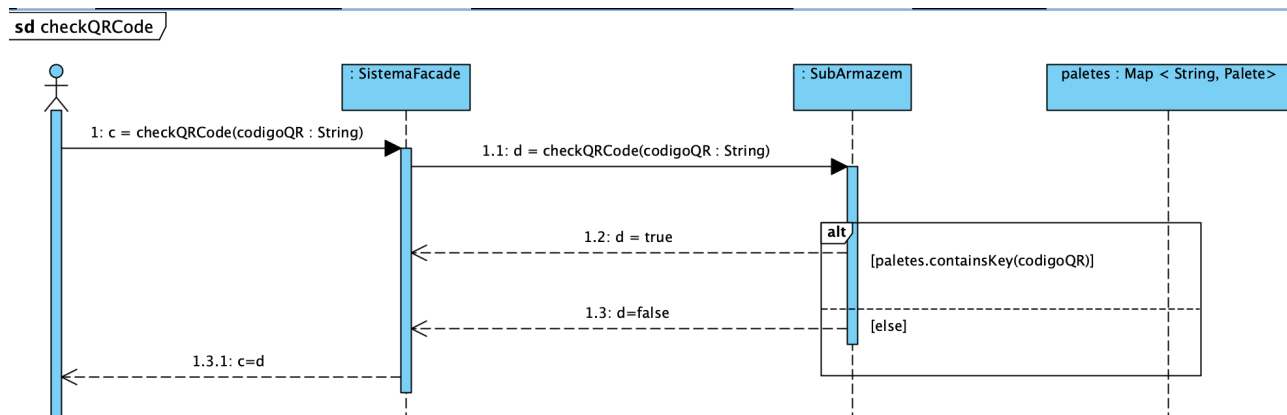


Figura 1.8: checkQRcode

O ator interage com o Sistema Facade para se verificar o QRCode de uma paleta.

Registrar Produto

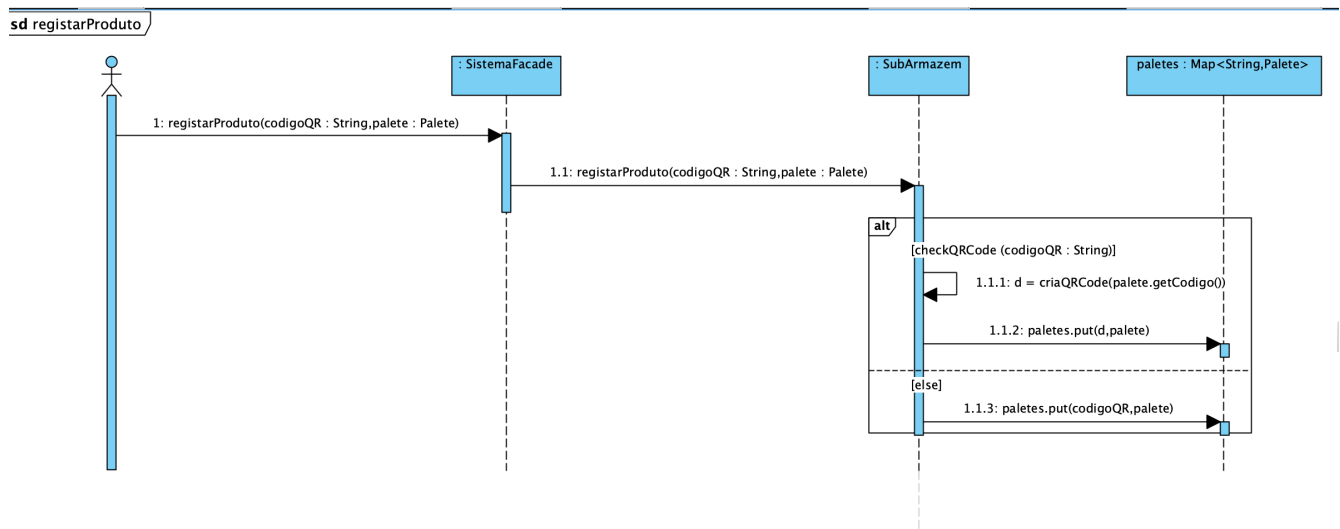


Figura 1.9: registrarProduto

O ator comunica com o Sistema Facade para registrar um produto.

Criar QRCode

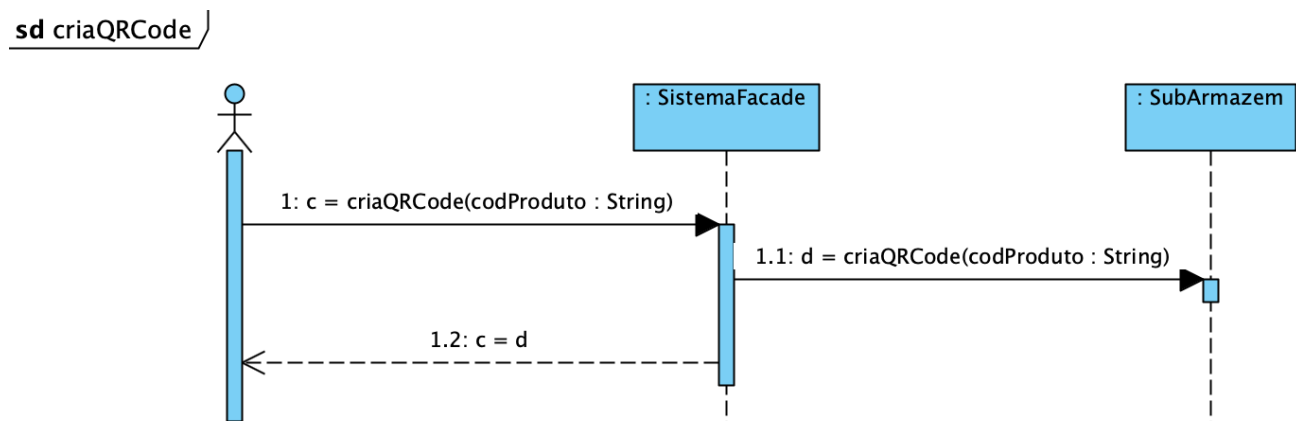


Figura 1.10: CriaQRCode

O ator comunica com o Sistema Facade para criar o QRCode referente a uma paleta.

Verifica Disponibilidade Robot

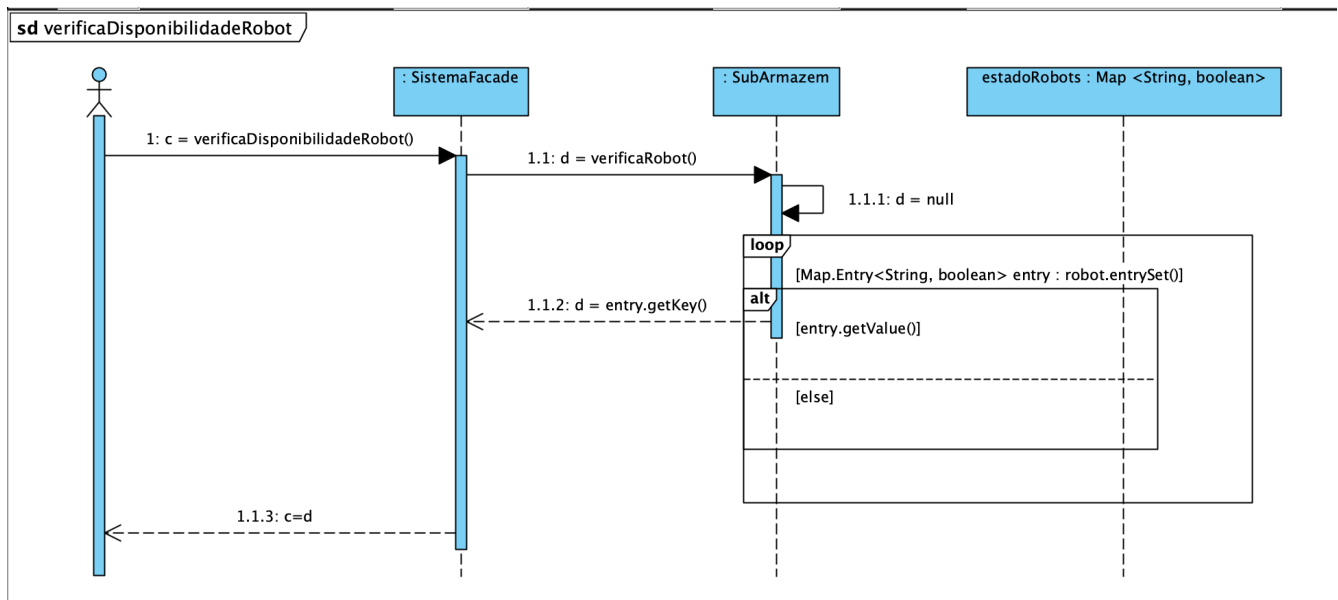


Figura 1.11: VerificaDisponibilidadeRobot

O ator comunica com o Sistema Facade para verificar se existe um Robot disponível. O subsistema armazém devolve o código do primeiro robot que está disponível.

Elege Robot

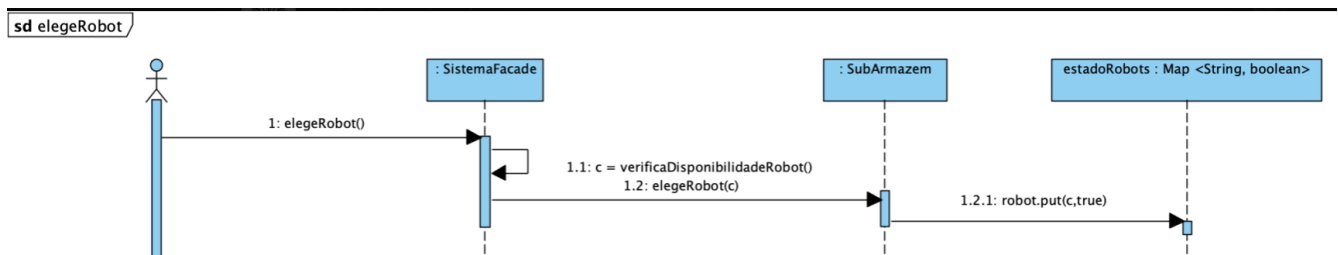


Figura 1.12: elegeRobot

O ator comunica com o Sistema Facade para eleger um robot para efetuar um transporte.

Calcula Percurso Robot

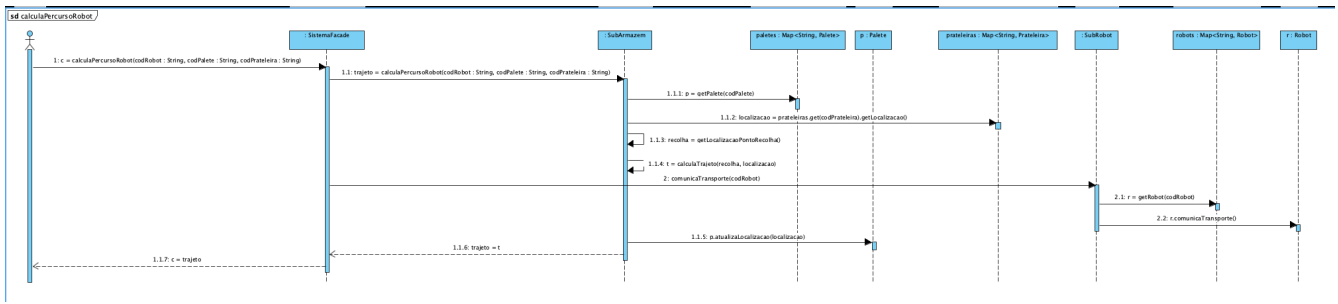


Figura 1.13: calculaPercursoRobot

O sistema comunica com o subsistema do armazém e esse subsistema vai calcular o percurso do robot mediante a localização do destino. O subsistema do armazém terá de atualizar a localização da palette. O sistema vai comunicar com o robot para dar a ordem de fazer o trajeto.

Get Coordenadas Paletes

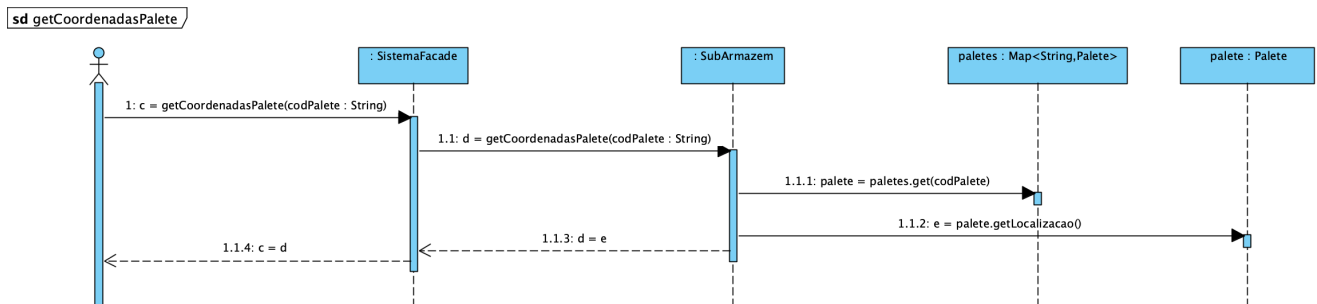


Figura 1.14: GetCoordenadasPalette

O sistema comunica com o subsistema do armazém para obter a localização de uma determinada palette.

Recolha Efetuada

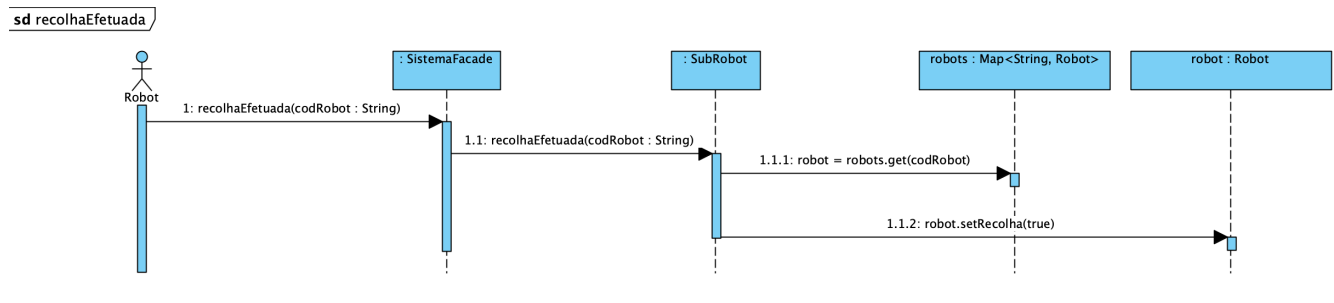


Figura 1.15: RecolhaEfetuada

O Robot comunica com o Sistema Facade para informar que recolheu uma determinada palete. O sistema, neste caso, tem de comunicar com o subsistema dos robots para atualizar o estado de recolha do robot.

Descarga Efetuada

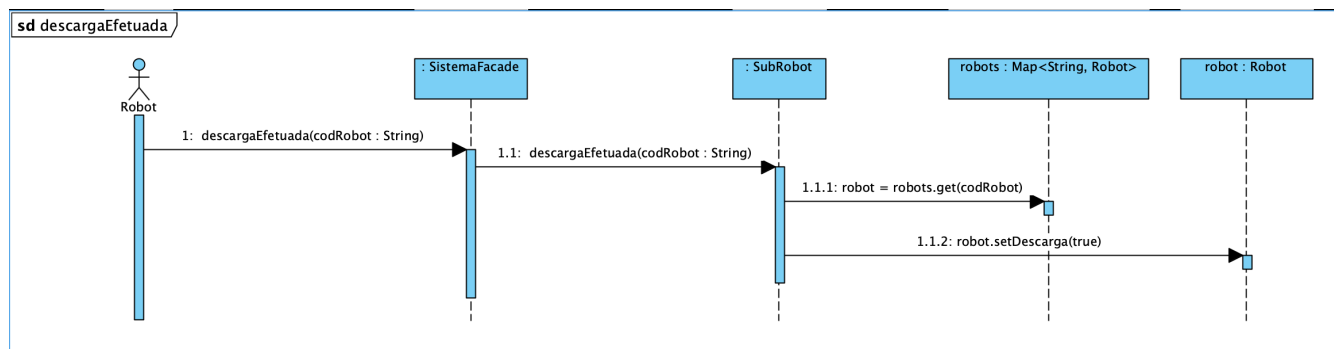


Figura 1.16: DescargaEfetuada

O Robot comunica com o Sistema Facade para informar que descarregou uma determinada palete. O sistema, neste caso, tem de comunicar com o subsistema dos robots para atualizar o estado de descarga do robot.

Validar Disponibilidade

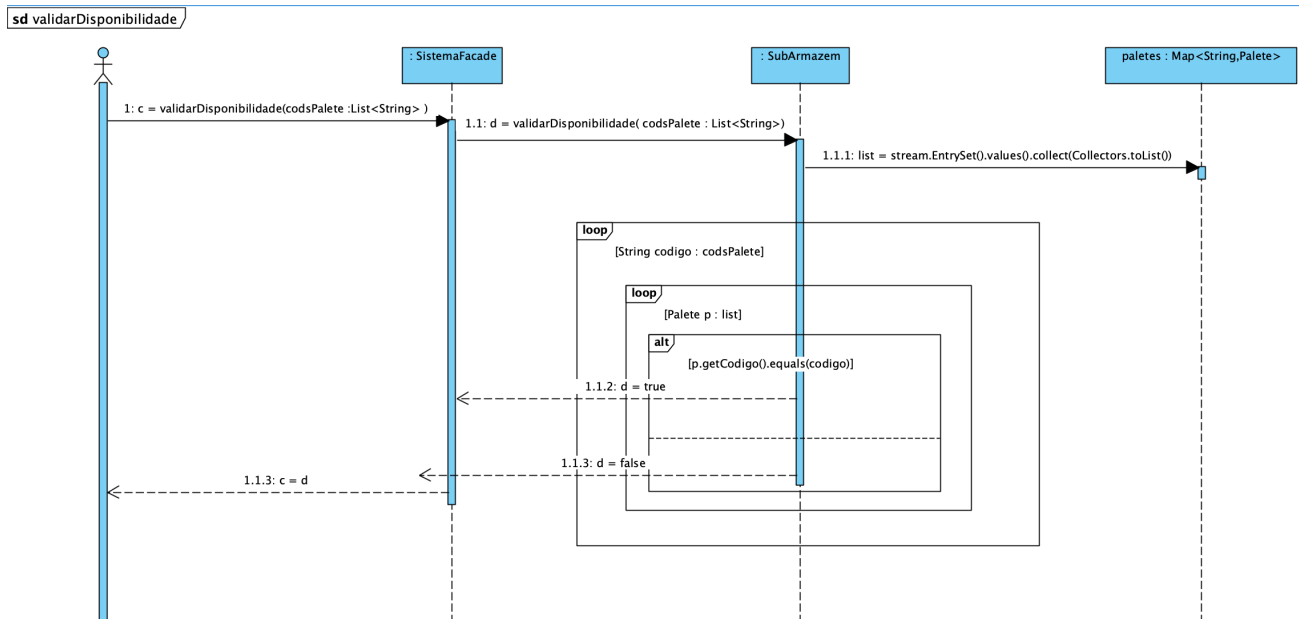


Figura 1.17: validarDisponibilidade

O ator comunica com o Sistema Facade para validar a disponibilidade de uma listagem de paletes.

Regista Requisição



Figura 1.18: RegistaRequisicao

O ator comunica com o Sistema Facade para registar uma requisição. O subsistema da requisição cria um código de registo respetivo.

Cancela Requisição

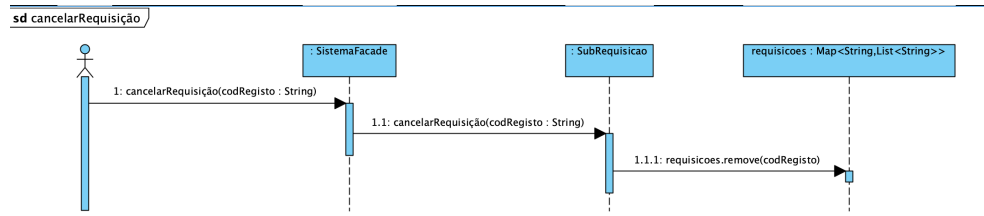


Figura 1.19: cancelaRequisicao

O ator comunica com o Sistema Facade para cancelar uma requisição do subsistema de requisições.

Registrar Palete Falta



Figura 1.20: registrarPaleteFalta

O ator comunica com o Sistema Facade para registrar paletes em falta referentes a uma requisição.

Registo Paletes a Entregar

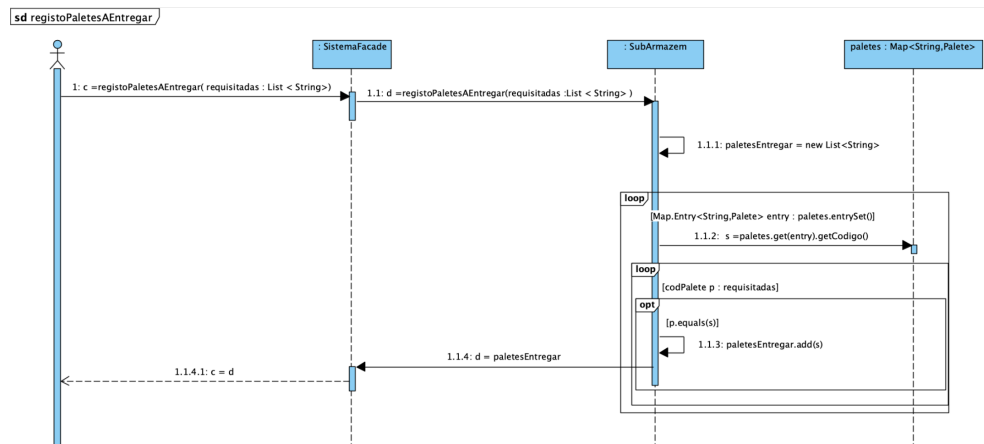


Figura 1.21: registoPaletesAEntregar

O ator comunica com o Sistema Facade para registrar paletes a entregar mediante a listagem de paletes requisitadas.

Diagrama de Classes

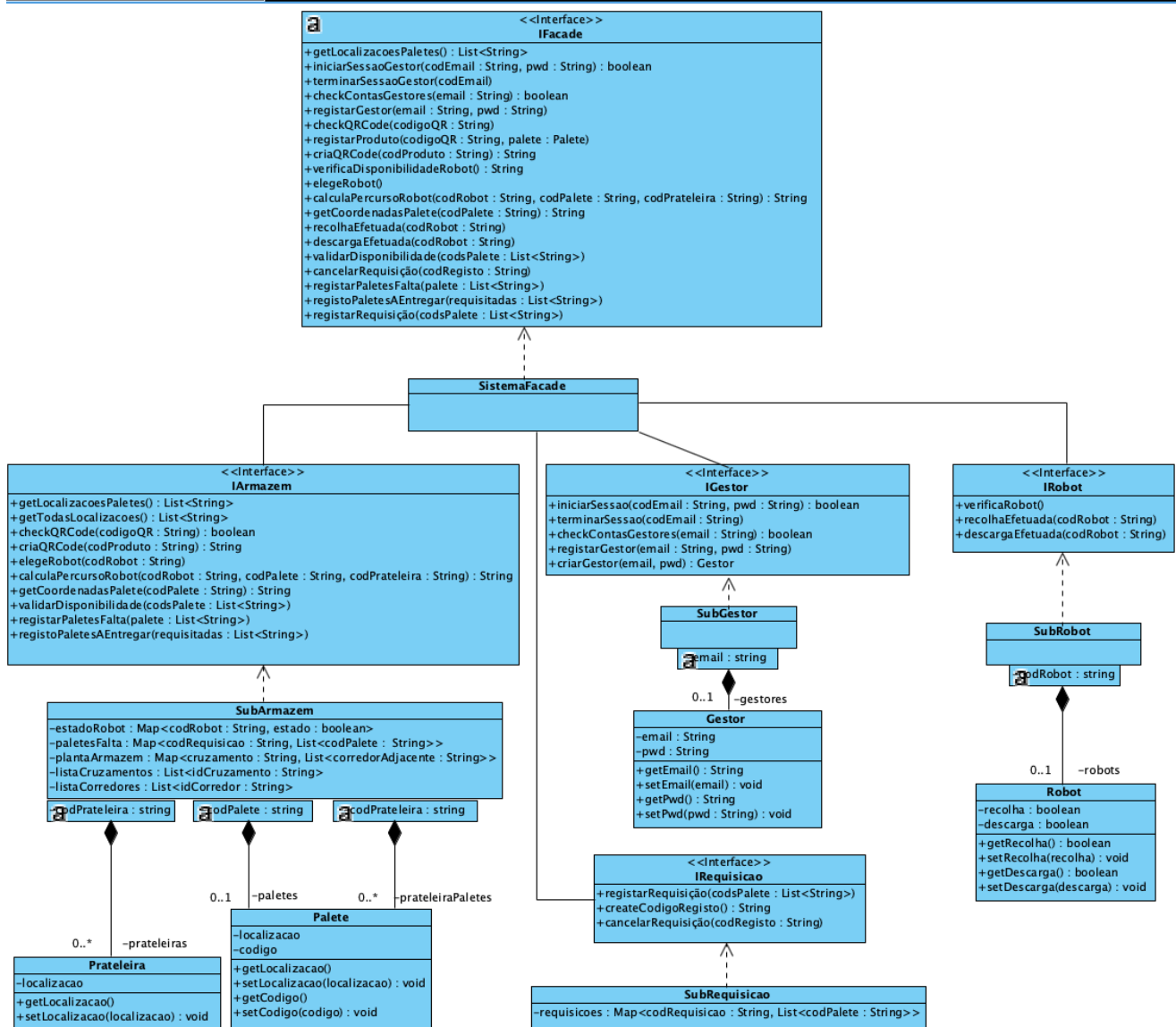


Figura 1.22: Diagrama de Classes

O nosso sistema implementa uma interface que será a intergace visívelao exterior (IFacade). Da mesma maneira, o sistema apenas tem conhecimento de interfaces dos seus subsistemas. Estas interfaces apenas contêm os métodos necessários para devolver a informação requerida. Os subsistemas terão atributos e estes serão privados (em que o sistema principal não terá nenhuma informação acerca deles).

Neste caso, durante o desenvolvimento dos diagramas de sequência, sentimos a necessidade de criar estruturas de dados para cada subsistema.

SubArmazém :

- **estadoRobot** : estrutura para identificar um robot e o seu estado de disponibilidade.
- **paletesFalta** : estrutura para identificar as paletes que estão em falta.
- **plantaArmazem** : estrutura para definir o mapa de um armazém.
- **listaCruzamentos** : estrutura para identificar os cruzamentos entre corredores do armazém.
- **listaCorredores** : estrutura para identificar os corredores do armazém.
- **prateleiras** : estrutura para associar um código de prateleira com uma prateleira.
- **paletes** : estrutura para associar um código QR-code de uma paleta com uma paleta.
- **prateleiraPaletes**: estrutura para associar um código de prateleira com uma listagem de paletes.

SubRobot :

- **robots** : estrutura para identificar um robot e o seu respetivo código de identificação.

SubGestor :

- **gestores**: estrutura para identificar um gestor e o seu respetivo email.

SubRequisição :

- **requisicoes**: estrutura para identificar um código de uma requisição com a listagem de paletes associada.

Diagrama de Componentes

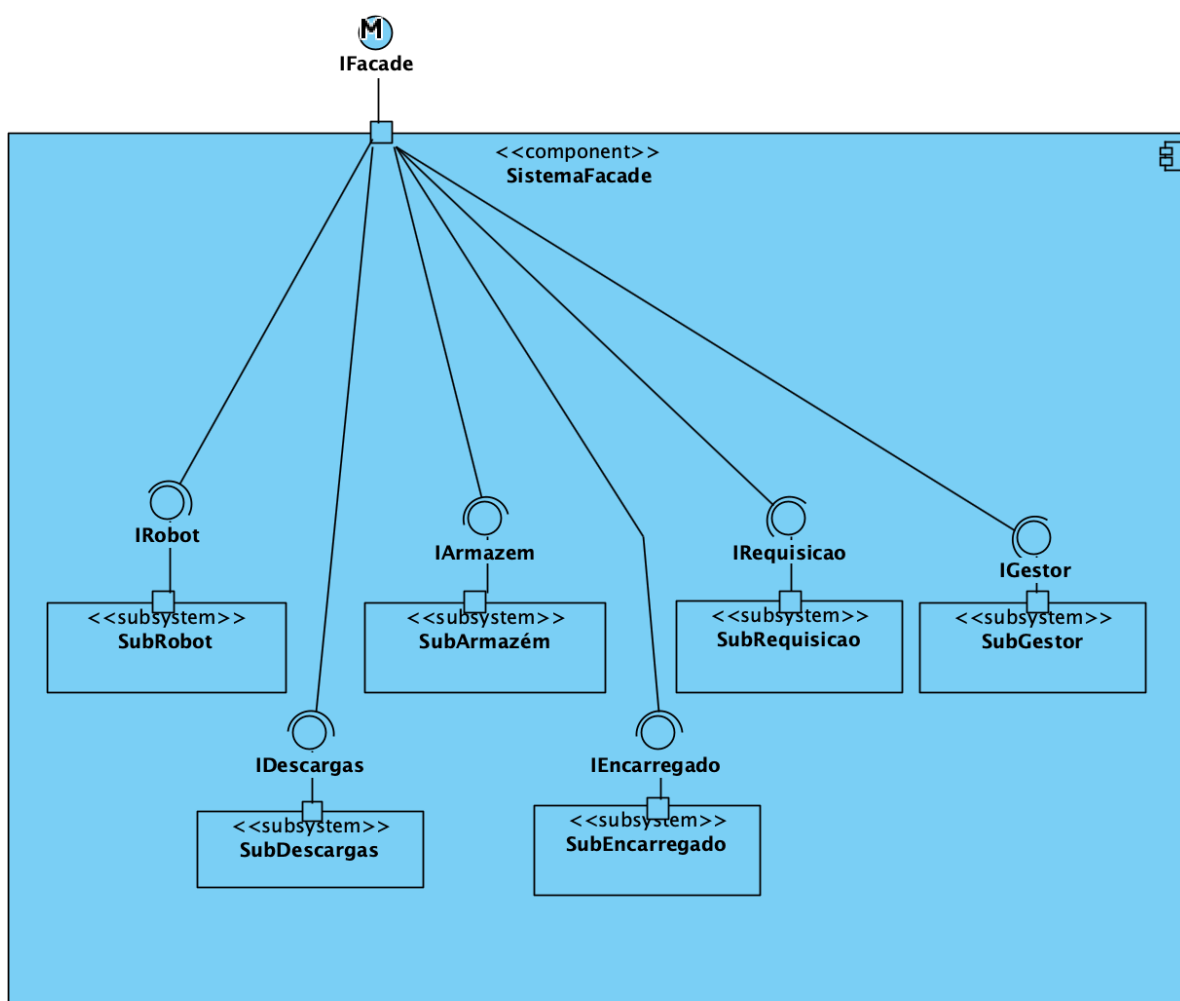


Figura 1.23: Diagrama de Componentes

O nosso Diagrama de Componentes (SistemaFacade) contém vários subsistemas: SubRobot, SubArmazém, SubRequisicao, SubDescargas, SubEncarregado, SubRequisicao, SubGestor. Estes subsistemas serão os responsáveis pela gestão do nosso sistema.

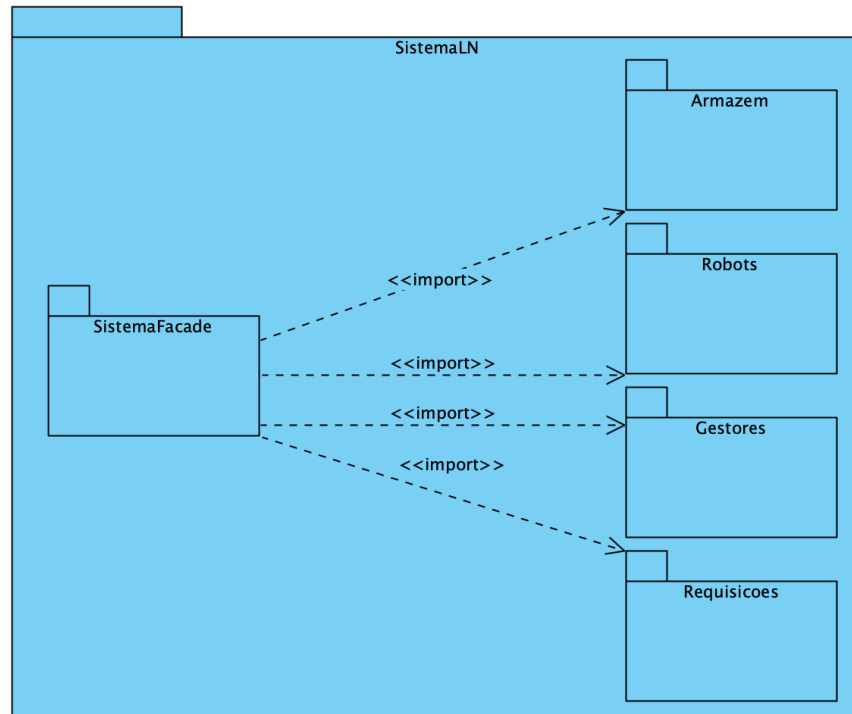


Figura 1.24: Diagrama de Packages

Este diagrama de packages é **referente apenas à parte lógica de negócio** a implementar. Numa fase posterior, será necessário implementar packages referentes à parte lógica da camada da interface e à camada de operações de acesso a dados. Assim, esperamos subdividir todas as classes do nosso sistema nestas 3 partes lógicas.

Na nossa camada de lógica de negócio, nesta fase, subdividimos em cinco packages mais pequenos, uma vez que se trata de uma camada muito extensa, como podemos ver no nosso diagrama acima.

- O principal papel do package ***SistemaFacade*** será o de fazer a ligação da interface com a camada lógica.
- Dentro do package ***Armazem*** teremos todo o tratamento necessário relativamente à de descargas e recolhas de paletes e ao mapeamento do armazém que o sistema implementará.
- Dentro do package ***Robots*** teremos todo o tratamento necessário relativo à gestão de todos os robots integrantes no armazém.
- Dentro do package ***Gestores*** teremos todo o tratamento necessário relativo à gestão de todos os gestores integrantes no armazém.
- Dentro do package ***Requisicoes*** teremos todo o tratamento necessário relativo à gestão de todas as requisições feitas ao armazém.

Como identificado no diagrama de componentes, sentimos também necessidade de criar subsistemas para englobar os encarregados e a gestão das descargas. Eventualmente, estes dois subsistemas também serão implementados em respetivos packages para que a gestão seja mais fácil e cómoda.

Capítulo 2

Conclusão

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

Ainda assim, tivemos que implementar um diagrama de classes onde sao colocados todas as operações necessárias, tais como os relacionamentos entre o sistema, as interfaces dos subsistemas, e os próprios subsistemas.

Para finalizar, tivemos de formular o diagrama de packages, para complementar a fase pretendida. Tivemos, ainda, que acrescentar alguns Use Cases não utilizados na primeira fase para uma melhor realização da fase 2.

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