



Universidade do Minho

Licenciatura em Engenharia Informática

# Desenvolvimento de Sistemas de Software

## E.S. Ideal

### Grupo 35

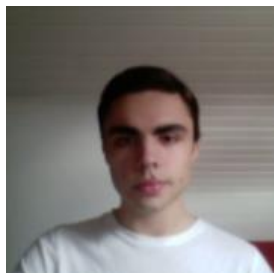
Guilherme Rio (a100898)



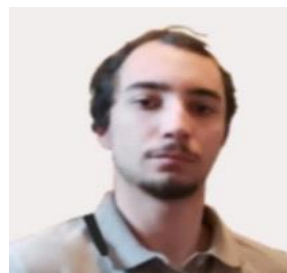
Rui Cerqueira (a100537)



Tomás Valente (a10054)



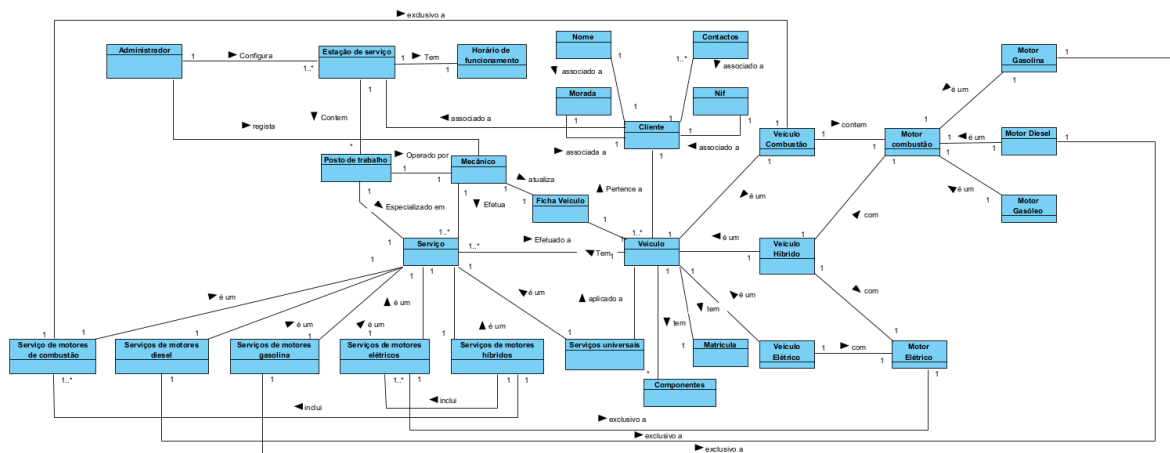
Diogo Cunha (a100481)



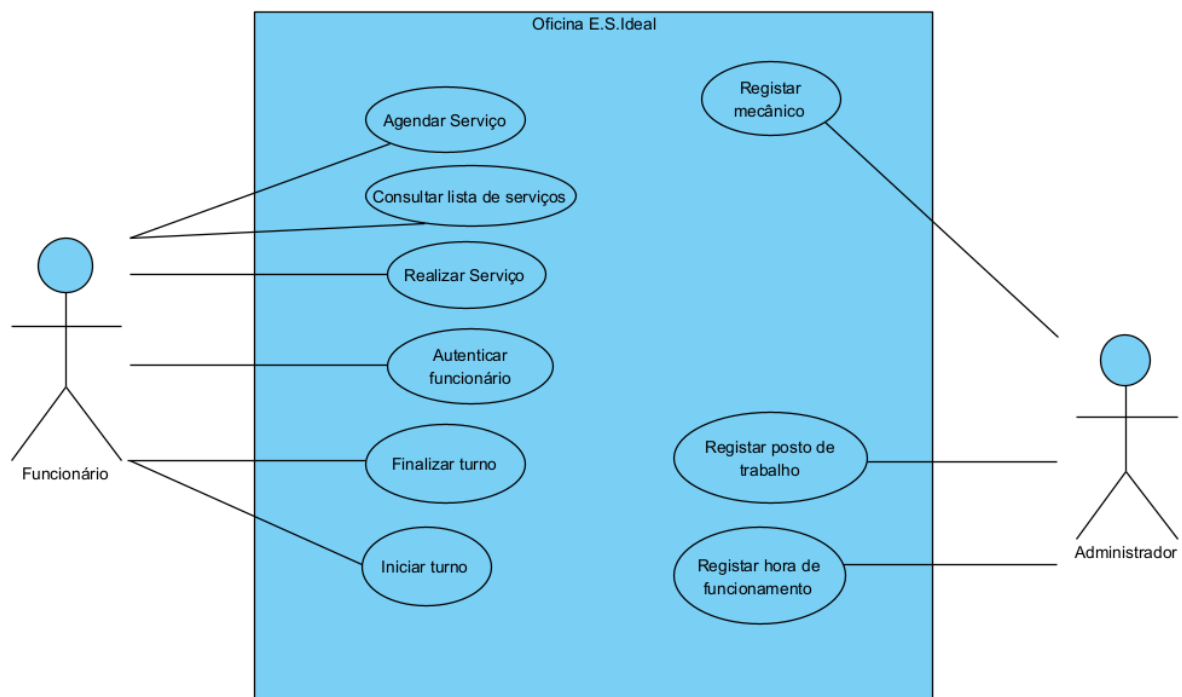
Nuno Aguiar (a100480)

Repositório: <https://github.com/LEI-DSS/trabalho-dss-grupo-35>

## 1. Modelo de Domínio



## 2. Diagrama de Use Case



### 3. Descrições dos Use Case

#### USE CASE – Registrar Mecânico

USE CASE:	Registrar mecânico
DESCRIÇÃO:	Processo de registo de todos os mecânicos que trabalham na estação de serviço.
CENÁRIOS:	A Diana vai à nova estação de serviço da E.S.Ideal em Gualtar instalar o sistema, inclusive registrar os mecânicos da nova estação de serviço.
PRÉ-CONDIÇÃO:	True
PÓS-CONDIÇÃO:	Mecânico registado na estação de serviço
FLUXO NORMAL:	1. Administrador insere o nome do mecânico 2. Administrador insere o posto em que o mecânico irá trabalhar 3. O sistema gera um número de identificação para o mecânico e emite-lhe um cartão de funcionário

#### USE CASE – Registrar Posto de Trabalho

USE CASE:	Registrar posto de trabalho
DESCRIÇÃO:	Processo de registo de todos os postos de trabalho existentes na estação de serviço
CENÁRIOS:	A Diana vai à nova estação de serviço da E.S.Ideal em Gualtar instalar o sistema, inclusive registrar os postos de trabalhos existentes, definindo para cada um o tipo de serviço que pode realizar.
PRÉ-CONDIÇÃO:	True
PÓS-CONDIÇÃO:	Posto de trabalho registado
FLUXO NORMAL:	1. Administrador insere o nome do posto de trabalho 2. Administrador insere o tipo de serviço a ser praticado no posto 3. Sistema regista o posto de trabalho
FLUXO DE EXCEÇÃO	(1)[Existe posto de trabalho com o mesmo nome](passo 1)  1.1 Sistema avisa que já existe posto de trabalho com o nome proposto e termina a operação

#### USE CASE – Registrar Hora de Funcionamento

USE CASE:	Registrar hora de funcionamento
DESCRIÇÃO:	Processo de registo do horário de funcionamento na estação de serviço
CENÁRIOS:	A Diana vai à nova estação de serviço da E.S.Ideal em Gualtar instalar o sistema, inclusive registrar o horário de funcionamento
PRÉ-CONDIÇÃO:	True
PÓS-CONDIÇÃO:	O horário de funcionamento da estação fica registado
FLUXO NORMAL:	1. Administrador insere os dias em que a estação de serviço está aberta 2. Administrador insere os dias em que a estação de serviço está aberta 3. Sistema regista horario de funcionamento

#### USE CASE – Autenticar Funcionário

USE CASE:	Autenticar Funcionário
DESCRIÇÃO:	O funcionário faz login no sistema
CENÁRIOS:	O Diogo entra ao serviço usando o seu cartão de funcionário no posto em que vai trabalhar
PRÉ-CONDIÇÃO:	Funcionario está registado no sistema
PÓS-CONDIÇÃO:	O funcionário está autenticado
FLUXO NORMAL:	1. Funcionario insere o seu cartão para entrar no sistema 2. Sistema verifica as competências do funcionário para trabalhar naquele posto
FLUXO DE EXCEÇÃO	(1) [falta de competências](passo 2)  2.1 Sistema não valida as competências e termina a operação

## USE CASE – Iniciar Turno

<b>USE CASE:</b>	Iniciar turno
<b>DESCRIÇÃO:</b>	Registo do início do turno de um funcionário
<b>CENÁRIOS:</b>	O sistema verifica que o Diogo tem competências para trabalhar naquele posto e regista a hora de início do turno
<b>PRÉ-CONDIÇÃO:</b>	Funcionario está autenticado no sistema
<b>PÓS-CONDIÇÃO:</b>	O sistema regista o início do turno
<b>FLUXO NORMAL:</b>	1. Sistema regista o início do turno do funcionário e a data de entrada

## USE CASE – Agendar Serviço

<b>USE CASE:</b>	Agendar Serviço
<b>DESCRIÇÃO:</b>	Cliente agenda o serviço que pretende que seja realizado
<b>CENÁRIOS:</b>	A Maria leva o carro à estação de serviço e pede um check-up; O Manuel vai à estação de serviço e pede uma lavagem
<b>PRÉ-CONDIÇÃO:</b>	O cliente está registado na estação em que vai agendar o serviço
<b>PÓS-CONDIÇÃO:</b>	O serviço fica agendado
<b>FLUXO NORMAL:</b>	1. Funcionário regista o pedido de serviço no sistema 2. O funcionário analisa no sistema os serviços previstos 3. A ficha do veículo é atualizada para assinalar a necessidade do serviço 4. O sistema calcula a ordem de serviço necessária à execução dos trabalhos 5. O sistema calcula a data do serviço em função das disponibilidades dos postos de serviço 6. O sistema informa o funcionário da data em que ficaria agendado o serviço 7. O serviço é agendado
<b>FLUXO DE EXCEÇÃO</b>	(1)[Cliente não quer agendar naquela data](passo 6) 6.1 Cliente não pretende agendar na data apresentada

## USE CASE – Consultar Lista de Serviços

<b>USE CASE:</b>	Consultar lista de serviços
<b>DESCRIÇÃO:</b>	Mecânico visualiza a lista de serviços a realizar
<b>CENÁRIOS:</b>	O Diogo verifica no sistema qual a lista de serviços que lhe estão atribuídos
<b>PRÉ-CONDIÇÃO:</b>	Mecânico autenticado
<b>PÓS-CONDIÇÃO:</b>	Lista de serviços em display
<b>FLUXO NORMAL:</b>	1. Funcionário seleciona a opção de consultar serviços 2. Sistema procura lista de serviços a realizar 3. Sistema apresenta lista de serviços

## USE CASE – Realizar Serviço

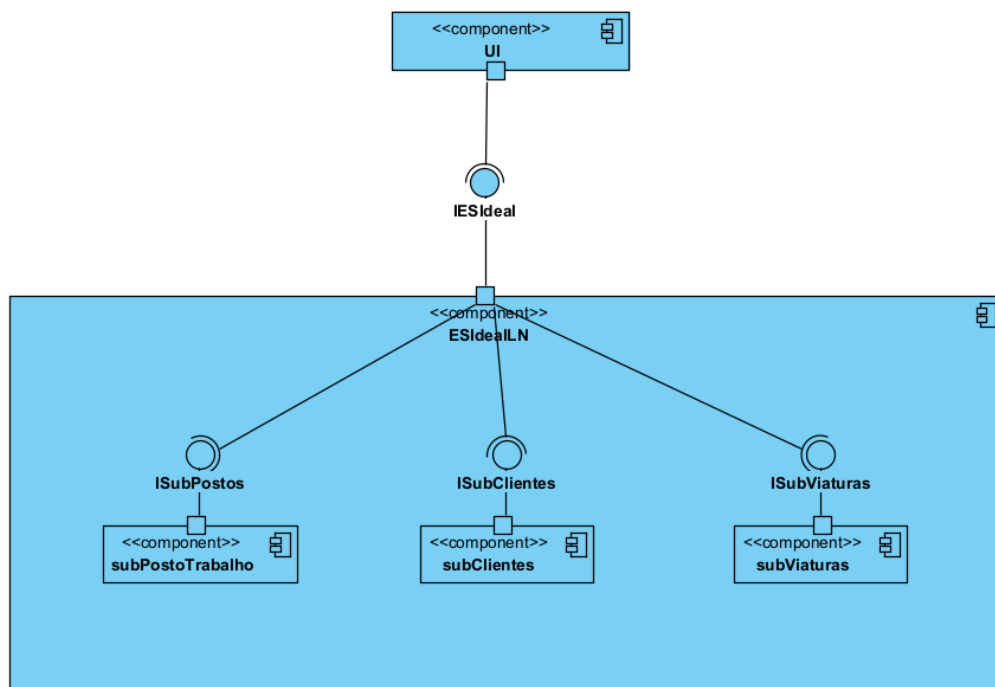
<b>USE CASE:</b>	Realizar Serviço
<b>DESCRIÇÃO:</b>	Funcionário realiza serviço requisitado
<b>CENÁRIOS:</b>	A Maria leva o carro à estação de serviço e pede um check-up; O Manuel vai à estação de serviço e pede uma lavagem
<b>PRÉ-CONDIÇÃO:</b>	O cliente está registado na estação em que vai ser realizado o serviço assim como o funcionário
<b>PÓS-CONDIÇÃO:</b>	O serviço é concluído
<b>FLUXO NORMAL:</b>	1. O funcionário sinaliza o início do serviço 2. O funcionário sinaliza a conclusão do serviço 3. O funcionário atualiza a ficha do veículo para indicar que o serviço está feito
<b>FLUXO ALTERNATIVO</b>	(1) [Cliente decide abandonar a estação de serviço enquanto o serviço é realizado](passo 3) 3.1 Sistema notifica o cliente da conclusão do serviço através de SMS 3.2 Regressa ao passo (4)
<b>FLUXO DE EXCEÇÃO</b>	(2) [Serviço não foi realizado](passo 3) 3.1 Funcionário escreve no sistema o motivo pelo qual o serviço não foi realizado 3.2 Funcionário atualiza a ficha do veículo para indicar que o serviço não está feito

## USE CASE – Finalizar Turno

USE CASE:	Finalizar turno
DESCRIÇÃO:	Registo do fim do turno de um funcionário
CENÁRIOS:	Diogo utiliza novamente o cartão para registar o fim do seu turno de trabalho e o sistema regista a hora.
PRÉ-CONDIÇÃO:	Funcionario está autenticado no sistema
PÓS-CONDIÇÃO:	O sistema regista o fim do turno
FLUXO NORMAL:	1. Funcionário introduz o seu cartão no sistema 2. Sistema verifica cartão de funcionário 3. Sistema regista o final do turno e a hora de saída do funcionário

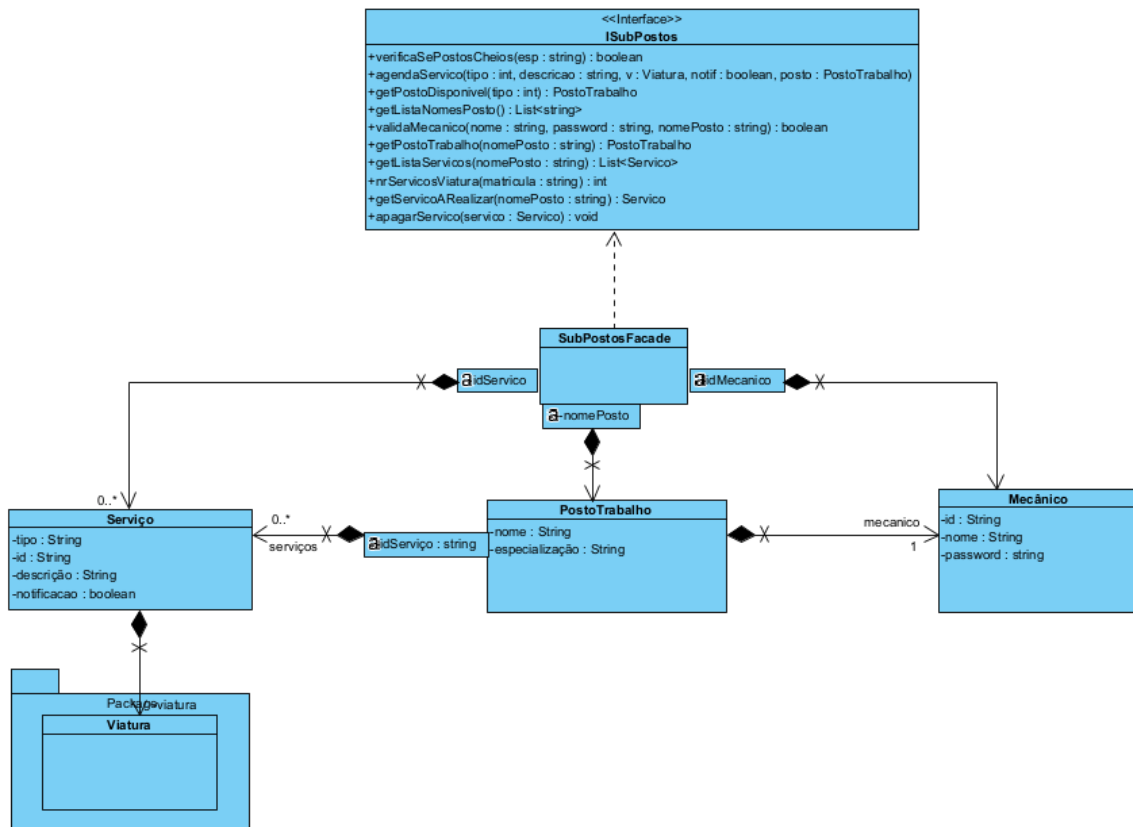
## 4. Modelação conceptual

### 4.1. Diagrama de Componentes

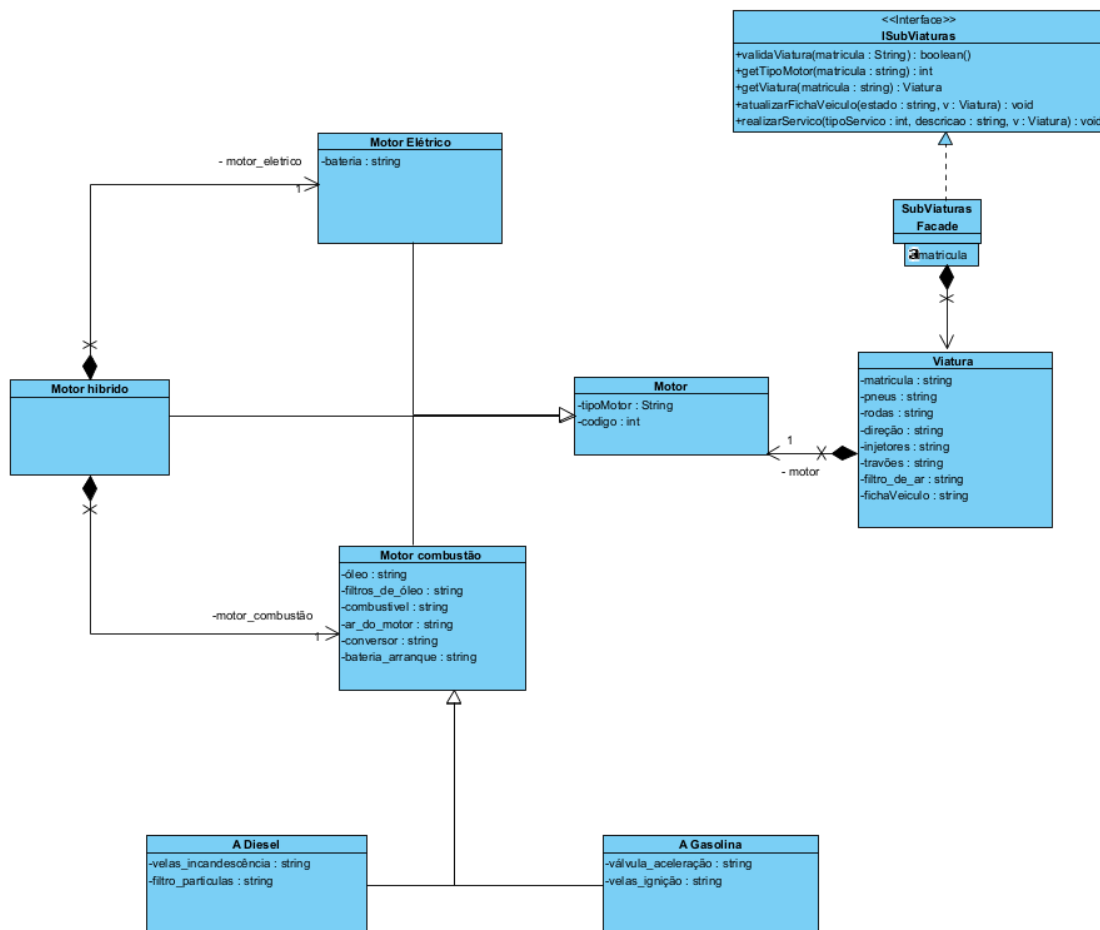


Decidimos dividir as responsabilidades do sistema em 3 subsistemas: SubPostos, SubViaturas e SubClientes.

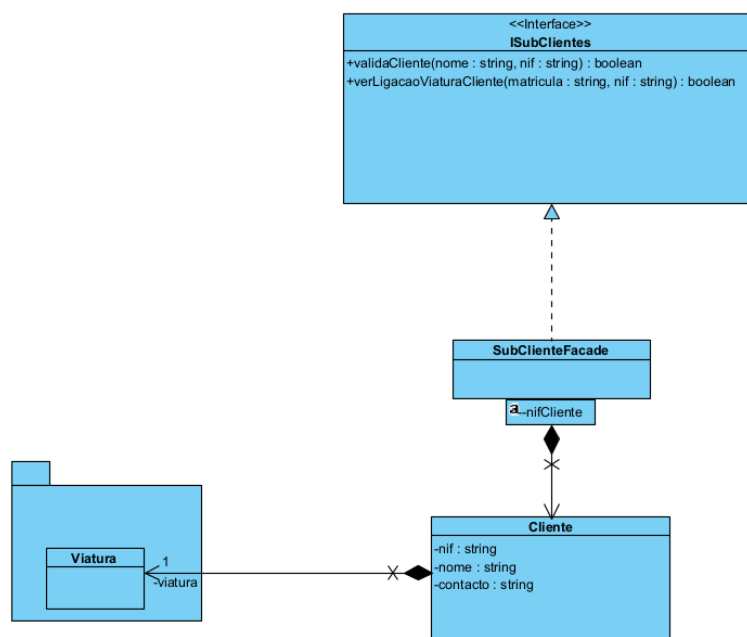
## 4.2. Diagrama de Classes de SubPostos



### 4.3. Diagrama de Classes de SubViaturas

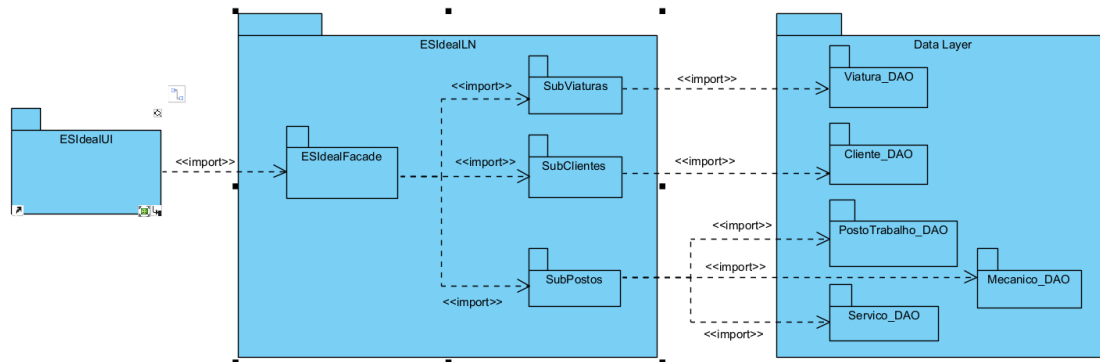


### 4.4. Diagrama de Classes de SubClientes

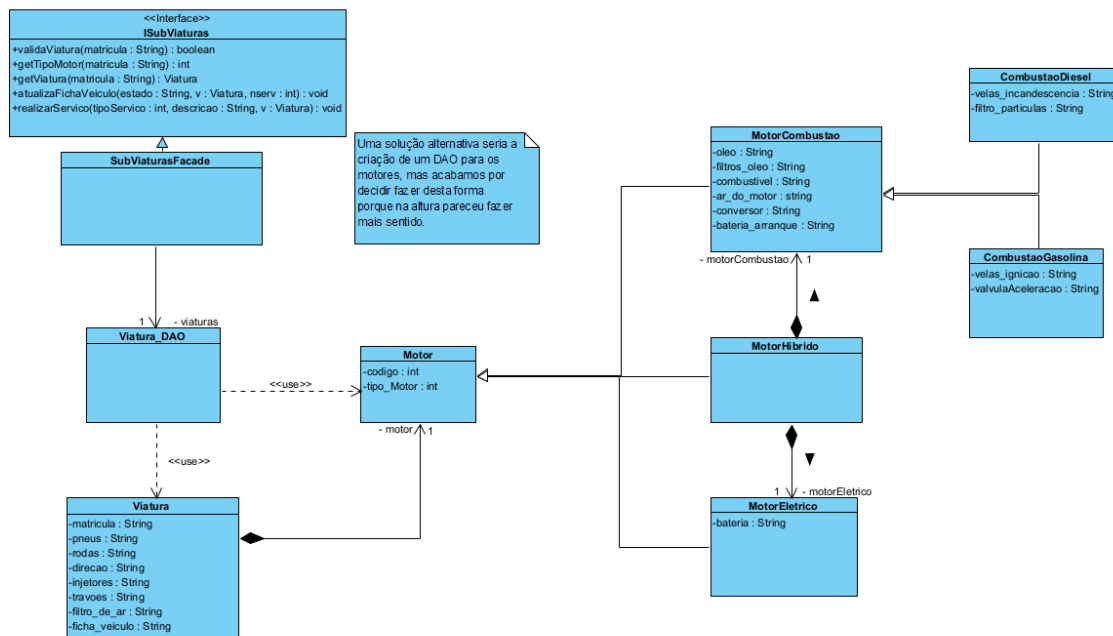


## 5. Solução Efetiva (Implementação de DAOs)

### 5.1. Diagrama de Packages

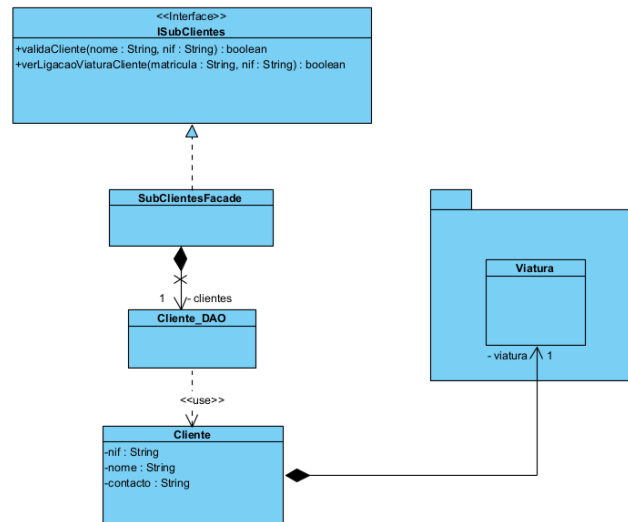


### 5.2. Diagrama de Classes SubViaturas com DAO

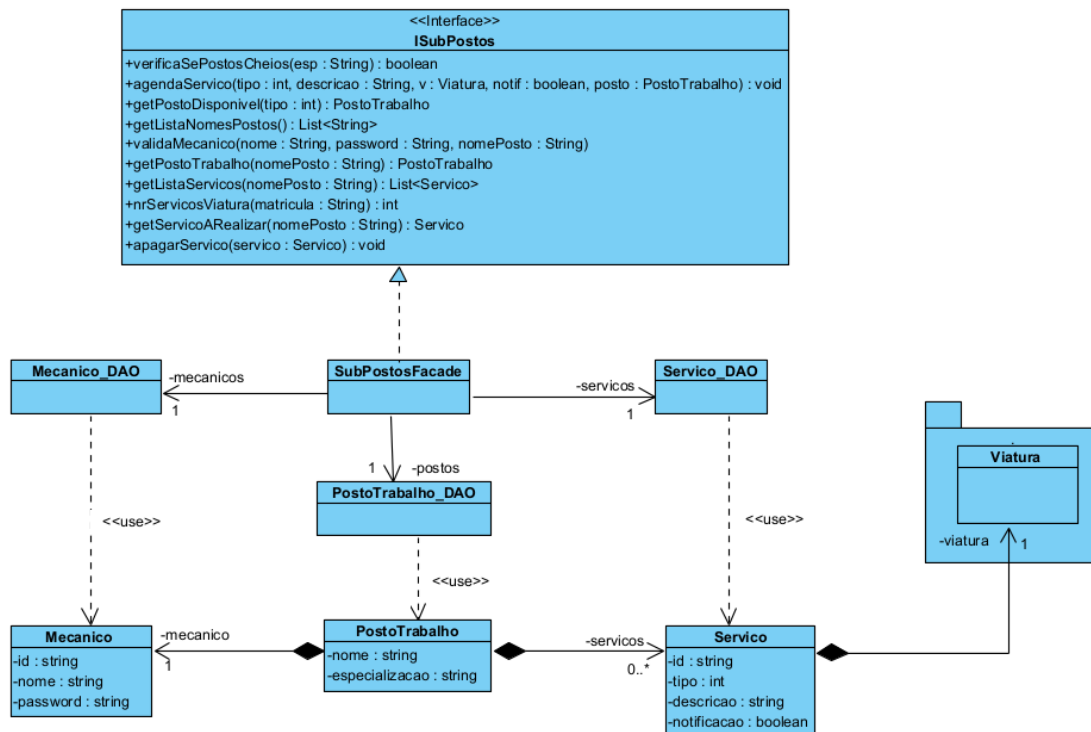




### 5.3. Diagrama de Classes SubClientes com DAO



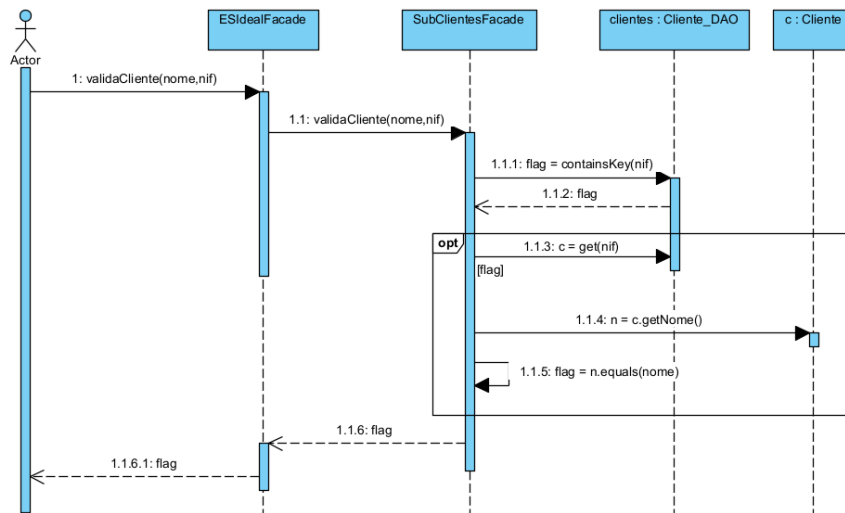
### 5.4. Diagrama de Classes SubPostos com DAO



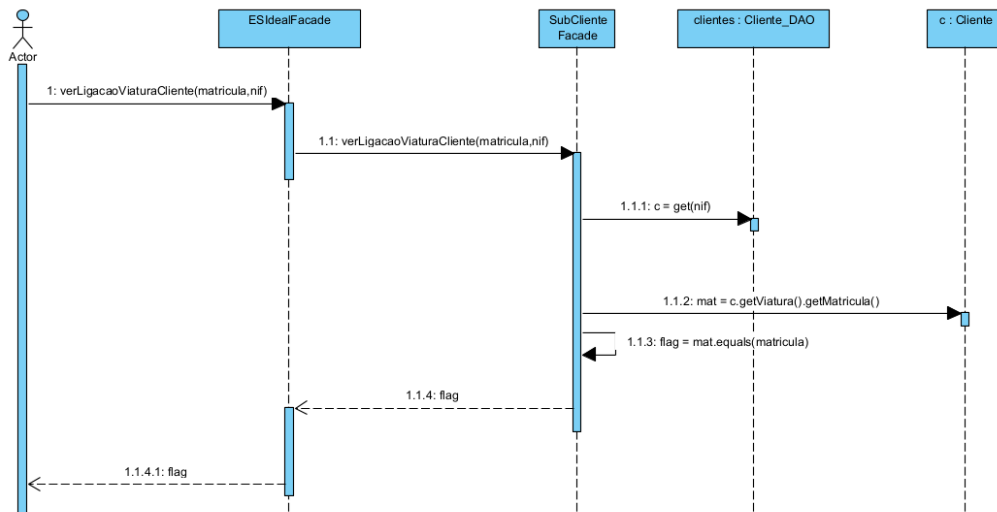
## 5.5. Diagramas de Sequência

### 5.5.1. SubClientes:

#### - validaCliente(String nome, String nif)

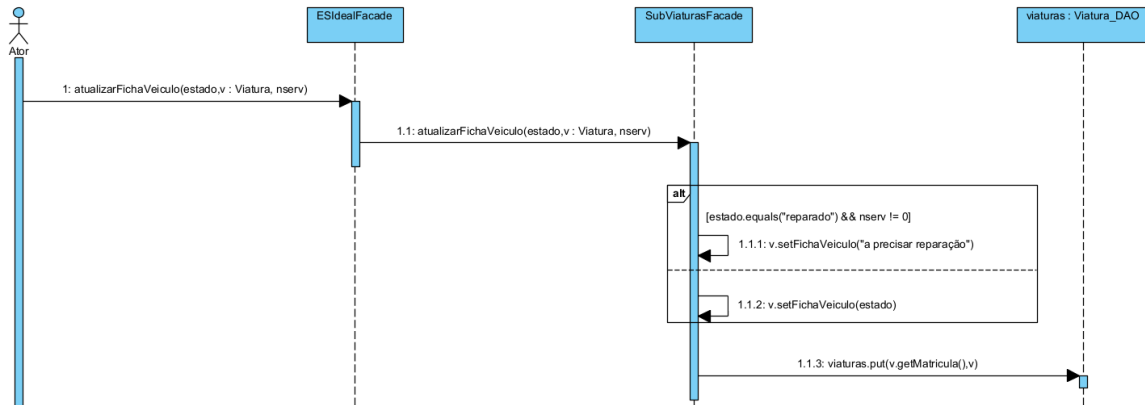


#### - verLigacaoViaturaCliente(String matrícula, String nif)

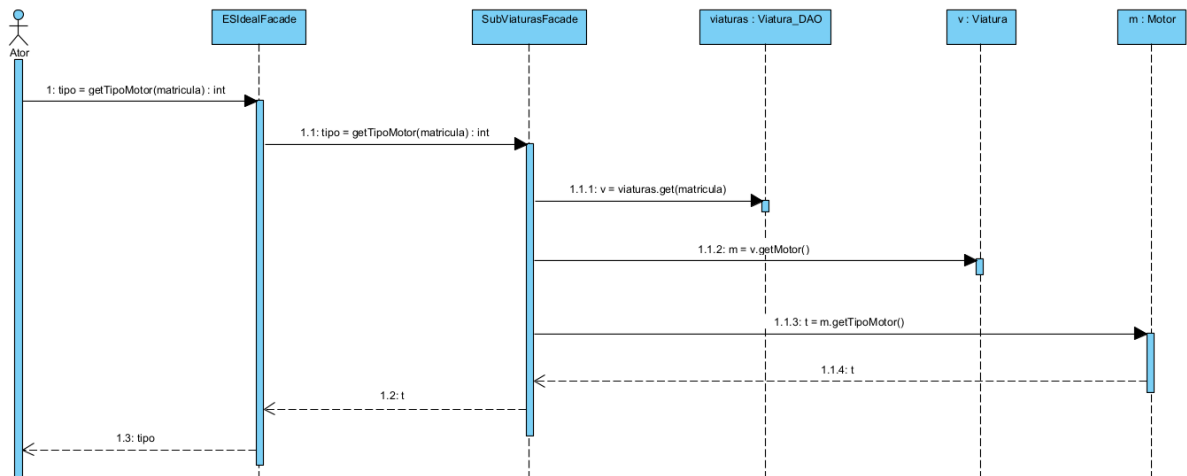


## 5.5.2. SubViaturas:

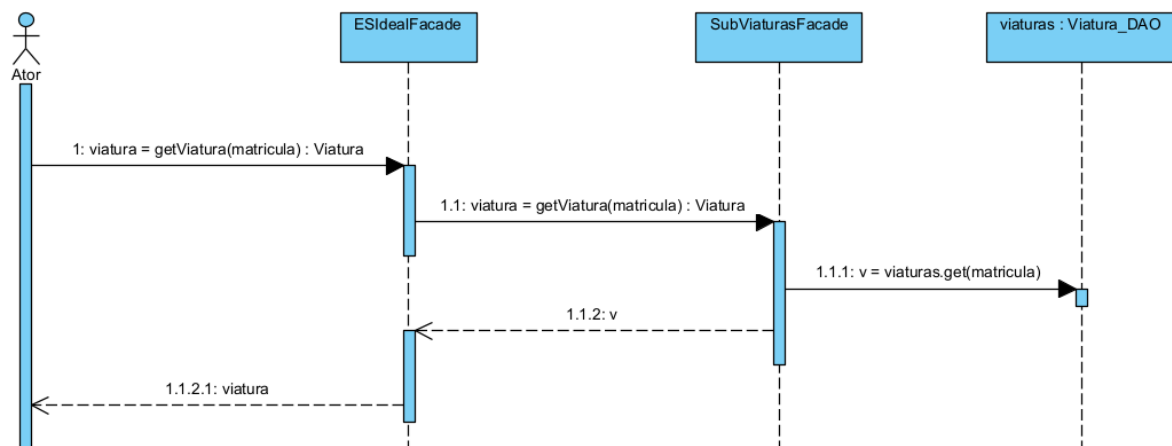
### - atualizarFichaVeiculo(String estado, Viatura v, int nserv)



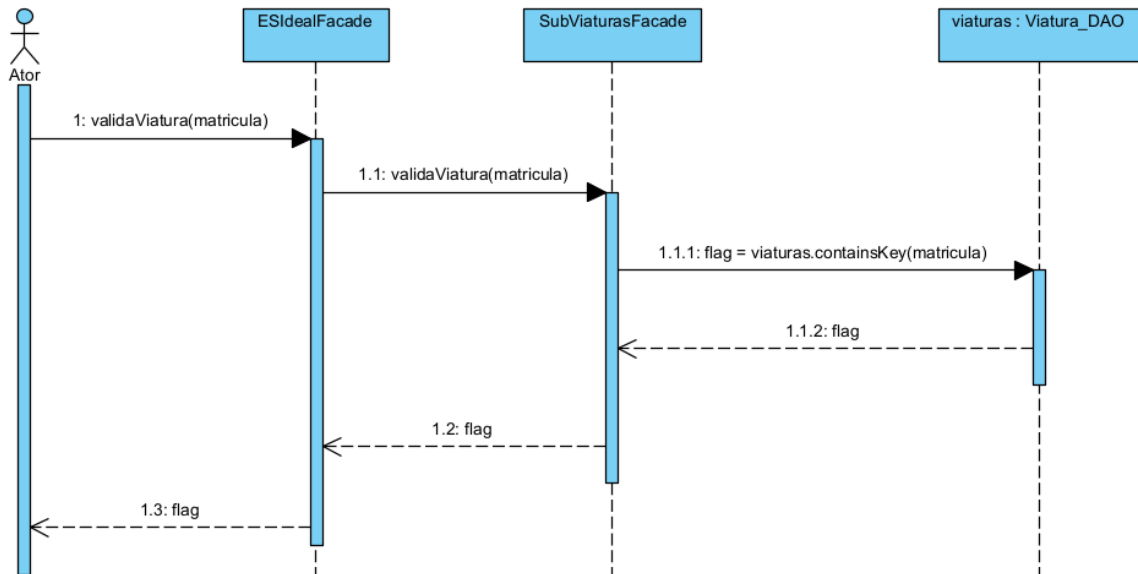
### - getTipoMotor(String matrícula)



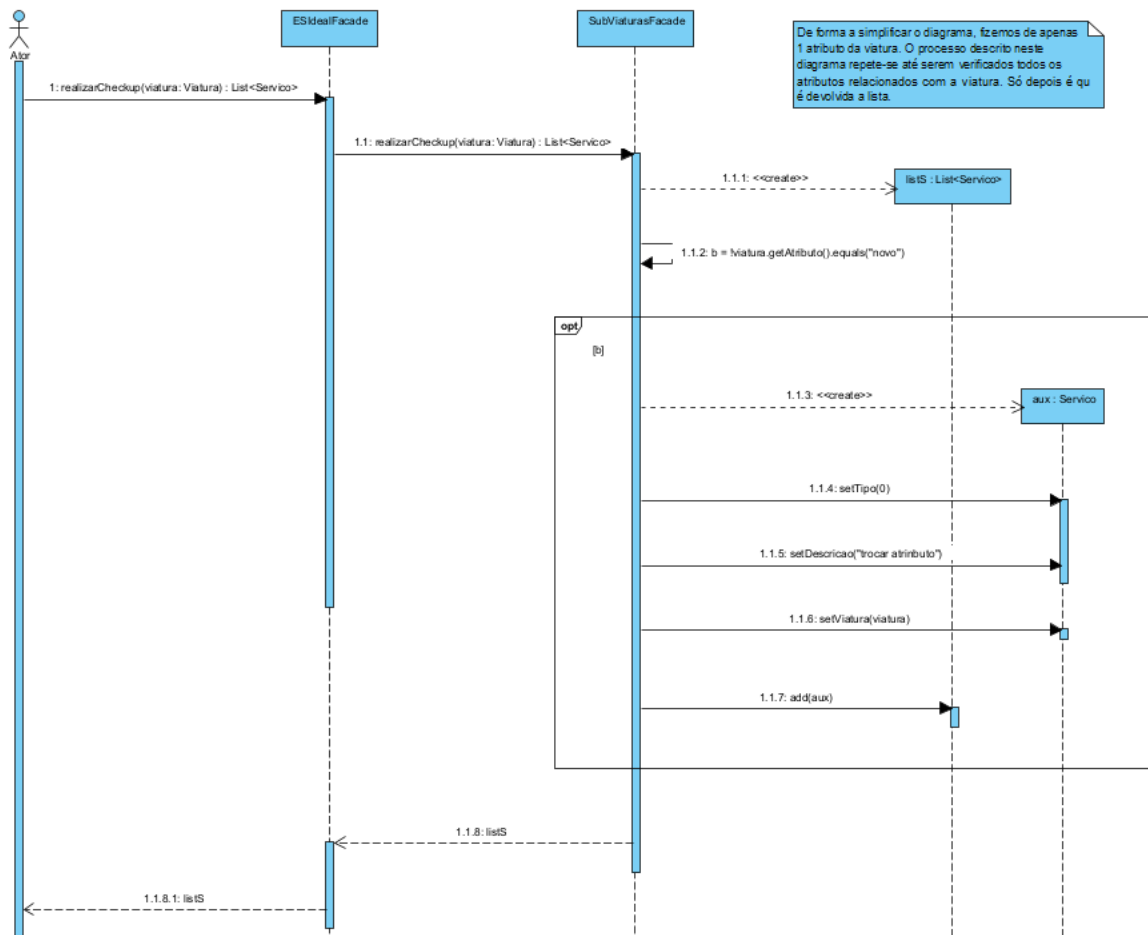
### - getViatura(String matrícula)



## - validaViatura(String matrícula)

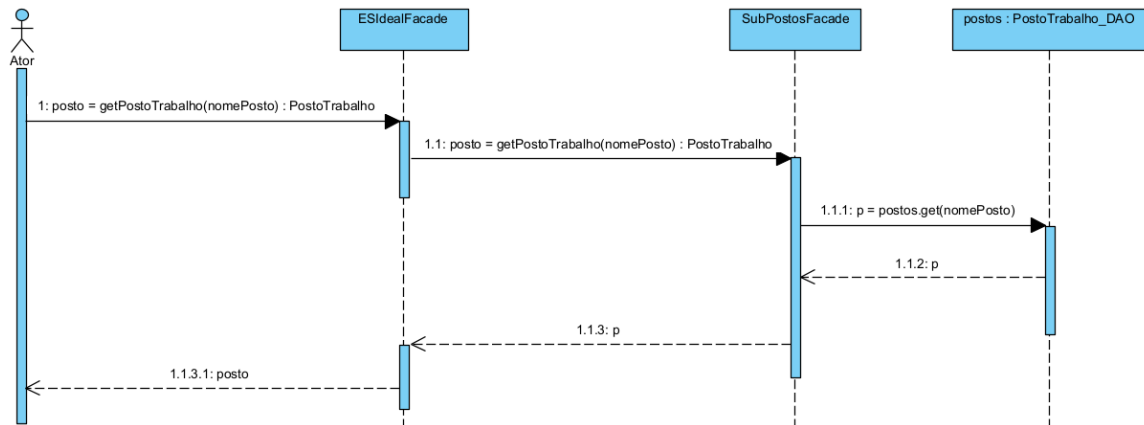


## - realizarCheckup(Viatura viatura)

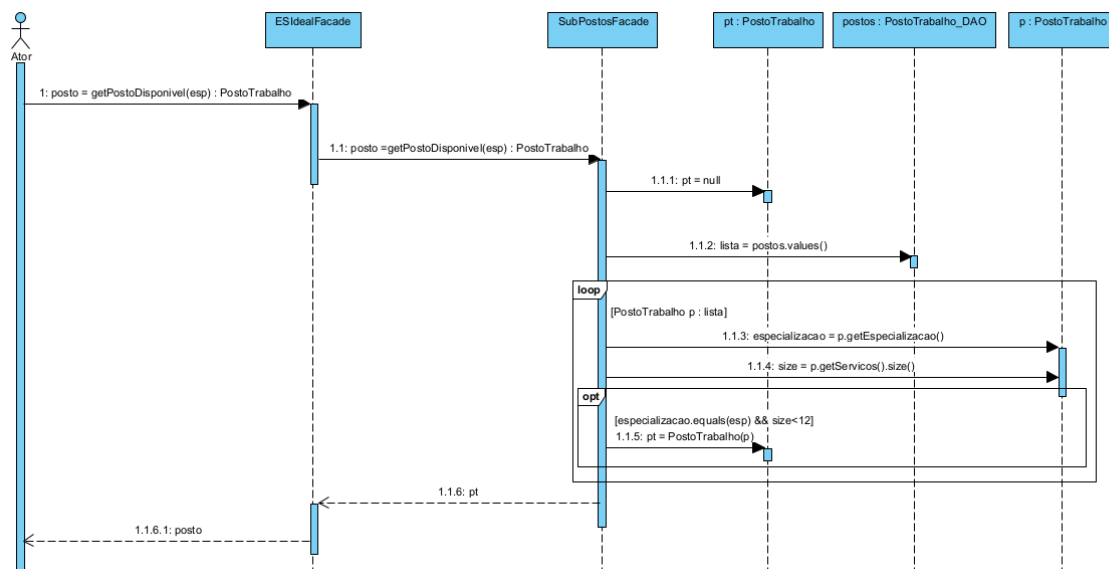


### 5.5.3. SubPostos:

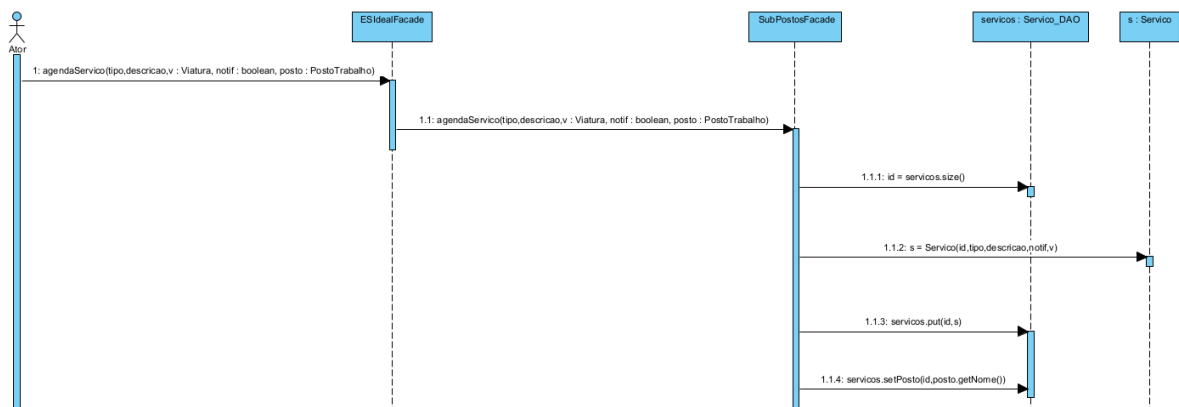
#### - getPostoTrabalho(String nomePosto)



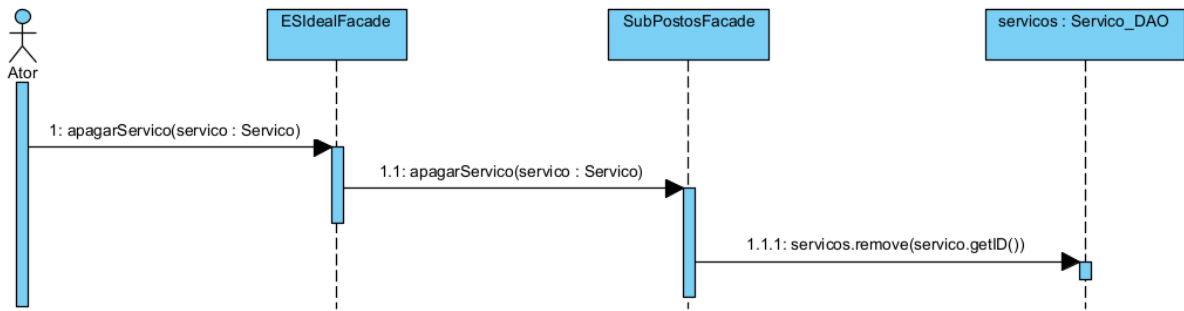
#### - getPostoDisponivel(String esp)



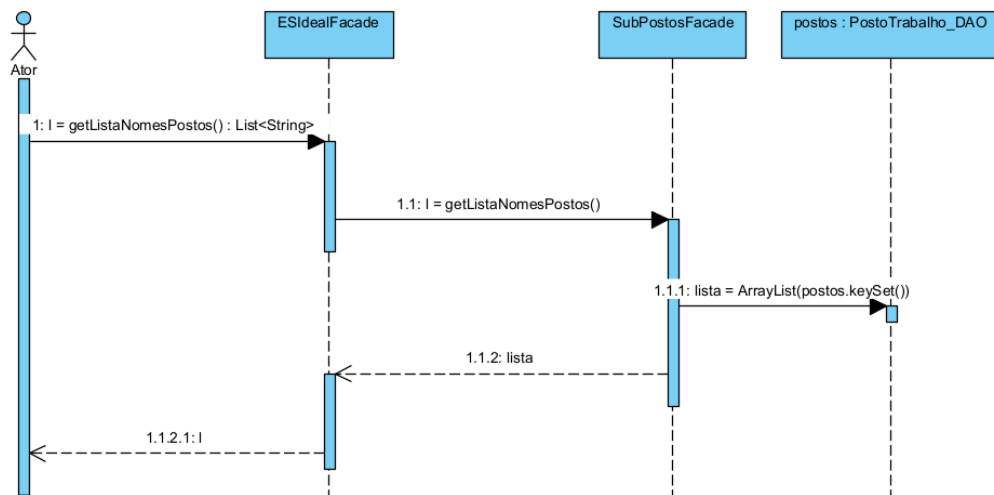
#### - agendaServico(int tipo, String descricao, Viatura v, boolean notif, PostoTrabalho posto)



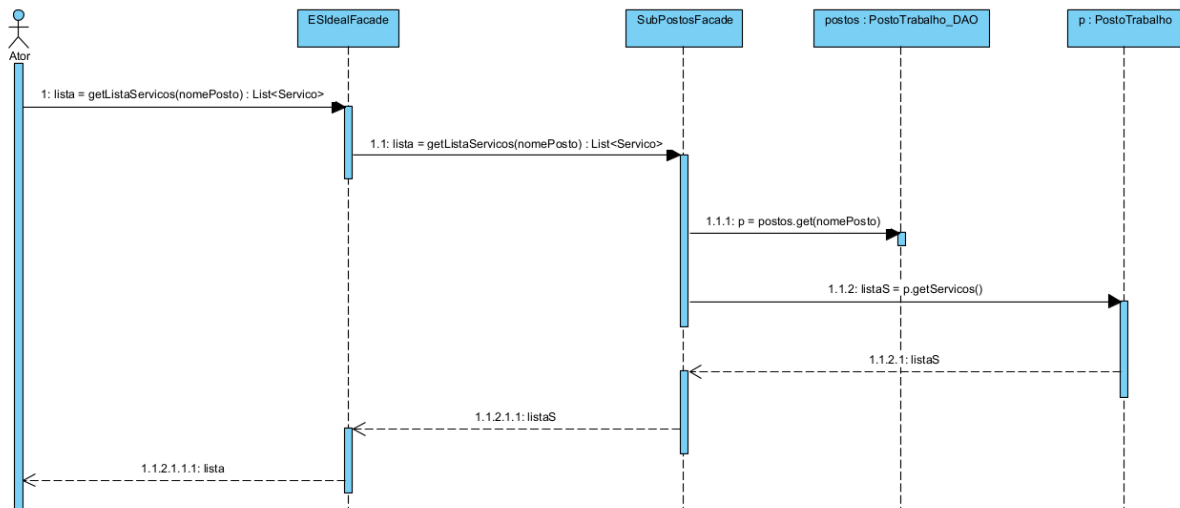
## - apagarServico(Servico servico)



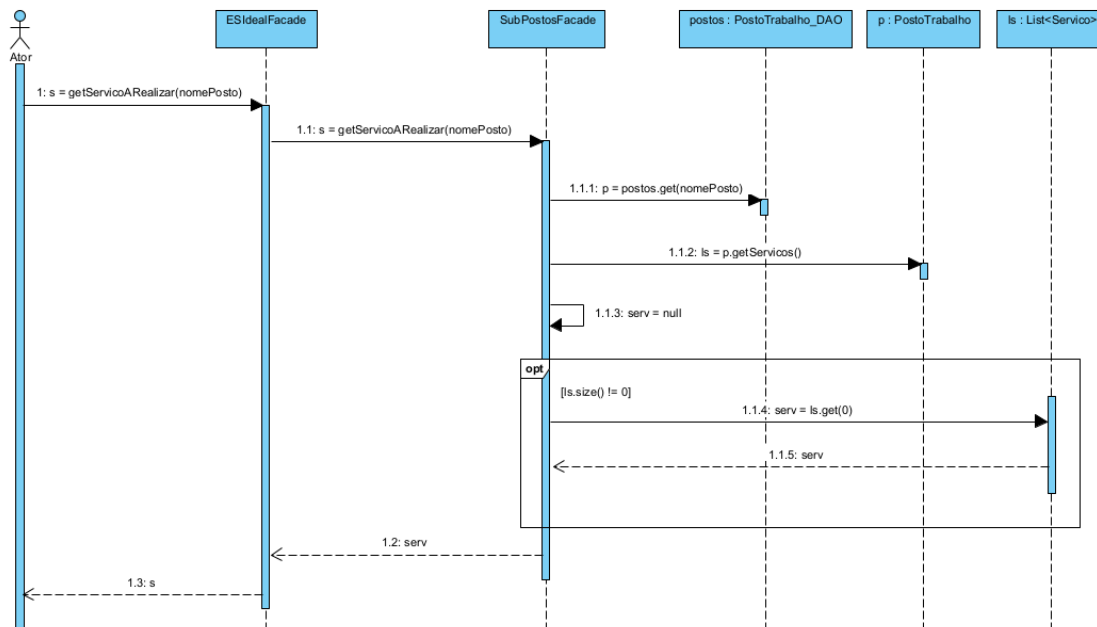
## - getListaNomesPostos()



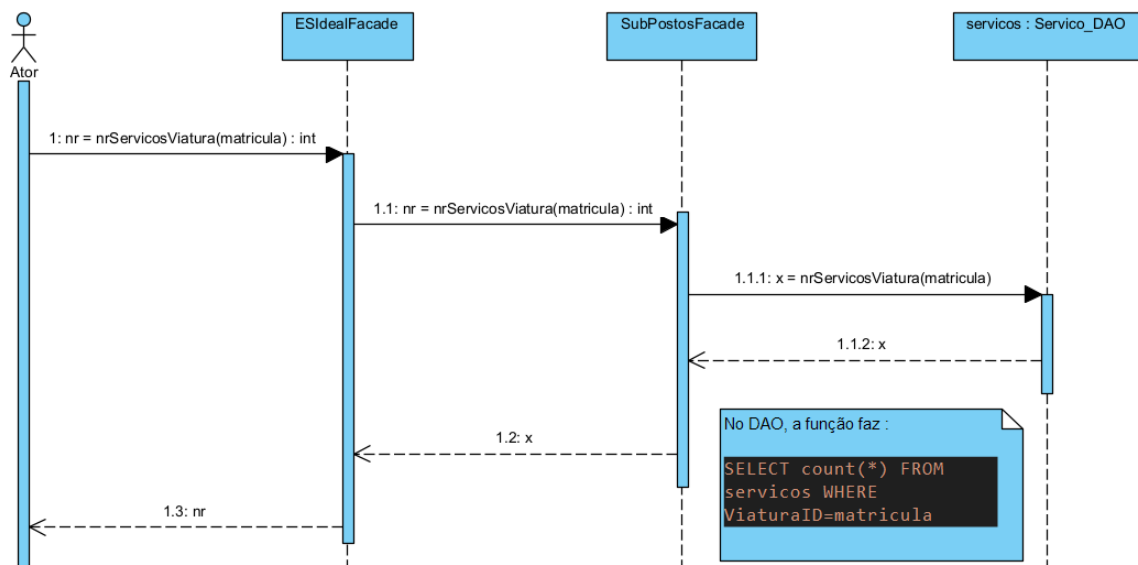
## - getListaServicos(String nomePosto)



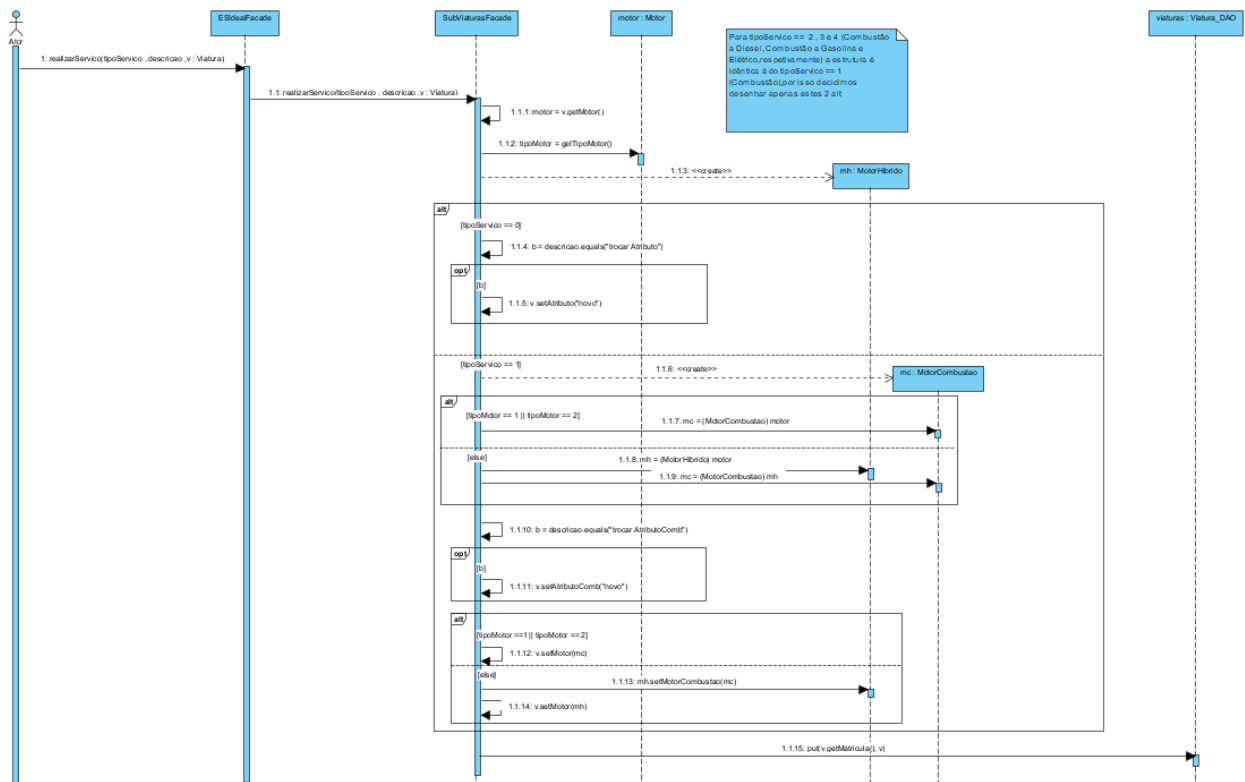
## - getServicoARealizar(String nomePosto)



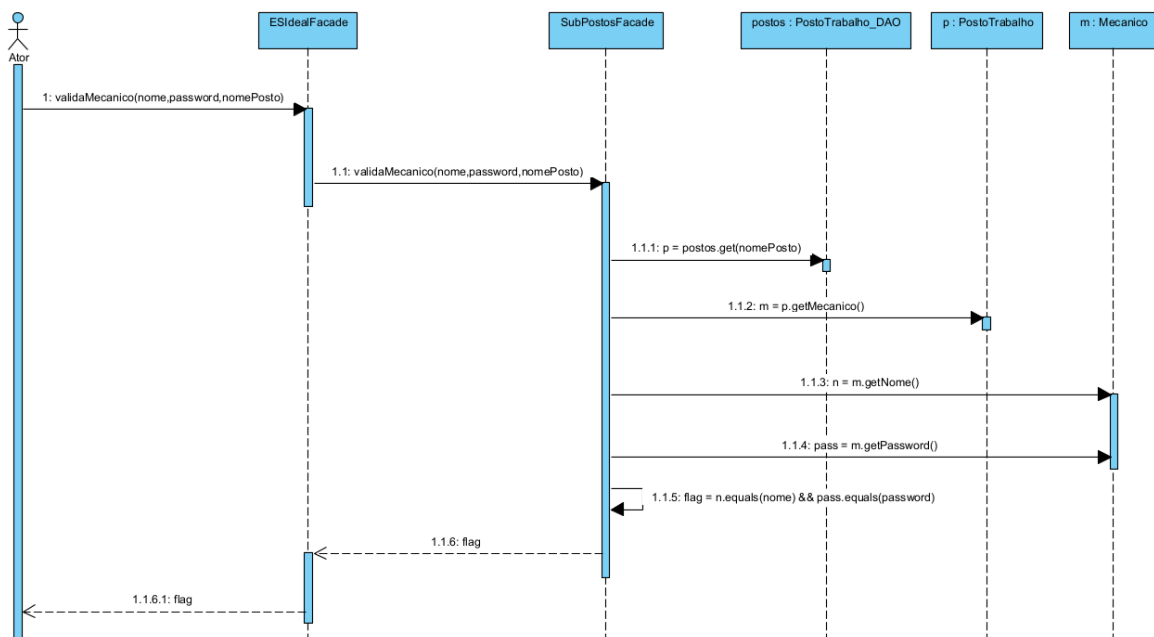
## - nrServicosViatura(String matrícula)



## - realizarServico(int tipoServico , String descricao ,Viatura v)

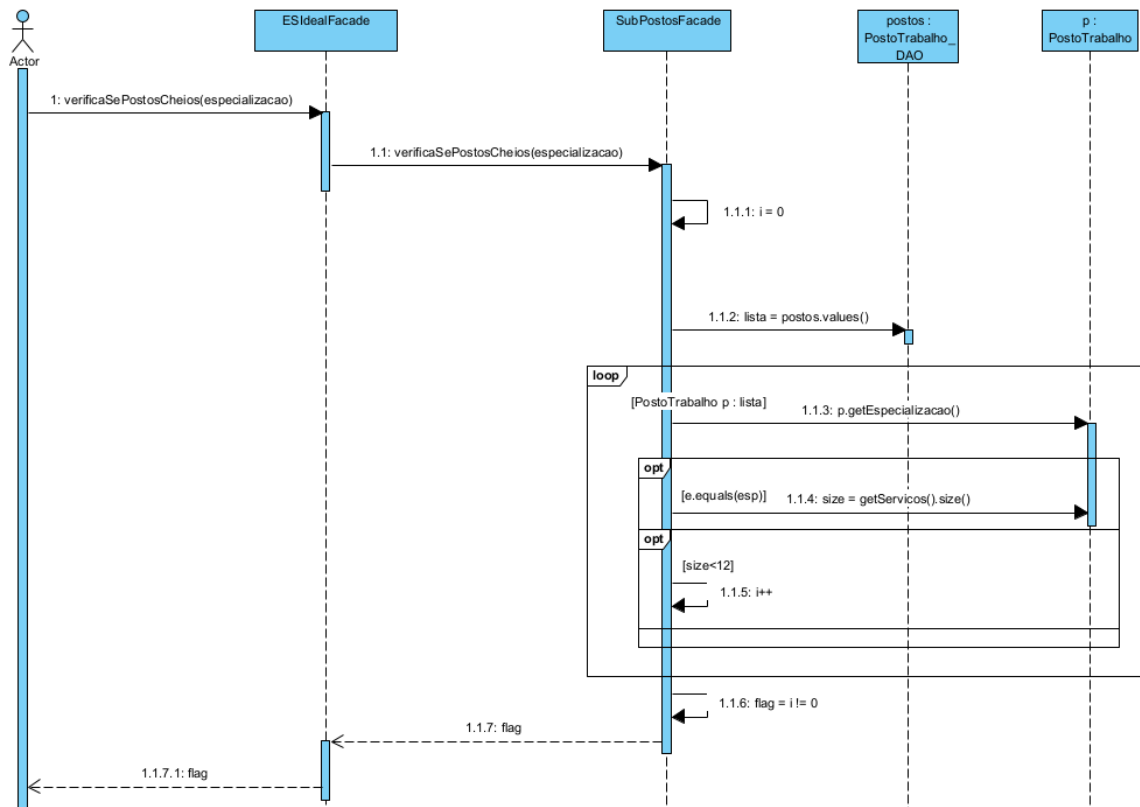


## - validaMecanico(String nome, String password, String nomePosto)





## - verificaSePostosCheios(String especializacao)



## 6. Enunciado do Projeto:



Enunciado TP DSS 2023.pdf