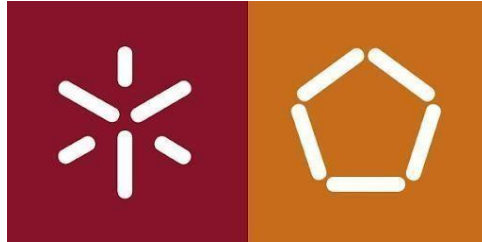


Universidade do Minho  
Licenciatura em Engenharia Informática



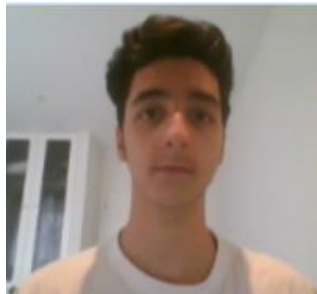
***Desenvolvimento de Sistemas de Software***  
***Grupo20***

*Braga, janeiro 2024*

**Trabalho realizado por:**



Diogo Gabriel Lopes  
Miranda  
a100839



João Ricardo  
Ribeiro Rodrigues  
a100598



Sandra Fabiana  
Pires Cerqueira  
a100681

URL do repositório: <https://github.com/LEI-DSS/trabalho-dss-grupo-20>

# Índice

1.Introdução	3
2.1 Modelo de Domínio	4
2.2 Diagrama de use cases	5
2.3 Especificação textual dos Use Cases	6
2.3.1 Autenticação no sistema	6
2.3.2 Verificar serviços do dia	6
2.3.3 Agendar um serviço	7
2.3.4 Realizar um serviço	8
2.3.5 Realizar um checkup	8
2.3.6 Enviar notificação para recolha do veículo	9
2.3.7 Cancelar um serviço	10
2.4 Diagrama de Componentes	11
2.5 Diagrama de Classes	12
2.5.1 Diagrama de classes Cliente	12
2.5.2 Diagrama de classes Mecânico	13
2.5.3 Diagrama de classes Posto de Trabalho	13
2.6 Diagramas de classes com DAO'S	14
2.6.1 Diagrama de classes Cliente com DAO'S	14
2.6.2 Diagrama de classes Mecânico com DAO'S	15
2.6.3 Diagrama de classes PostoDeTrabalho com DAO'S	16
2.7 Diagramas de Sequência	17
2.7.1 regNovosServico	17
2.7.2 consultFichaVeiculo	17
2.7.4 darHorarios	17
2.7.5 validCredenciais	18
2.7.6 realizarServico	18
2.7.7 validCompetencias	19
2.7.8 verfCompatibilidade	19
2.8 Conclusão e Análise dos Resultados Obtidos	20

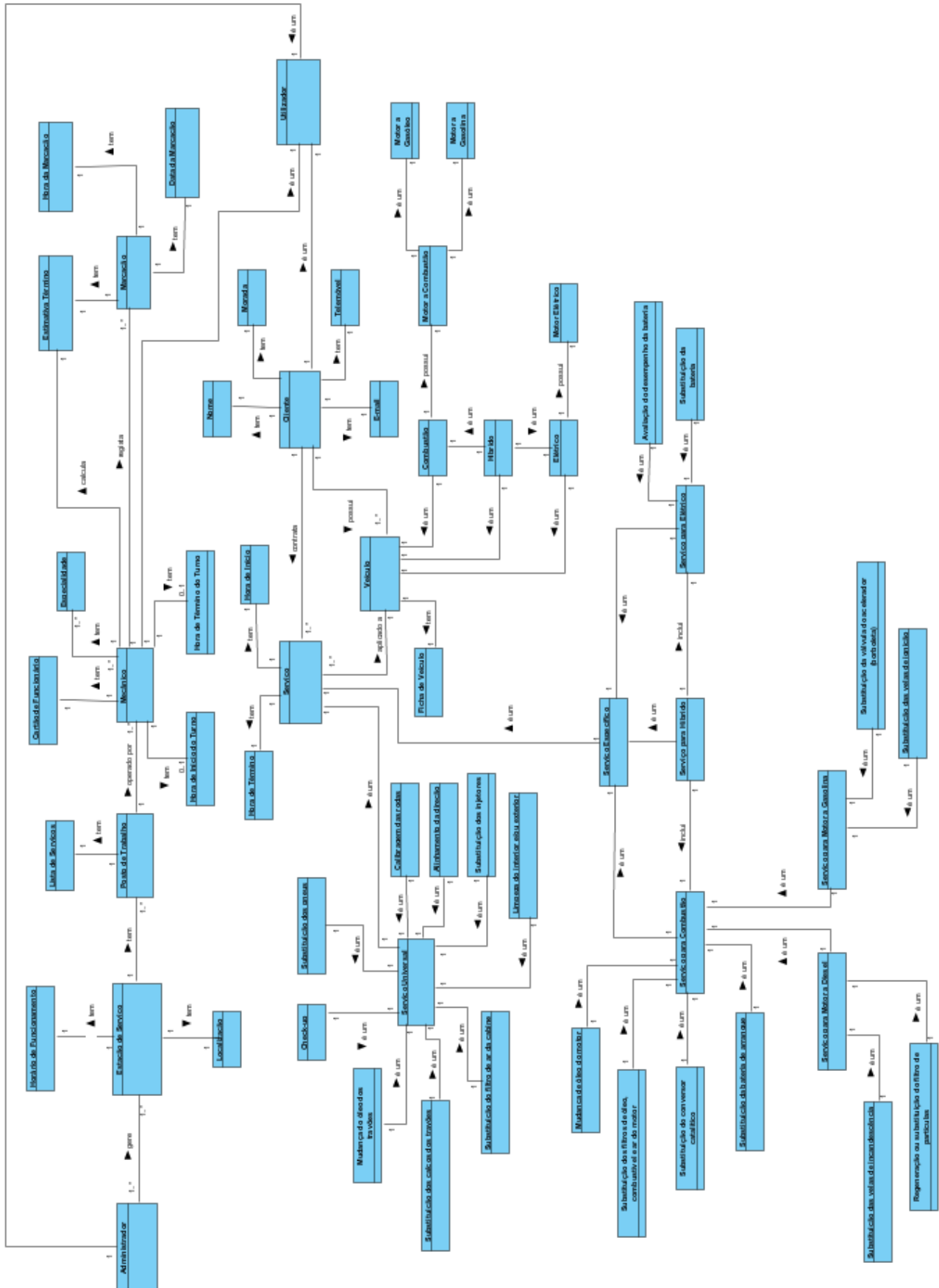
# **1.Introdução**

Este relatório técnico foi elaborado no âmbito da unidade curricular de Desenvolvimento de Sistemas de Software no qual nos foi proposta a elaboração de um sistema que permita automatizar o processo de admissão e encaminhamento de clientes dentro de uma estação de serviço.

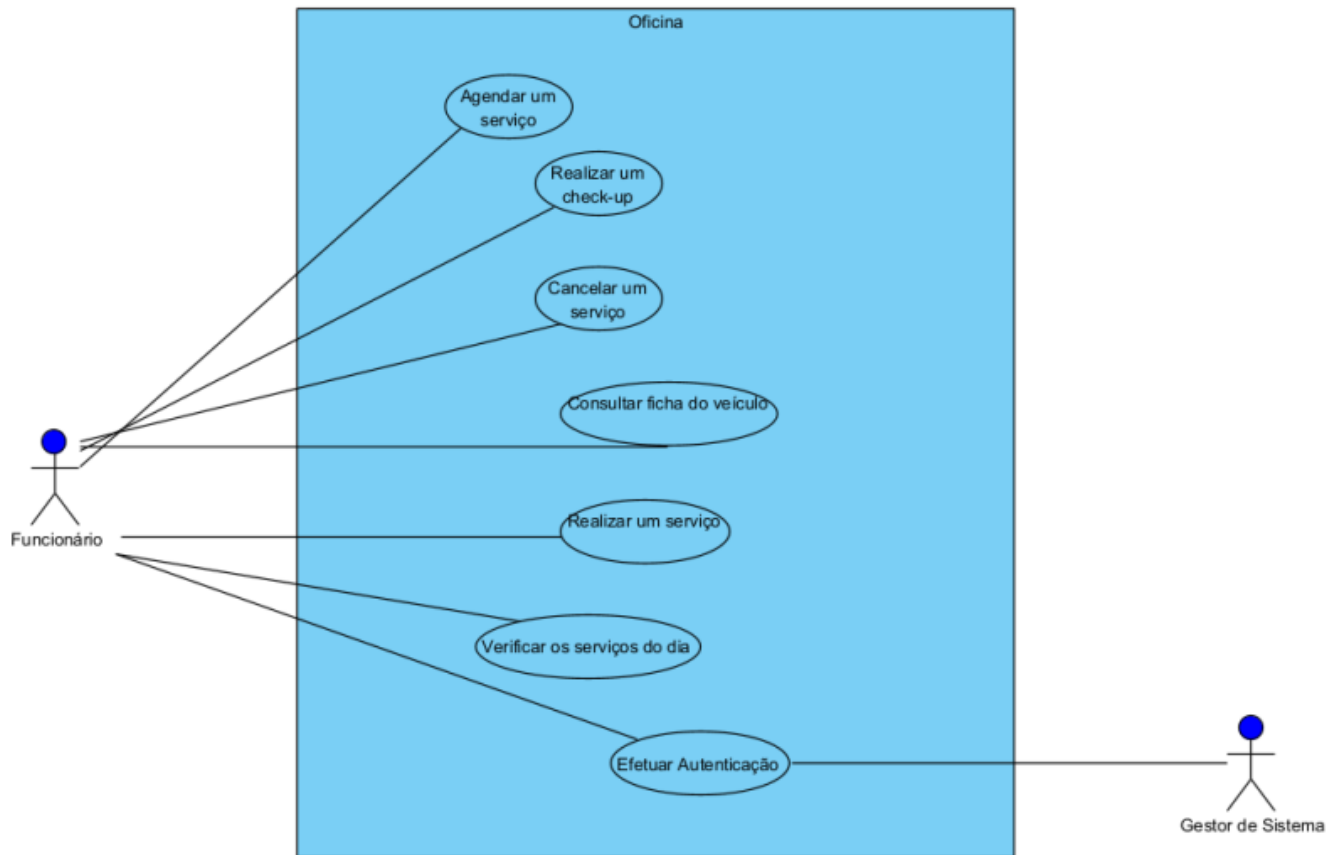
Nele estão contidos todos os Diagramas que serviram de base e sustentam a arquitetura do nosso sistema.

## 2.1 Modelo de Domínio

A imagem abaixo é referente ao modelo de domínio do nosso sistema, no entanto devido a dimensão do mesmo este não se encontra muito legível, portanto enviamos em anexo junto com este documento o ficheiro do mesmo para que possa ser analisado com mais detalhe.



## 2.2 Diagrama de use cases



## 2.3 Especificação textual dos Use Cases

### 2.3.1 Autenticação no sistema

**Use Case:** Autenticação no sistema

**Descrição:** Na oficina, um mecânico acede à aplicação usando as suas credenciais

**Pré-Condição:** True.

**Pós-Condição:** O mecânico consegue aceder ao sistema com sucesso.

**Fluxo Normal (1):**

1. Mecânico introduz as suas credenciais no sistema
2. Sistema confirma que as credenciais inseridas são válidas
3. Mecânico acede ao sistema

**Fluxo Alternativo (2): [Sistema não valida as credenciais] (passo 3):**

- 3.1 Sistema informa que as credenciais inseridas não são válidas.
- 3.2 Sistema termina o processo.

### 2.3.2 Verificar serviços do dia

**Use case:** Verificar os serviços do dia

**Descrição:** Um mecânico acede ao sistema no posto de trabalho, para verificar a lista de serviços que estão associados aquele posto nesse dia.

Para tal, insere as suas credenciais. O sistema verifica se as credenciais são válidas. A lista é apresentada, com os serviços a serem realizados e os respetivos horários.

**Pré-condição:** O mecânico tem de estar autenticado.

**Pós-condição:** O mecânico sabe que serviços tem de ser realizados naquele posto.

**Fluxo normal (1):**

1. Mecânico fornece os dados do posto de trabalho.
2. Sistema verifica que os dados fornecidos são válidos.
2. Sistema apresenta a lista de serviços.
3. Mecânico verifica a lista de serviços associados ao posto de trabalho, para esse dia.

**Fluxo de exceção (2) : [Os dados fornecidos pelo mecânico são inválidos] (passo1):**

- 1.1 Sistema informa que os dados fornecidos pelo mecânico estão incorretos
- 1.2 Sistema termina processo

### 2.3.3 Agendar um serviço

**Use case:** Agendar um serviço

**Descrição:** O cliente dirige-se à estação de serviço para marcar determinados serviços que pretende que realizem no seu veículo. Indica ao mecânico quais os serviços que deseja que sejam efetuados. O mecânico regista os serviços pretendidos na ficha do veículo e o sistema verifica as disponibilidades de datas e horas para o realizar nesse dia.

**Pré-condição:** O cliente possui um veículo que necessita de um serviço.

**Pós-condição:** Os serviços solicitados são registados no sistema, para o dia acordado.

**Fluxo normal (1):**

1. O Mecânico fornece ao sistema a matrícula do veículo em questão
2. O sistema verifica que existe um veículo registado com aquela matrícula
3. O Mecânico fornece ao sistema os serviços requisitados
4. Sistema verifica que serviço está disponível para o tipo de motor do veículo em questão
5. Sistema verifica que existem disponibilidades para esses serviços nesse dia
6. Sistema fornece as horas disponíveis para esse dia
7. Mecânico informa o sistema de qual a hora escolhida
8. Sistema verifica que existe uma ficha do veículo
9. O sistema atualiza a ficha do veículo adicionando a marcação dos serviços

**Fluxo de Exceção(2): [Serviço não disponível para o tipo de motor] (passo 5):**

- 2.1 Sistema informa que não é possível realizar o serviço no tipo de veículo pretendido.
- 2.2 O sistema termina o processo.

**Fluxo de Exceção(3): [Não existe disponibilidade para realizar o serviço] (passo 3):**

- 3.1 Sistema informa que não encontrou horas disponíveis para realizar o serviço nesse dia.
- 3.2 O sistema termina o processo.

**Fluxo de Exceção(4) : [Não foram aceites as datas apresentadas] (passo 4):**

- 4.1 Mecânico informa que não foram aceites as horas apresentadas
- 4.2 O sistema termina o processo.

**Fluxo de Exceção (5): [Não existe veículo registado com matrícula fornecida] (passo 2):**

- 7.1 Sistema informa que não existe um veículo registado com a matrícula fornecida

### 2.3.4 Realizar um serviço

**Use case:** Realizar um serviço

**Descrição:** O mecânico vai iniciar um serviço. Após a realização do mesmo, indica que o terminou e diz se o concluiu com sucesso ou não. Caso não tenha conseguido concluir, indica a razão pela qual não o fez. Após isso é atualizada a ficha do veículo indicando que este serviço foi realizado (pode também indicar que o serviço não pode ser feito e o motivo).

**Pré-condição:** Tem de existir uma marcação para esse serviço e tem de existir uma ficha do veículo.

O mecânico tem de estar autenticado.

**Pós-condição:** Ficha do veículo atualizada refletindo a realização ou não do serviço.

**Fluxo normal (1):**

1. Mecânico inicia o serviço
2. Mecânico executa o serviço
3. Mecânico informa o sistema que terminou o serviço
4. Mecânico fornece os dados do serviço ao sistema (realizou o serviço com sucesso ou não)
5. Sistema atualiza a ficha do veículo, indicando que o serviço foi realizado com sucesso

**Fluxo Alternativo: [Não foi possível realizar o serviço] (passo 4):**

- 4.1 Mecânico informa o sistema de que não foi possível realizar o serviço e o motivo.
- 4.2 Sistema atualiza a ficha do veículo, indicando que não foi possível realizar o serviço e o motivo.

### 2.3.5 Realizar um checkup

**Use case:** Realizar um checkup

**Descrição:** O mecânico realiza um check-up num veículo de modo a identificar os serviços necessários.

**Pré-condição:** Tem de existir uma marcação para esse serviço.

Tem de existir uma ficha do veículo.

O mecânico tem de estar autenticado.

**Pós-condição:** Ficha do veículo atualizada, refletindo os serviços necessários.

**Fluxo Normal:**

1. Mecânico inicia o check-up
2. Mecânico realiza o check-up ao veículo
3. Mecânico indica os serviços que o veículo precisa
4. Mecânico informa o sistema que terminou o check-up
5. Mecânico fornece os dados do serviço ao sistema (realizou o serviço com sucesso ou não)
6. Sistema atualiza a ficha do veículo, registando os serviços que o veículo necessitará.



### 2.3.6 Enviar notificação para recolha do veículo

**Use case:** Enviar notificação para recolha

**Descrição:** Após o término de todos os serviços marcados para um determinado veículo, o sistema envia uma mensagem ao cliente informando-o de que o veículo se encontra pronto para recolha.

**Pré-condição:** Mecânico tem de estar autenticado no sistema e a ficha veículo tem de estar atualizada.

**Pós-condição:** A notificação de recolha foi enviada para o cliente.

**Fluxo normal:**

1. O mecânico informa o sistema que terminou um serviço.
2. Sistema verifica que não existem mais serviços agendados para o veículo.
3. Sistema envia uma notificação ao cliente informando-o de que o veículo se encontra disponível para recolha.

**Fluxo de exceção: [Ainda existem serviços agendados] (passo 2)**

- 2.1.O sistema não envia a notificação.
- 2.2.O sistema termina o processo.

### 2.3.7 Cancelar um serviço

**Use case:** Cancelar um serviço

**Descrição:** Um cliente dirige-se à oficina e pede ao mecânico que cancele um serviço anteriormente marcado. O mecânico acede ao sistema e remove o serviço em questão da ficha do veículo.

**Pré-condição:** Existir um serviço marcado para o veículo do cliente que ainda não tenha sido começado.

Existir ficha do veículo em questão

Mecânico tem de estar autenticado

**Pós-condição:** O serviço foi removido da ficha do veículo.

**Fluxo normal:**

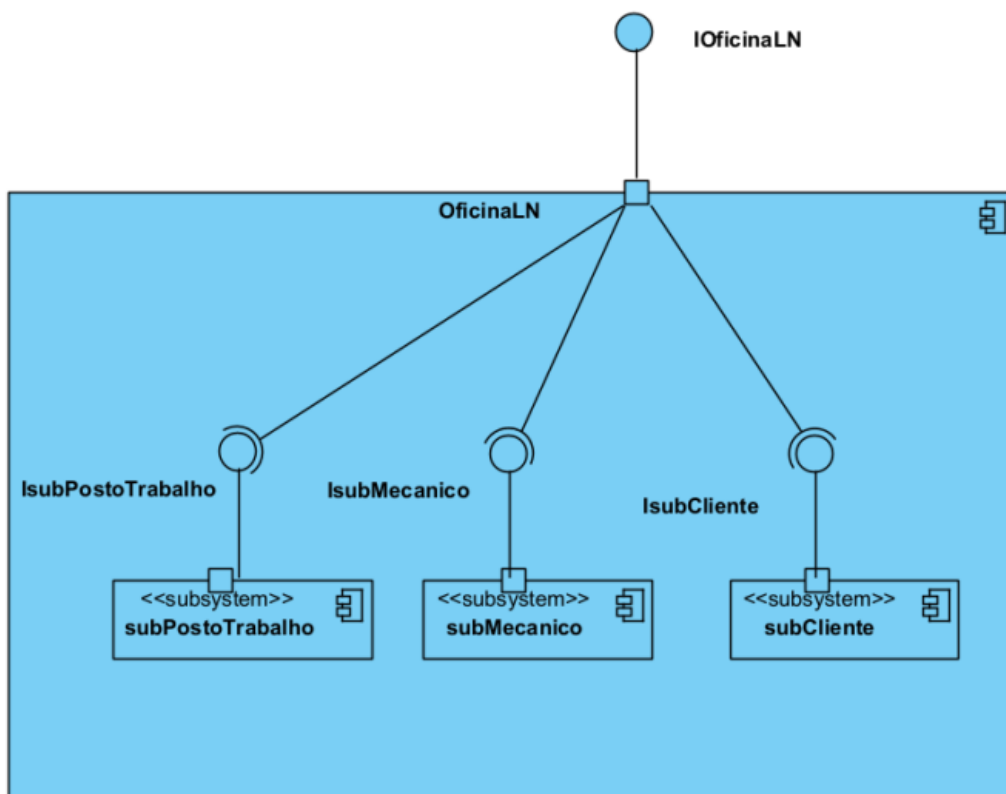
1. Mecânico fornece ao sistema os dados do veículo em questão
2. Mecânico acede à ficha do veículo do cliente em questão
3. Mecânico fornece ao sistema os serviços a desmarcar
4. Sistema atualiza a ficha do veículo, removendo os serviços em questão

**Fluxo de exceção:** [Desmarcação falhou] (passo 4)

- 4.1. Sistema informa que falhou a desmarcar o serviço.
- 4.2. Sistema termina processo.

## 2.4 Diagrama de Componentes

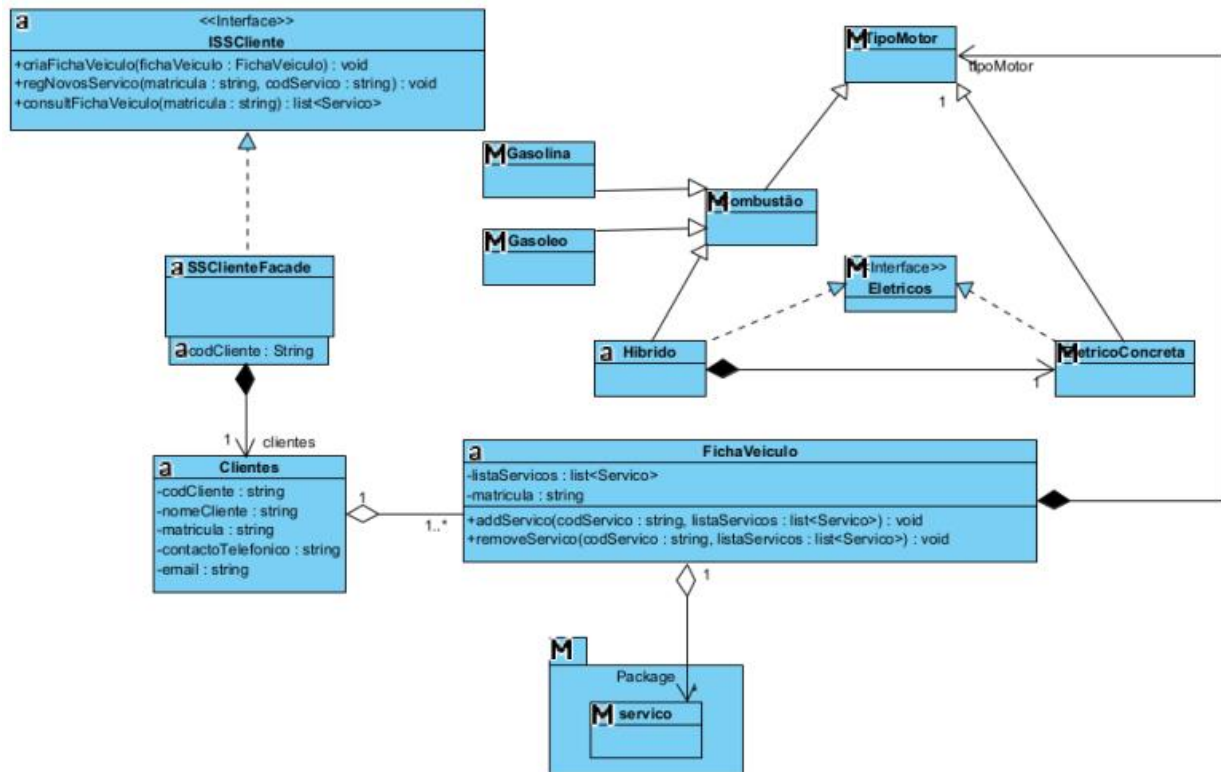
Tendo os use cases acima definidos, foi-nos possível dividi-los em subsistemas e chegar ao seguinte diagrama de componentes:



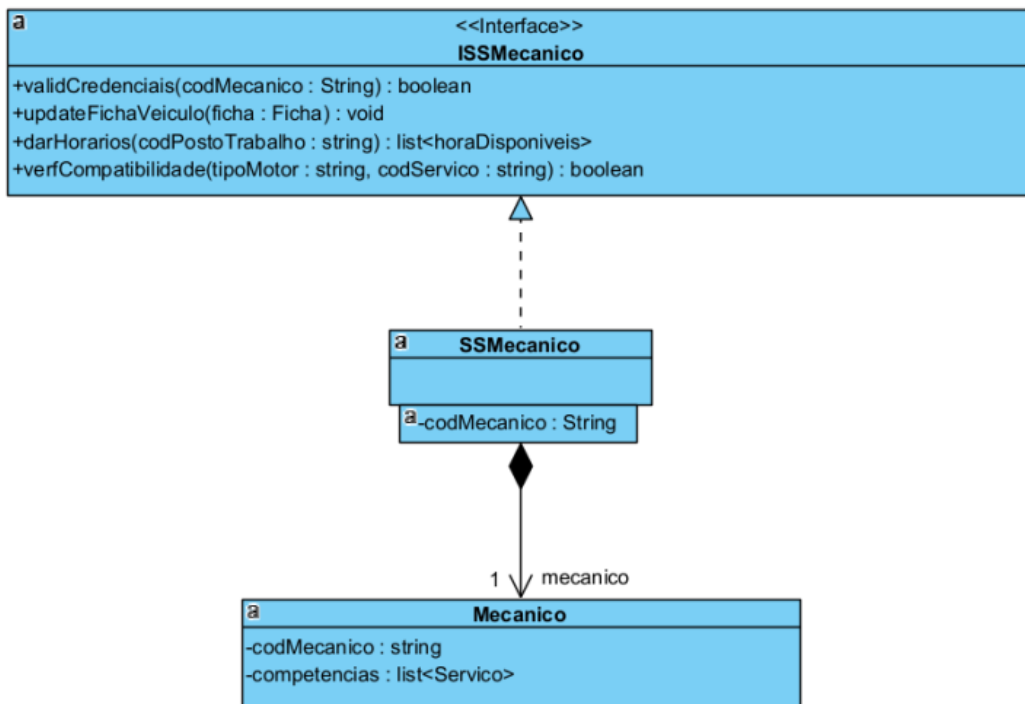
## 2.5 Diagrama de Classes

De seguida, temos os diagramas de classes desenvolvidos para cada subsistema do programa:

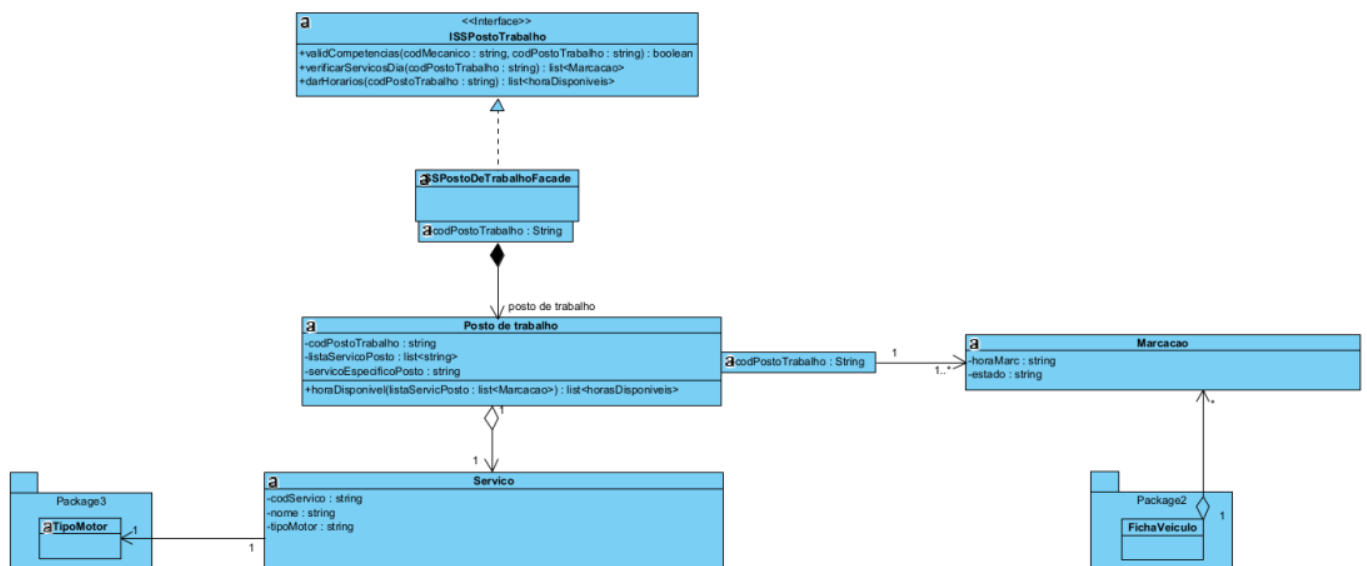
### 2.5.1 Diagrama de classes Cliente



## 2.5.2 Diagrama de classes Mecânico

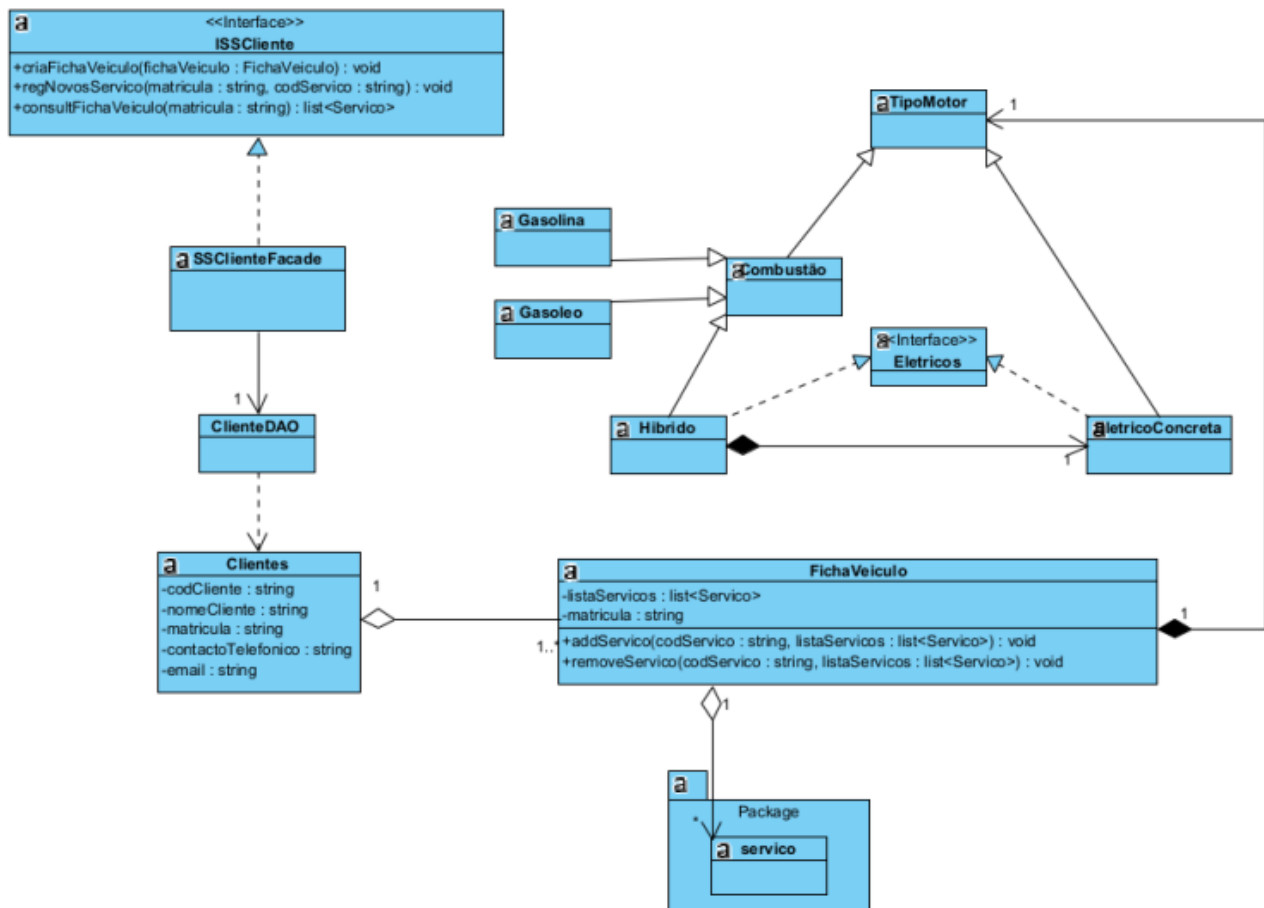


## 2.5.3 Diagrama de classes Posto de Trabalho

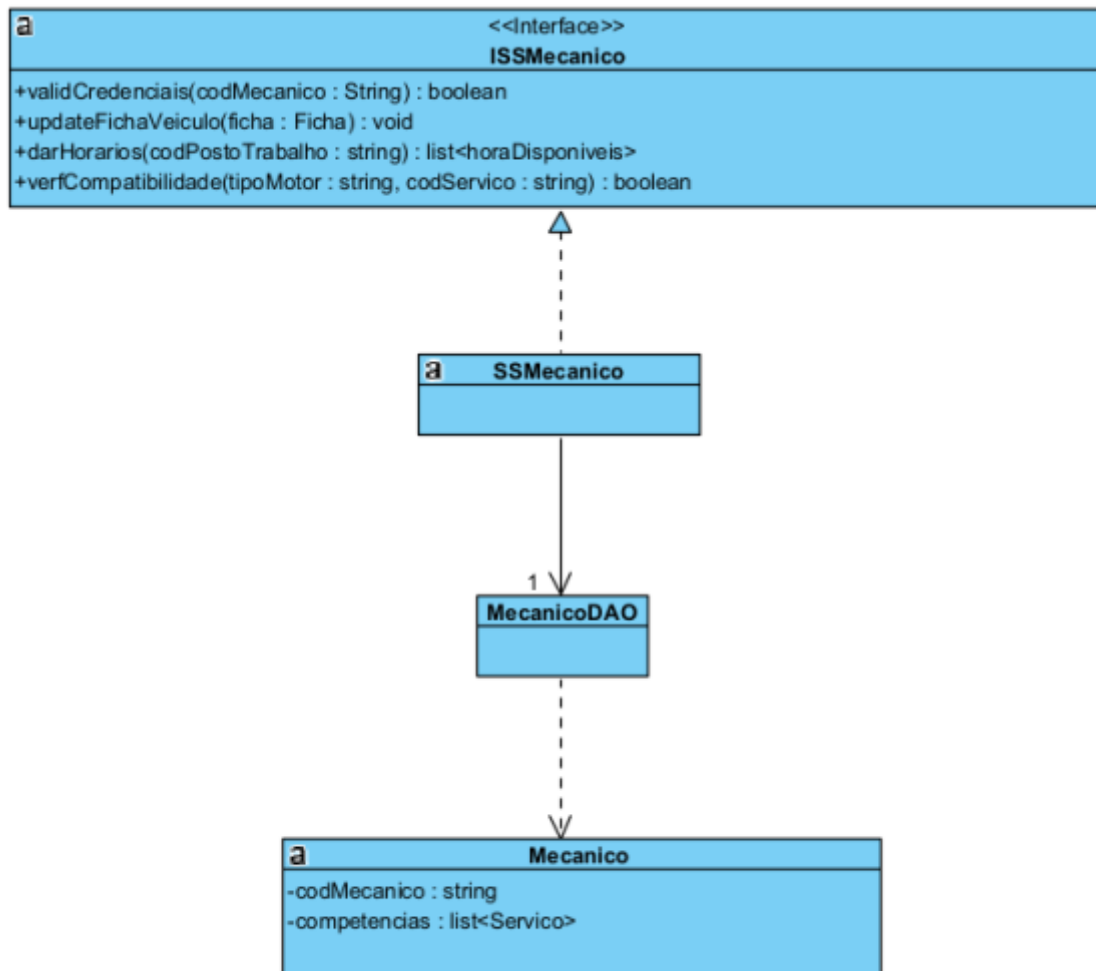


## 2.6 Diagramas de classes com DAO'S

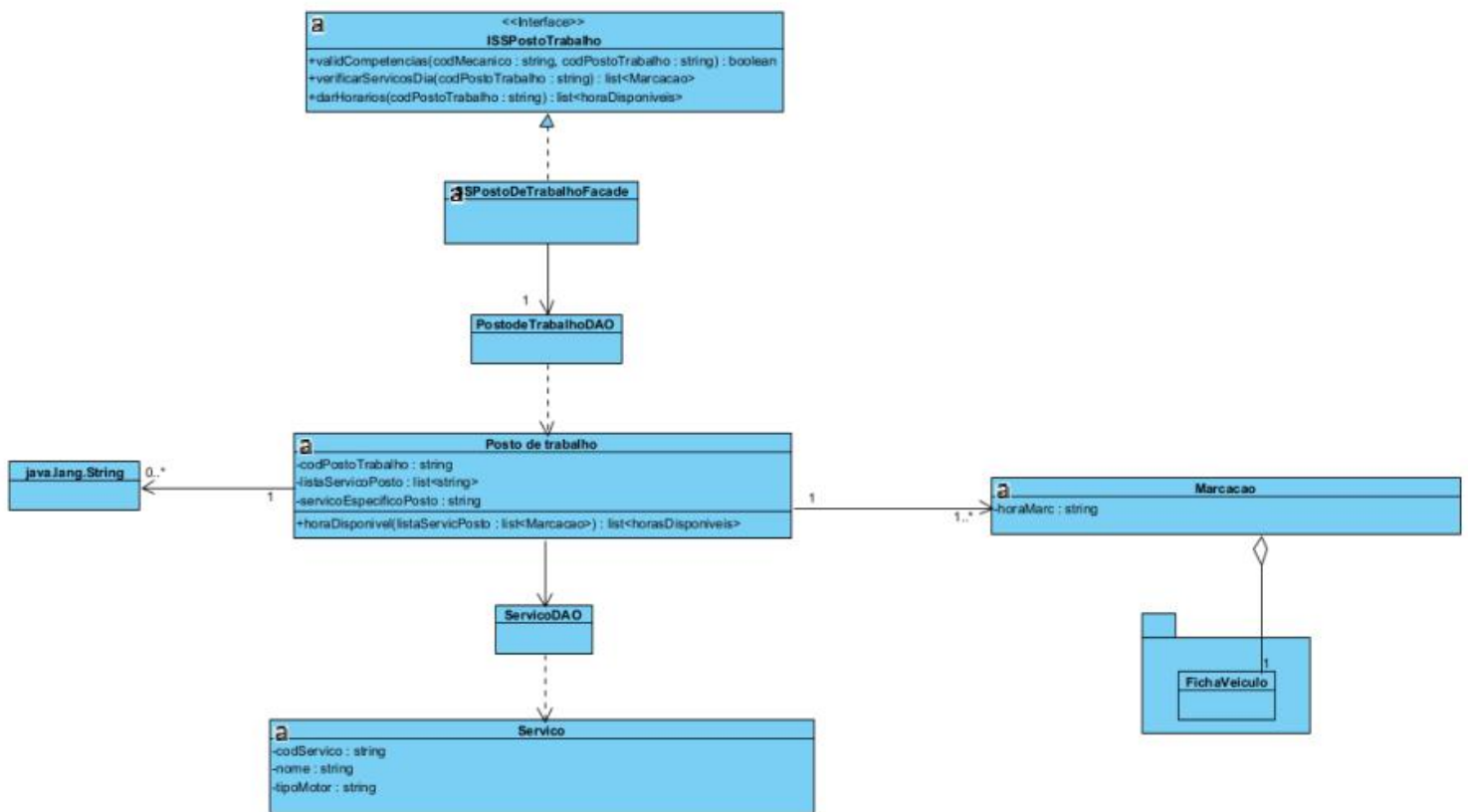
### 2.6.1 Diagrama de classes Cliente com DAO'S



### 2.6.2 Diagrama de classes Mecânico com DAO'S



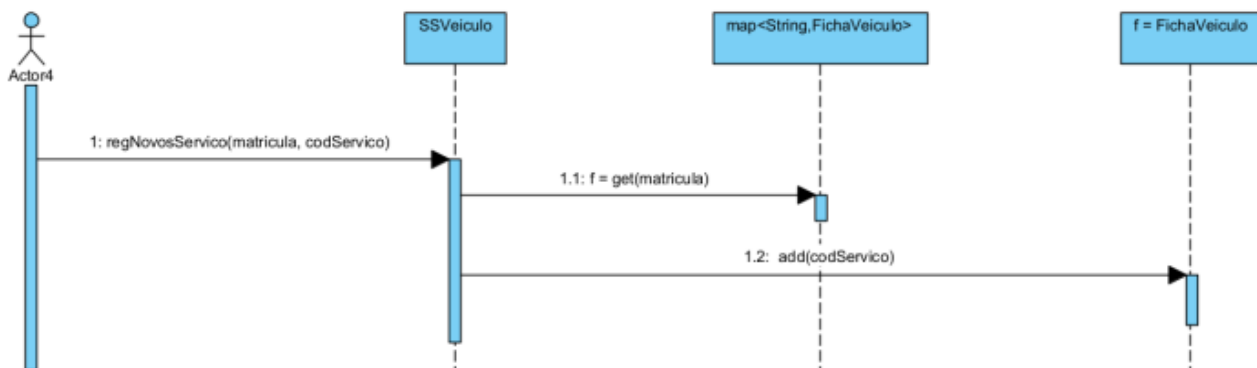
### 2.6.3 Diagrama de classes PostoDeTrabalho com DAO'S



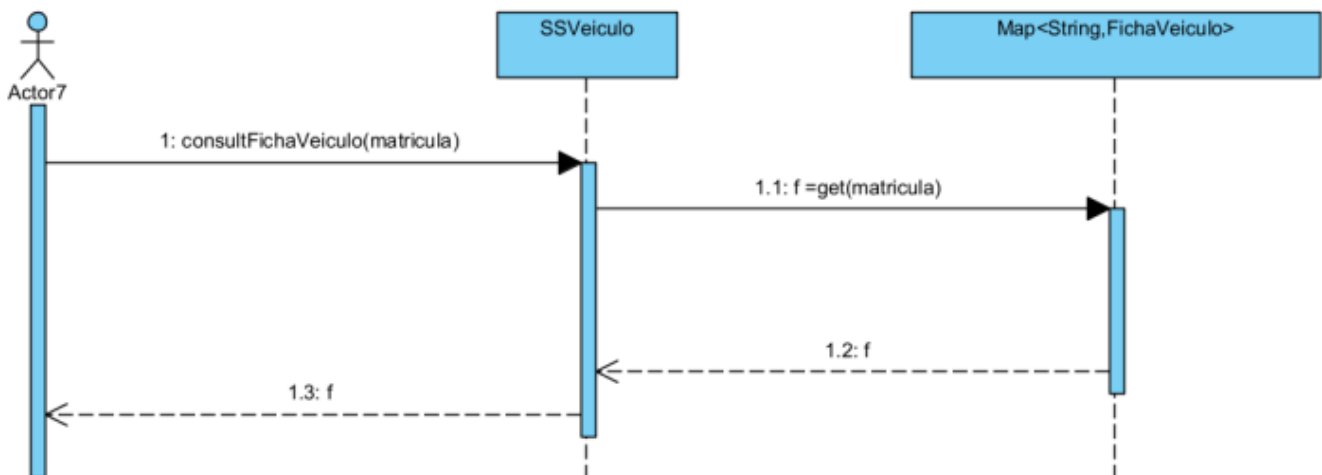


## 2.7 Diagramas de Sequência

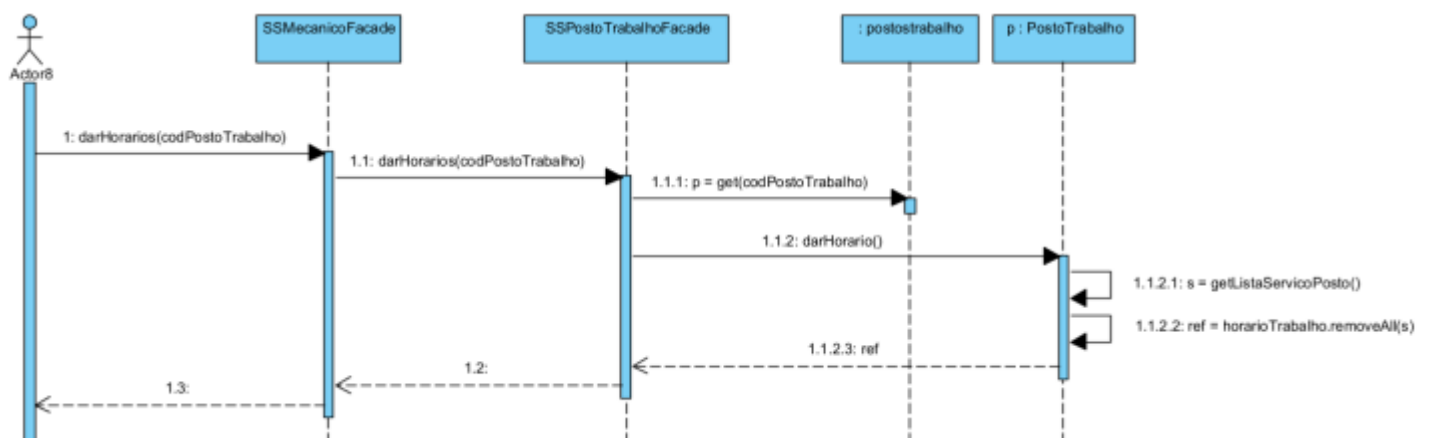
### 2.7.1 regNovosServico



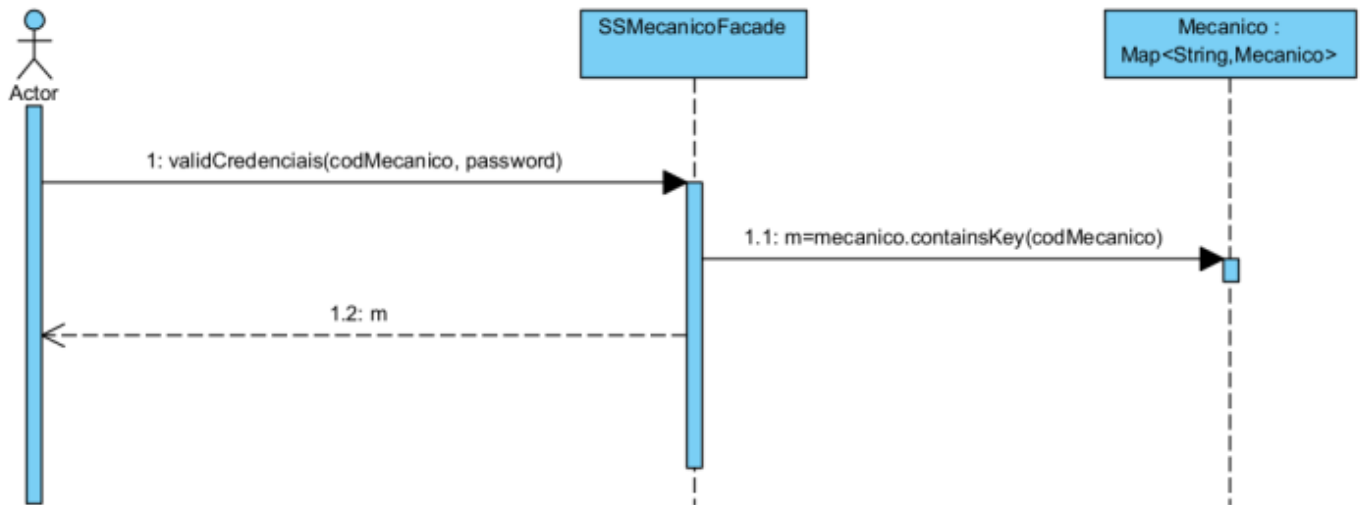
### 2.7.2 consultFichaVeiculo



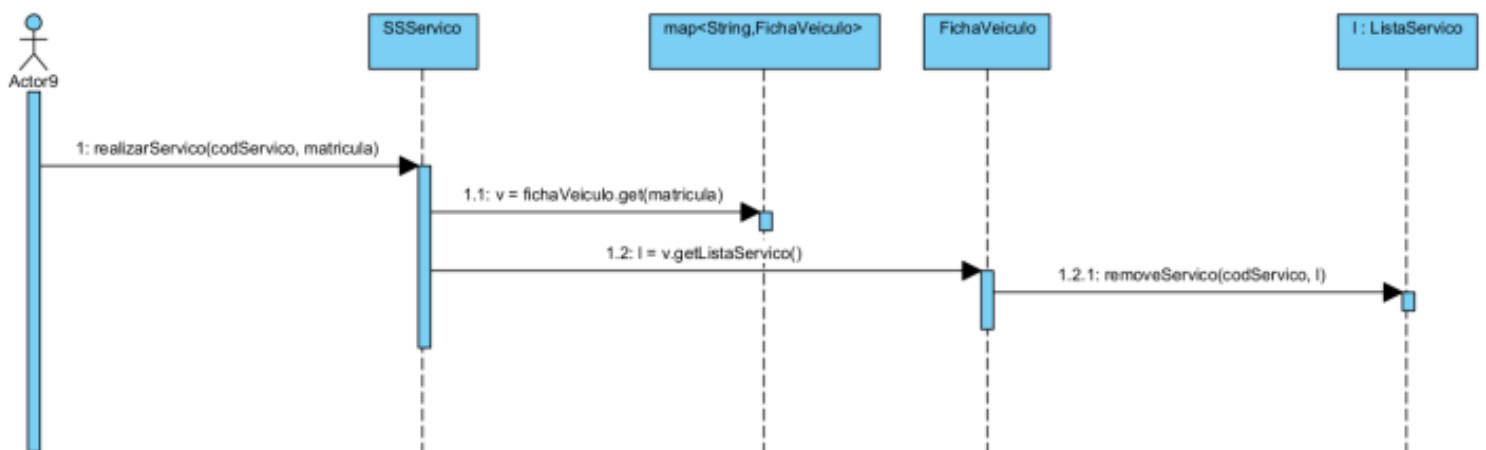
### 2.7.4 darHorarios



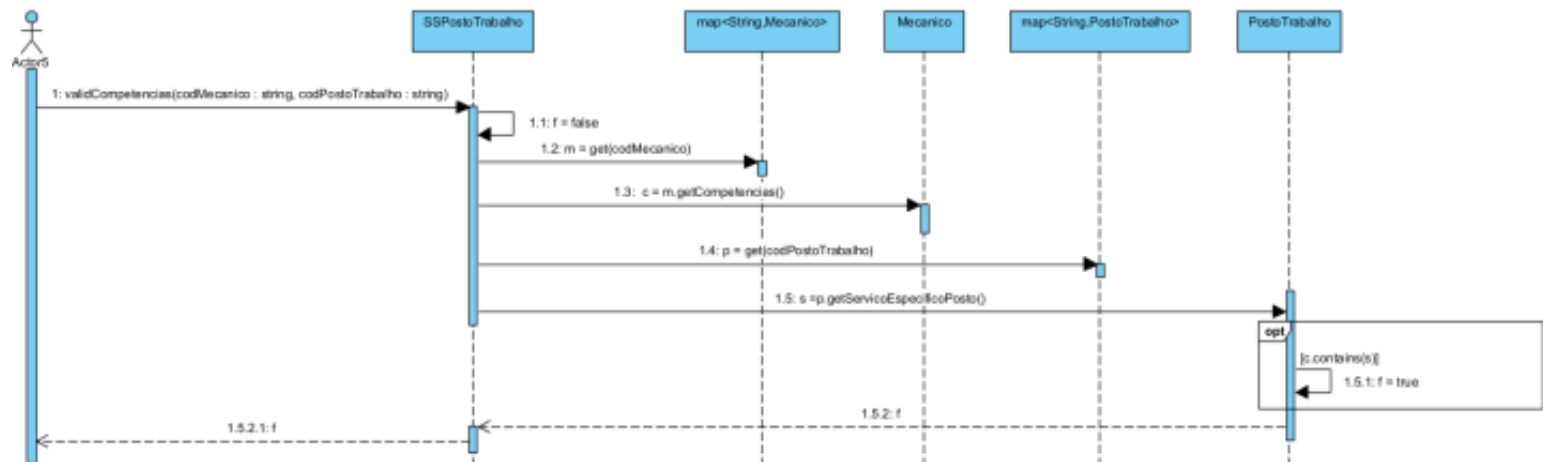
### 2.7.5 validCredenciais



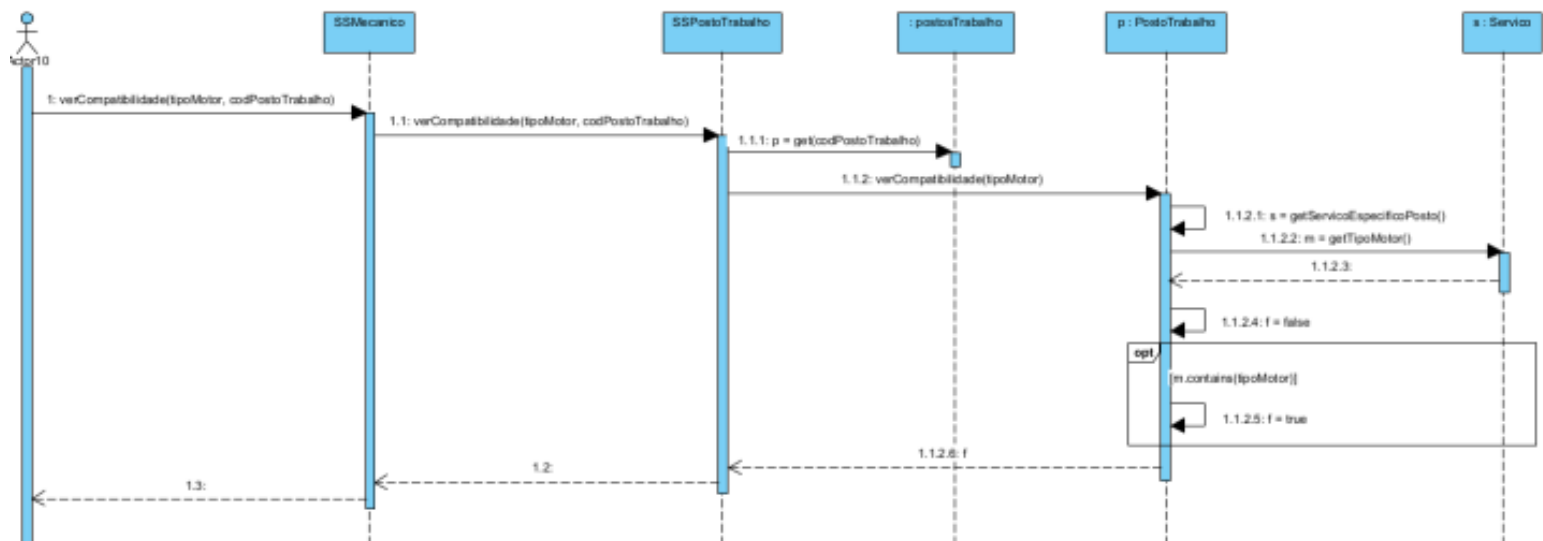
### 2.7.6 realizarServico



## 2.7.7 validCompetencias



## 2.7.8 verCompatibilidade



## 2.8 Conclusão e Análise dos Resultados Obtidos

Numa primeira fase começamos por elaborar o Modelo de Domínio e o Modelo de Use Cases. Sendo que ambos são o resultado da interpretação que fizemos do problema que nos foi proposto, abrangendo apenas os cenários que nos foram pedidos para ter em consideração.

Através do primeiro modelo, obtivemos uma representação visual das entidades relacionadas com o nosso problema, e através do segundo conseguimos descrever as funcionalidades que pretendemos que o nosso sistema suporte, representando os requisitos funcionais que identificamos com base no problema.

De seguida, tendo os use cases definidos, construímos o nosso Diagrama de Componentes (fornecendo uma visão física da arquitetura do sistema), os Diagramas de Classes (ilustrando os relacionamentos entre as classes do nosso sistema), inicialmente sem DAO'S, e os diagramas de sequência dos métodos mais importantes que iríamos ter de implementar (especificam como funcionam as operações do nosso sistema).

Posteriormente, procedemos à alteração dos diagramas de classes para que passassem a incorporar os DAO'S (tal como se pode ver na secção 2.6 deste relatório) de que viríamos a necessitar para lidar com os acessos e trocas de informação com a base de dados que dá suporte ao nosso programa. Acrescento ainda que a incorporação destes mesmos DAO'S também poderia ser realizada nos diagramas de sequência, porém, como as alterações eram mínimas acabamos por não o fazer, focando-nos mais nos de classes.

Tendo concluído todo o planeamento do projeto com todos os diagramas desenvolvidos, já tínhamos o modelo comportamental de como o nosso sistema deve funcionar e ainda as funcionalidades que pretenderíamos implementar no mesmo.

Portanto, passámos para a fase de implementação do projeto. Esta fase tornou-se num desafio ainda maior após a desistência de dois colegas do grupo, pois isto, para além de nos fazer perder tempo, obrigou-nos a reorganizar todo o planeamento efetuado a contar com eles. Decidimos, então, focar-nos na realização das tarefas essenciais como, marcação de serviços e realização dos mesmos por parte do mecânico

Apesar disto, fomos desenvolvendo o nosso programa da melhor forma que nos foi possível implementando os DAO's e os métodos presentes no nosso Diagrama de Classes.

Em suma, apesar do nosso grupo ter sido reduzido a apenas três elementos e ainda não possuir todos os requisitos pedidos pelo enunciado do trabalho, fomos fiéis ao nosso planeamento, seguimos o nosso modelo lógico e comportamental, implementamos as funcionalidades que consideramos de maior importância dentro do contexto de gestão de uma oficina e conseguimos atingir um sistema funcional, que poderia ser melhorado se a situação excecional que aconteceu como grupo de trabalho não se tivesse verificado.