

UNIVERSIDADE DO MINHO  
DEPARTAMENTO DE INFORMÁTICA  
DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE

Fase 3  
Grupo N<sup>o</sup> 23

Pedro Pereira (A80627)  
Sofia Marques (A87963)  
Pedro Pereira (A89232)  
Eduardo Silva (A89516)  
José Martins (A90122)

23 de Dezembro de 2020



Pedro Pereira A80627



Sofia Marques A87963



Pedro Pereira A89232



Eduardo Silva A89516



José Martins A90122

# Conteúdo

<b>1</b>	<b>Use Cases</b>	<b>3</b>
<b>2</b>	<b>Responsabilidades do Sistema</b>	<b>5</b>
<b>3</b>	<b>Diagrama de Classes</b>	<b>6</b>
<b>4</b>	<b>Modelo Lógico</b>	<b>7</b>
<b>5</b>	<b>Diagramas de Sequência</b>	<b>8</b>
<b>6</b>	<b>Diagrama de Máquina de Estados</b>	<b>9</b>
<b>7</b>	<b>Utilização do Programa</b>	<b>10</b>
<b>8</b>	<b>Conclusão</b>	<b>13</b>
<b>9</b>	<b>Anexos</b>	<b>14</b>
9.1	Gestor . . . . .	14
9.2	Leitor de Códigos QR . . . . .	15
9.3	Robot . . . . .	15
9.3.1	Notificação Para Transporte . . . . .	15
9.3.2	Notificação de Recolha de Paleta . . . . .	18
9.3.3	Notificação da Entrega da Paleta . . . . .	19

# Capítulo 1

## Use Cases

Para a terceira fase deste projeto foi-nos proposto, pelos docentes, uma lista de Use Cases a considerar e implementar, sendo esta lista a seguinte:

- Gestor:
  - Consultar listagem de localizações;
- Leitor de Códigos QR:
  - Comunicar código QR.
- Robot:
  - Sistema comunica ordem de transporte;
  - Notificar recolha de paletes;
  - Notificar entrega de paletes.

Após uma revisão destes use cases, decidimos aperfeiçoar a especificação do use case de Notificação para Transporte, de forma a representar mais corretamente o seu funcionamento:

Use Case		
Use Case:	Notificação para transporte	
Ator:	Robot	
Descrição:	O sistema notifica o robot para transporte de paletes	
Pré-Condição:	True	
Pós-Condição:	O robot está notificado	
	Ator	Sistema
Cenário Normal		1. Determina a localização da paleta
		2. Verifica se existem robots disponíveis
		3. Determina o robot mais próximo
		4. Calcula percurso
		5. Notifica o robot

Figura 1.1: Use Case - Notificação para Transporte

## Capítulo 2

# Responsabilidades do Sistema

Tendo corrigido a especificação dos use cases, tornou-se também necessário a retificação das responsabilidades do sistema definidas anteriormente.

Sendo assim, a tabela atualizada das responsabilidades do sistema é a apresentada abaixo:

Use Cases	Responsabilidades (sistema)	API	Subsistemas
Solicitar Listagem	Determinar as localizações das paletes	<code>getLocalizacoes() : Map&lt; String, String&gt;</code>	Armazém
Leitura de Código QR	Registrar palete no sistema	<code>registrarPaletaSistema(codQR: String, codProd: String, nomeProd: String)</code>	Armazém
Notificação Para Transporte	Determinar Localização da Palete	<code>determinarLocalizacao(codQR: String) : String</code>	Armazém
	Calcular percurso	<code>calcularPercurso(codRobot: String, locPaleta: String, locFinal:String): Percurso</code>	Armazém
	Verificar se existem robots disponíveis	<code>getListaRobotsDisponiveis() : List&lt;String&gt;</code>	Armazém
	Escolher o robot que está mais próximo	<code>getRobotMaisProximo(locPaleta: String, listRobot:List&lt;String&gt;):Robot</code>	Armazém
	Atualizar estado de robot	<code>atualizarEstadoRobot(codRobot:String, percurso:Percurso, codQR:String)</code>	Armazém
	Determinar uma prateleira vazia	<code>getPrateleiraVazia():String</code>	Armazém
	Atualizar estado de prateleira	<code>atualizarEstadoPrateleira(codPrateleira:String, ocupada:Boolean)</code>	Armazém
Notificação de Recolha de Palete	Registrar recolha de palete	<code>registrarRecolha(codRobot: String)</code>	Armazém
Notificação da Realização da Entrega de Palete	Registrar entrega de Palete	<code>registrarEntrega(codRobot:String)</code>	Armazém

Figura 2.1: Responsabilidades do Sistema

## Capítulo 3

# Diagrama de Classes

Um diagrama de classes é uma representação da estrutura e relações das classes. Nesse sentido, passámos por várias fases até obtermos um diagrama de classes final. Após analisar os use cases pedidos pelos docentes para a implementação final, e tendo em conta o tempo relativamente limitado que tínhamos para executar a implementação, em conjunto com dificuldades que tivemos durante a implementação do programa, em particular na implementação da base de dados, decidimos recorrer a algumas simplificações. Também não representamos as classes que não eram necessárias para os use cases cuja implementação não foi pedida.

Sendo assim, o nosso diagrama de classes para o projeto é o seguinte:

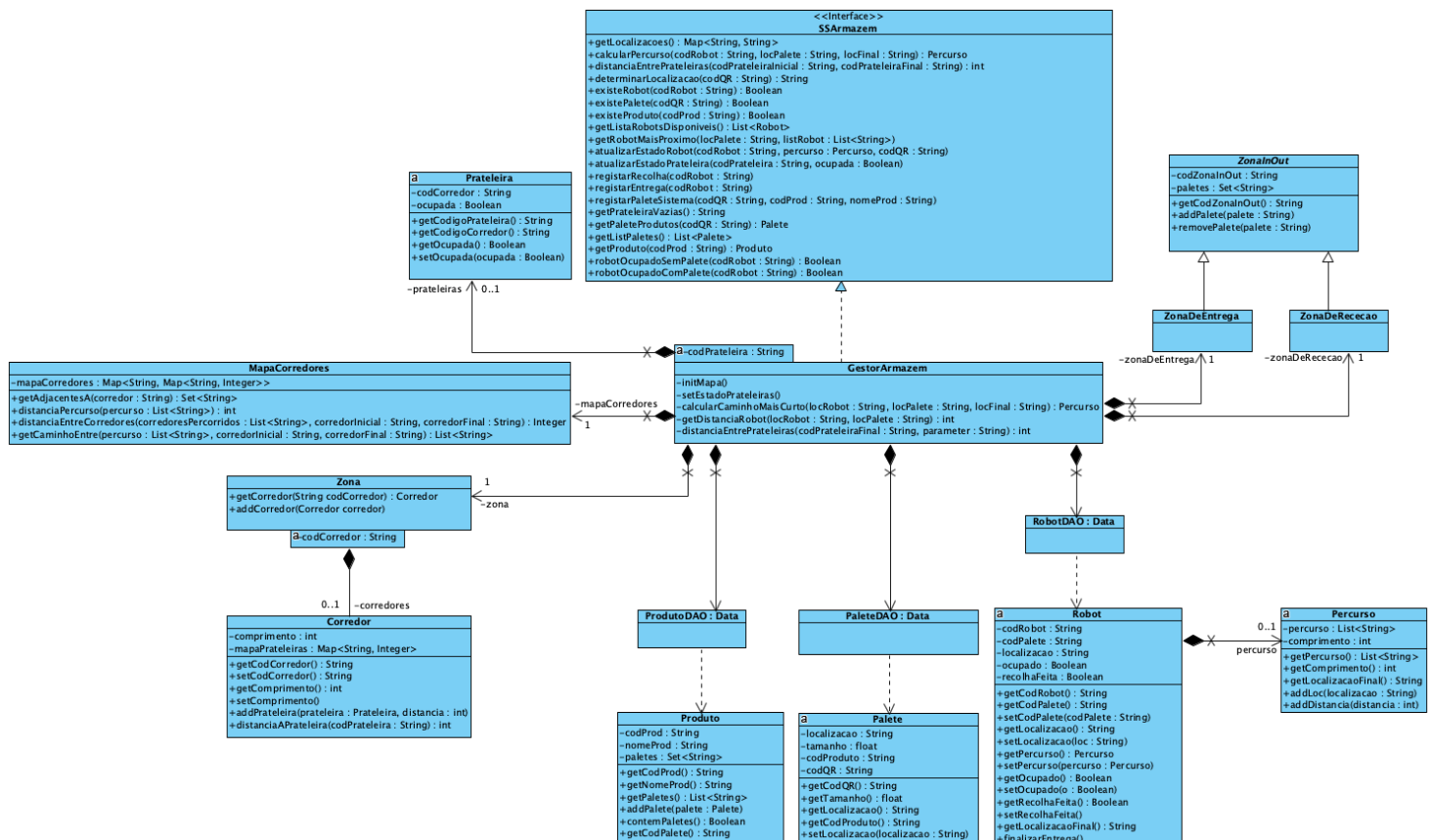


Figura 3.1: Diagrama de Classes

## Capítulo 4

# Modelo Lógico

O Modelo Lógico representa graficamente a base de dados e permite entender melhor o que cada tabela armazena.

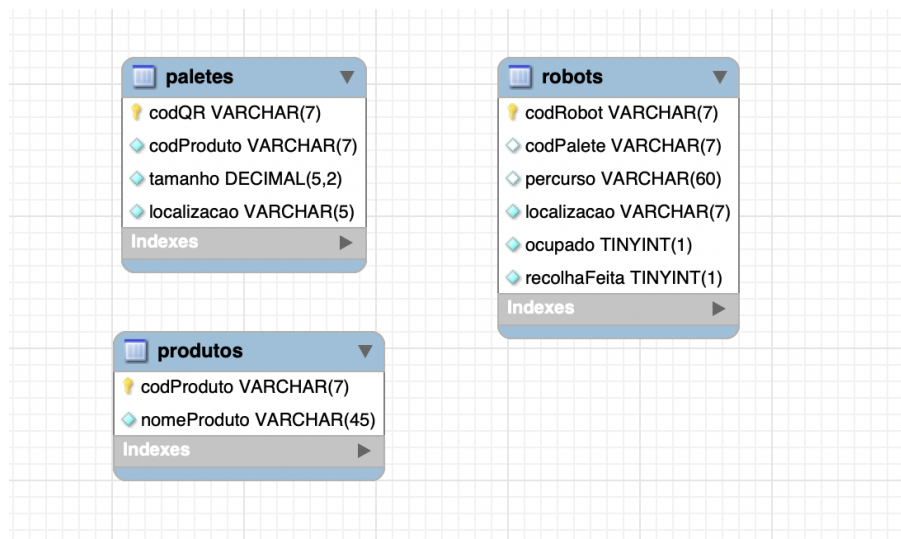


Figura 4.1: Modelo Lógico

## Capítulo 5

# Diagramas de Sequência

Estes diagramas têm como principal objetivo representar interações entre objetos através de mensagens que são trocadas entre eles, com uma ordenação temporal. Uma vez que a arquitetura do programa foi alterada, foi necessária a atualização dos diagramas de sequência apresentados na fase 2, de forma a representarem corretamente o funcionamento da lógica de negócio.

De seguida apresenta-se um exemplo de um diagrama de sequência produzido para esta fase e os restantes poderão ser encontrados em *Anexos*, no fim do relatório.

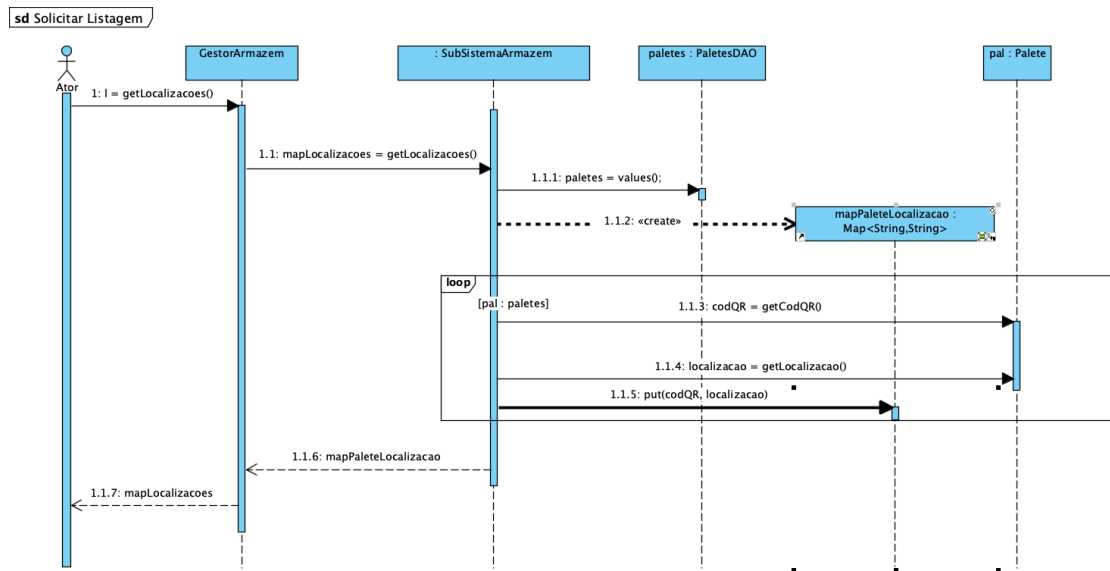


Figura 5.1: Exemplo de Diagrama de Sequências



## Capítulo 6

# Diagrama de Máquina de Estados

De forma a facilitar a implementação dos Menus decidimos elaborar um diagrama de Máquina de Estados. Este diagrama tem como função representar transições entre os vários estados do menu do programa. Posto isto, o nosso diagrama de máquina de estados é o seguinte:

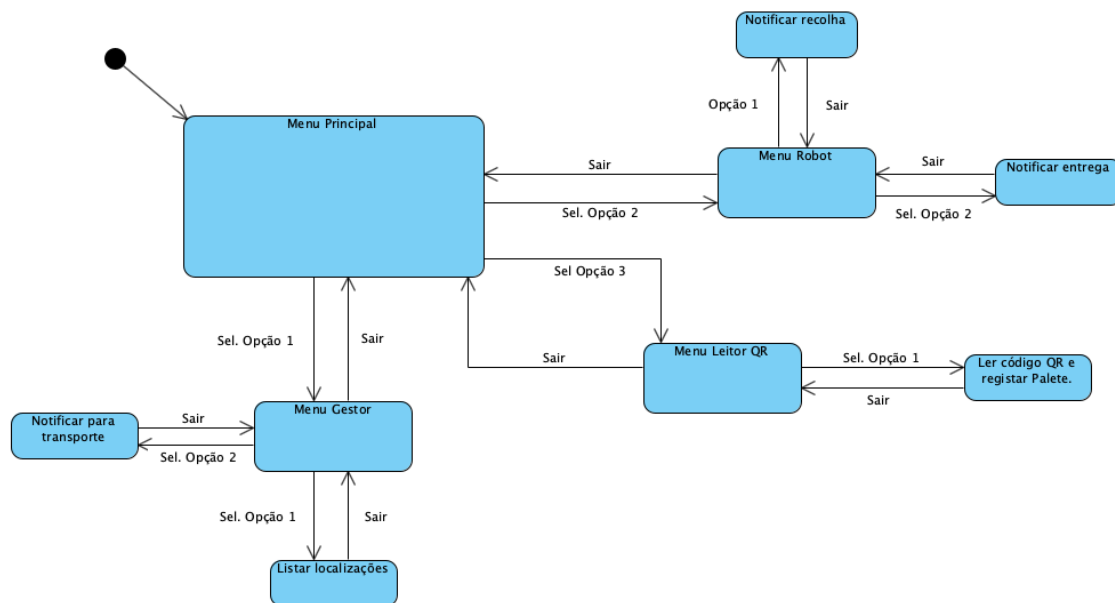


Figura 6.1: Diagrama de Máquina de Estados

# Utilização do Programa

[illegible]

Este é o menu principal, que dá acesso ao menu do Gestor, Robot e Leitor. A partir dele é possível aceder a todas as funcionalidades implementadas do Armazém.

```
-- --
|  \_  | |__ _ _ _ _ _  \_  | |__ _ _ _ | |__ _ _ _
| | \ | | / _ \ ' _ \ | | | | | _ / _ \ _ | | / _ \ ' _ \
| | | | _ \ | | | | | | | | | | _ \ _ \ | | ( ) | |
| | | | \_  | | | | \_  | | _ _ \_  | | _ \_  | |
1 - Listar localizações.
2 - Notificar para transporte.
0 - Sair.
Opção:
```

O Menu Gestor permite listar as localizações de todas as paletes do sistema, com informação detalhada acerca do seu estado e permite também tratar das notificações para transporte de paletes.



```
-- --
| \ / | | _ _ _ _ _ | | _ _ ( ) | _ _ _ _ _
| \ / | | / _ \ ' \ | | | | | | | / _ \ | _ / _ \ | ' _ | | | | |
| | | | _ / | | | | | | | | | _ / | | ( ) | |
| | | | \ _ _ | | | \ _ _ | | _ _ \ _ _ | | \ _ _ | |
1 - Ler código QR e registrar Paleta.
0 - Sair.
Opção: 1
```

Figura 7.5: Menu do Leitor

O menu Leitor permite registrar uma paleta no sistema.

```
Opção: 1
Código QR:   ex:(P_XXXXX)
P_00006
Insira o código do produto (7 caratères no máximo):
Prod01
Produto encontrado no sistema.
Paleta com código QR 'P_00006' e com código de produto 'Prod01' registada no sistema.
```

Figura 7.6: Resultado opção "Ler código QR e registrar Paleta".

Após a seleção da opção "Ler código QR e registrar Paleta", é pedido o código QR da paleta para registrar. Após a validação do código QR, é pedido o código e o nome do produto dessa paleta.

O processo de verificação e validação falha se o código QR da paleta já existe no sistema. Caso o código de produto já exista, o nome não vai ser pedido uma vez que esse produto já está registado no sistema.

## Capítulo 8

# Conclusão

Neste projecto conseguimos pôr em prática grande parte dos conhecimentos adquiridos nas aulas práticas e teóricas desta unidade curricular.

Pensamos ter conseguido uma implementação razoável, apesar de haver alguns pontos que poderiam ser melhorados caso houvesse mais tempo. Como referido anteriormente, tivemos problemas em implementar a base de dados, em particular, na utilização de chaves estrangeiras que permitiriam uma arquitetura mais clara e correta, e que infelizmente tivemos de contornar, não tendo chegado a uma implementação tão elegante e robusta, mas que ainda assim permitiu a conclusão das tarefas pedidas, ainda que não seja tão viável a longo prazo e para um desenvolvimento prolongado do programa. Apesar disto, todos os use cases pedidos estão implementados e com um funcionamento que pensamos que esteja dentro do esperado, como é possível ver na apresentação da UI apresentada anteriormente.

# Capítulo 9

## Anexos

### 9.1 Gestor

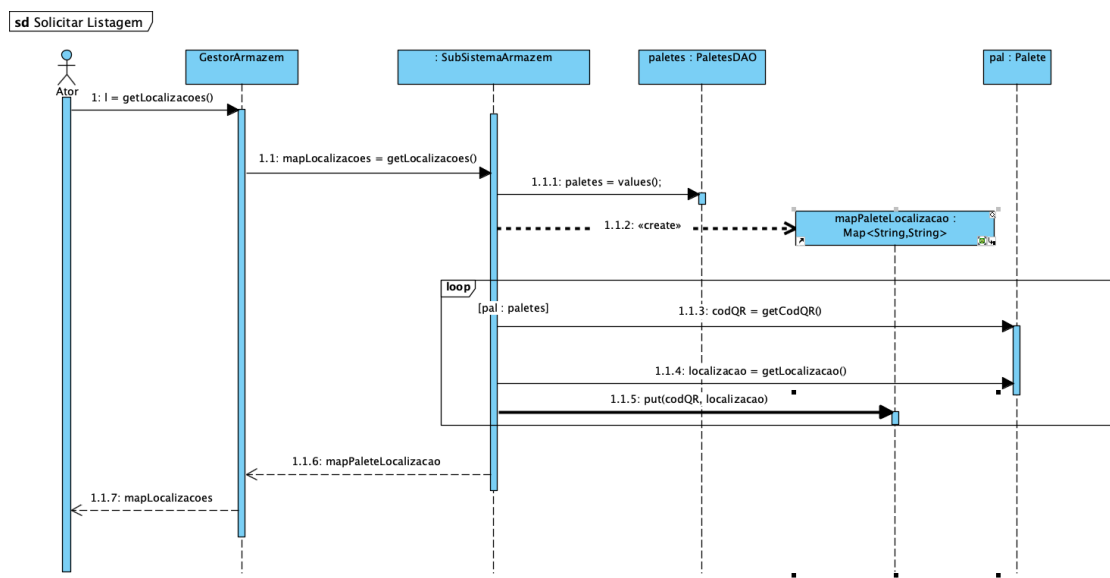


Figura 9.1: Diagrama de Sequência - Determinar As Localizações Das Paletes

## 9.2 Leitor de Códigos QR

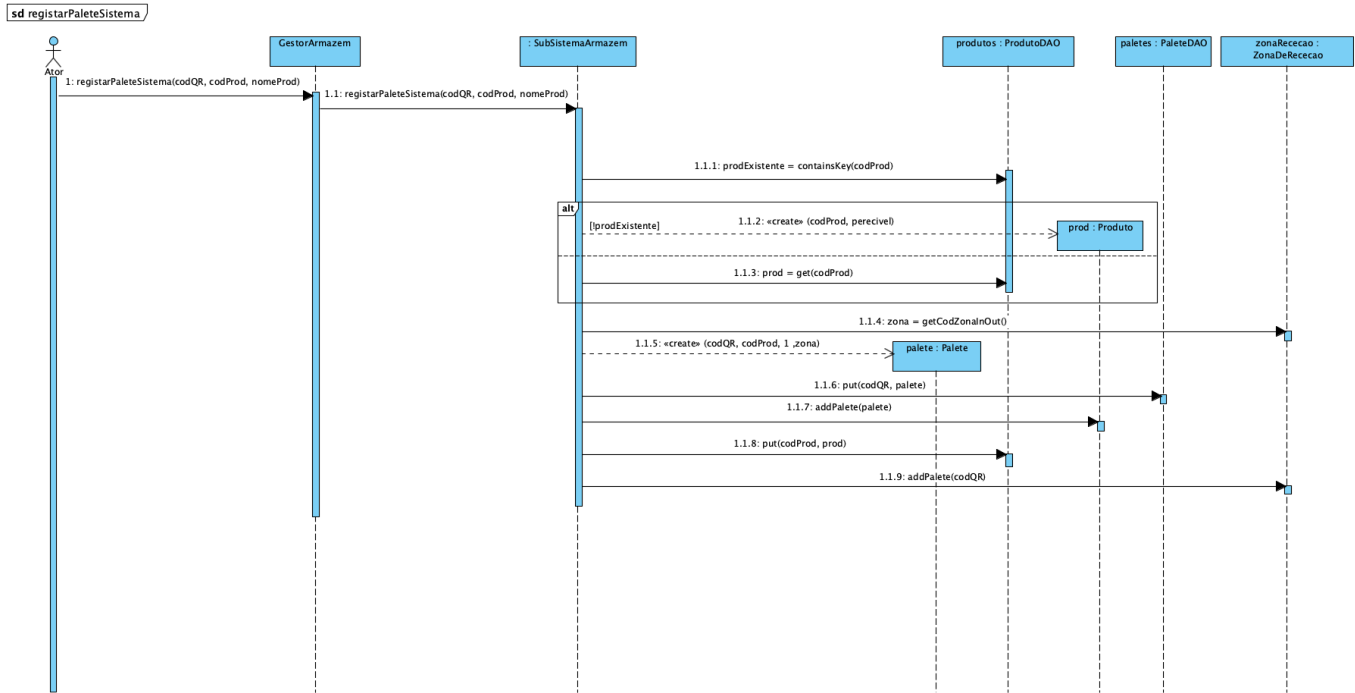


Figura 9.2: Diagrama de Sequência - Registrar Paleta no Sistema

## 9.3 Robot

### 9.3.1 Notificação Para Transporte

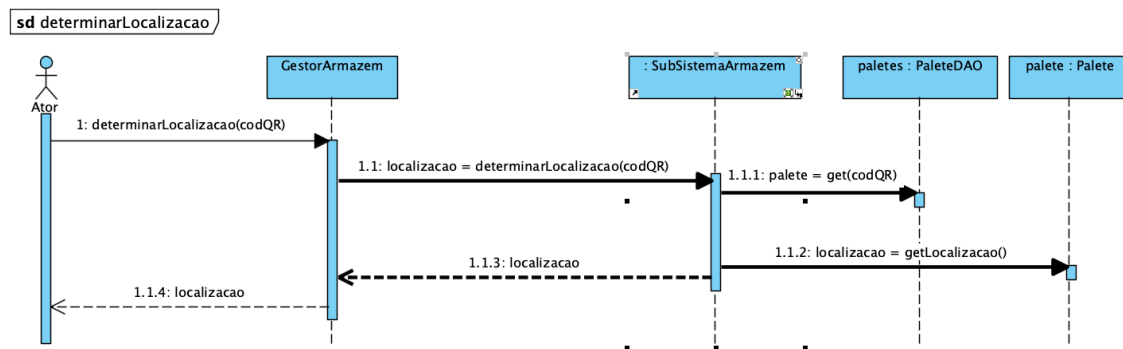


Figura 9.3: Diagrama de Sequência - Determinar Localização de uma Paleta

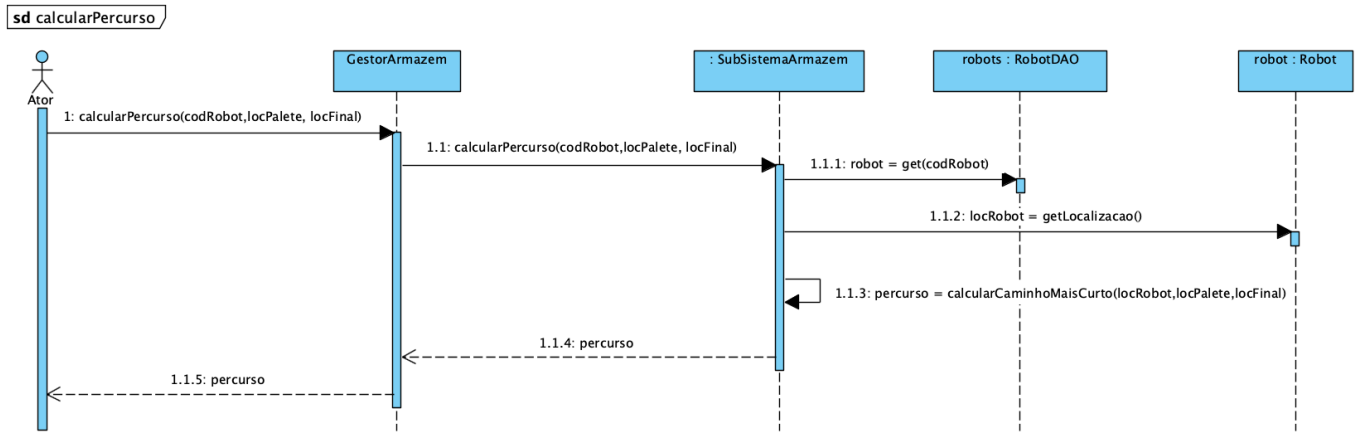


Figura 9.4: Diagrama de Sequência - Calcular Percurso

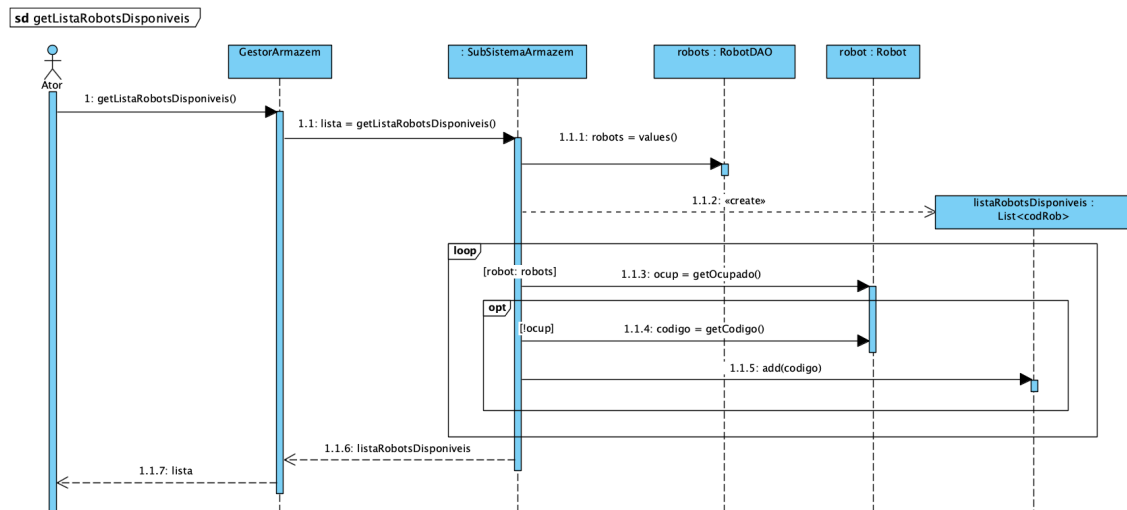


Figura 9.5: Diagrama de Sequência - Verificar se Existem Robots Disponíveis



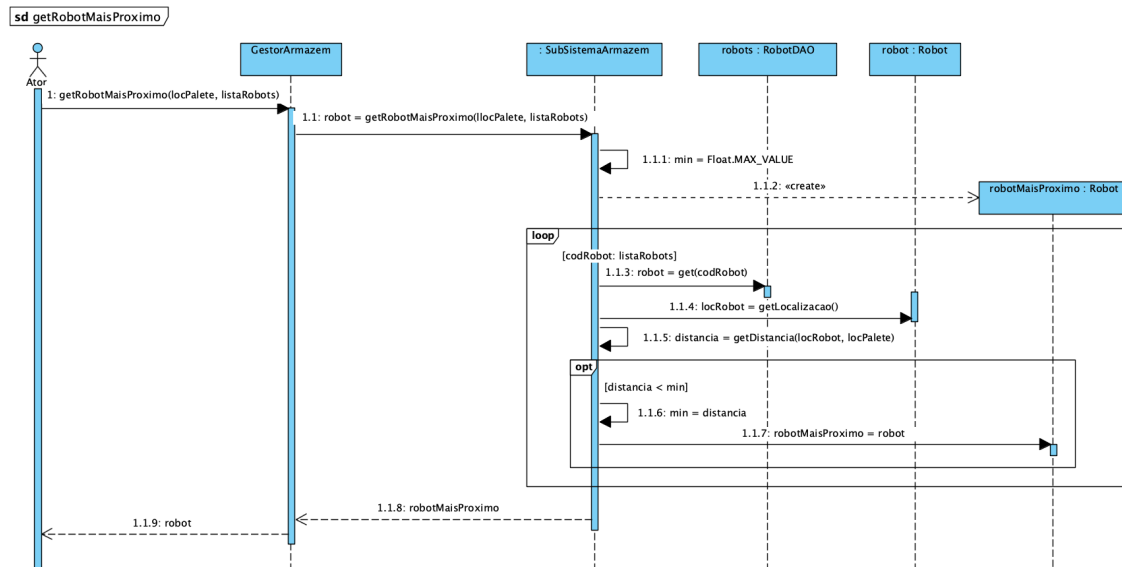


Figura 9.6: Diagrama de Sequência - Escolher o Robot Mais Próximo

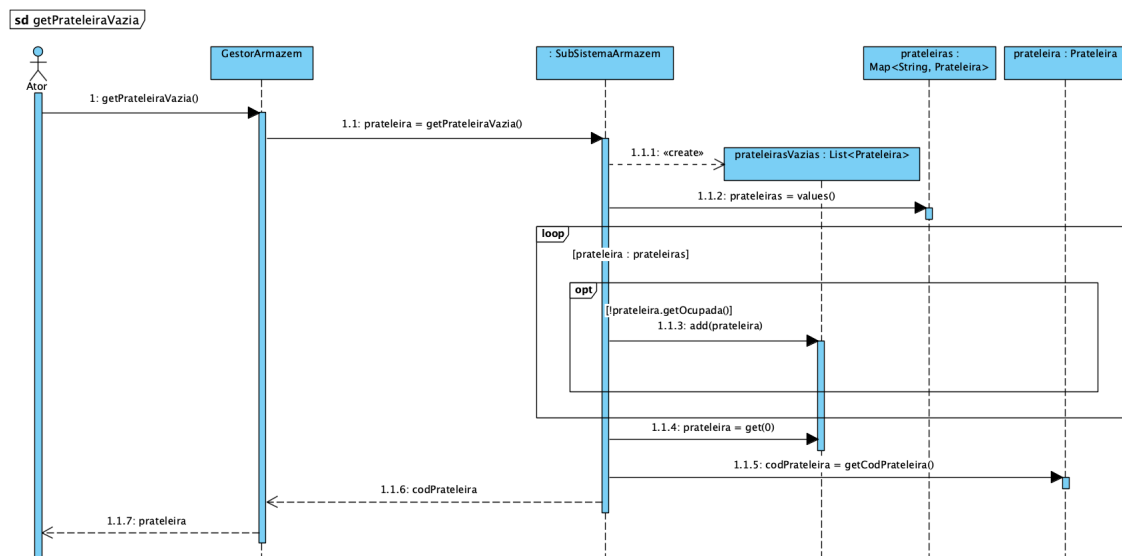


Figura 9.7: Diagrama de Sequência - Escolher uma prateleira vazia

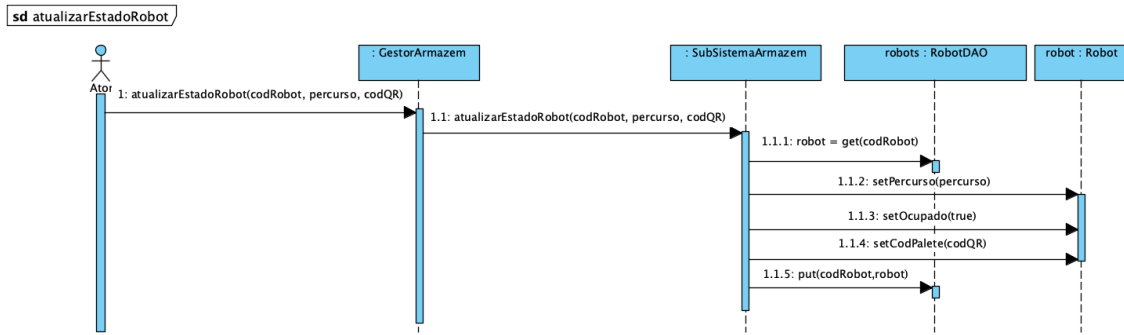


Figura 9.8: Diagrama de Sequência - Atualizar Estado de Robot

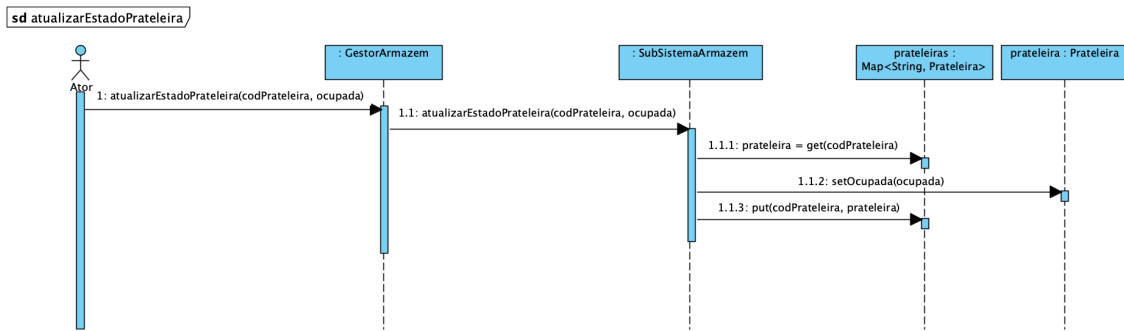


Figura 9.9: Diagrama de Sequência - Atualizar Estado de uma Prateleira

### 9.3.2 Notificação de Recolha de Paleta

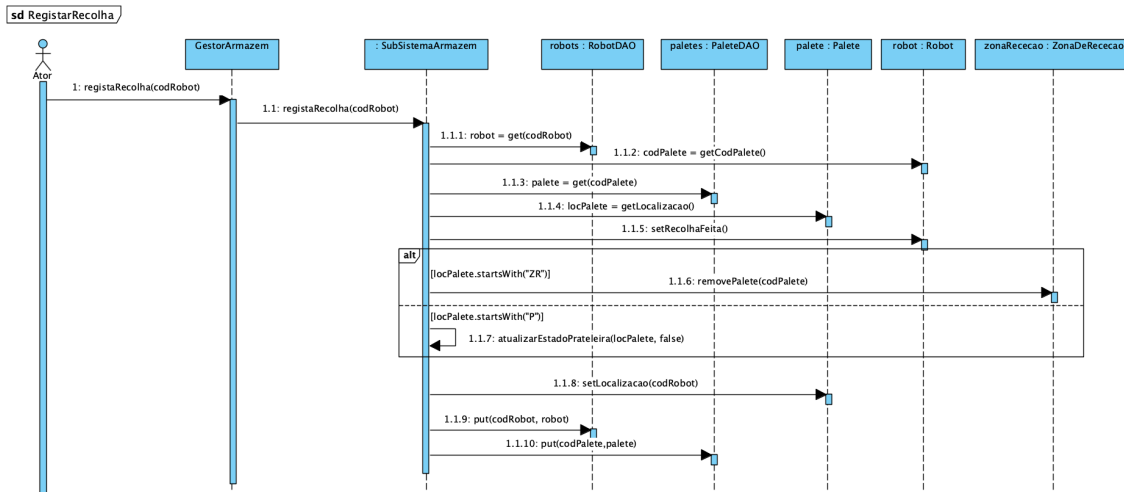


Figura 9.10: Diagrama de Sequência - Registrar Recolha de Paleta

### 9.3.3 Notificação da Entrega da Palette

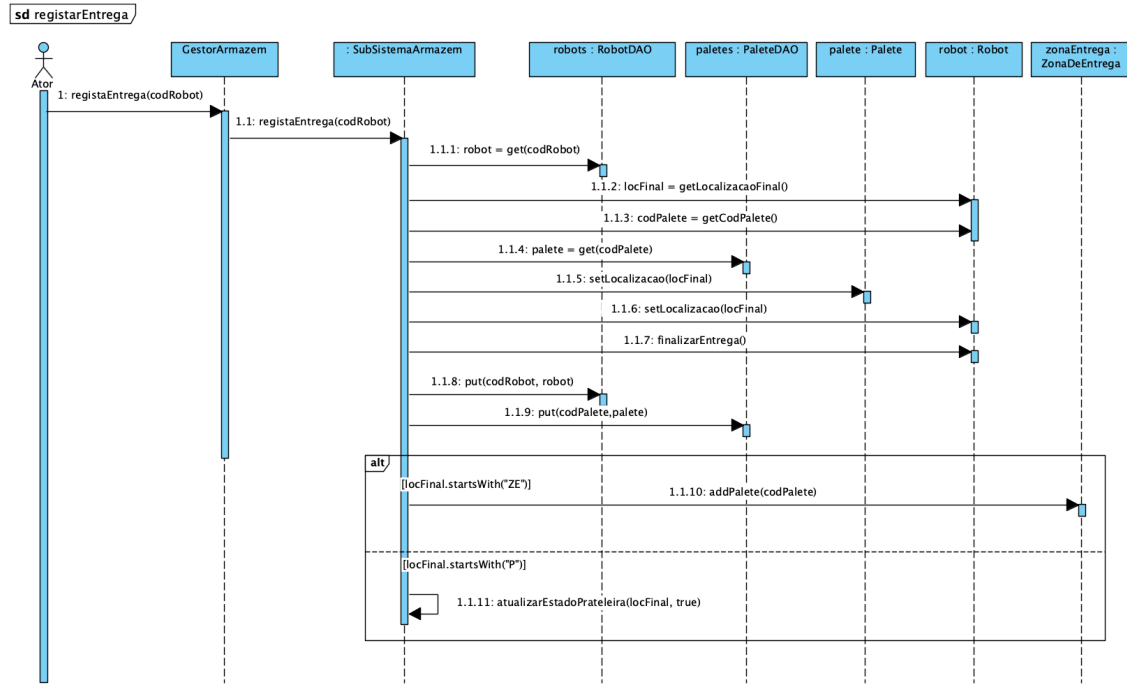


Figura 9.11: Diagrama de Sequência - Registrar Entrega de palette