

Desenvolvimento de Sistemas de Software

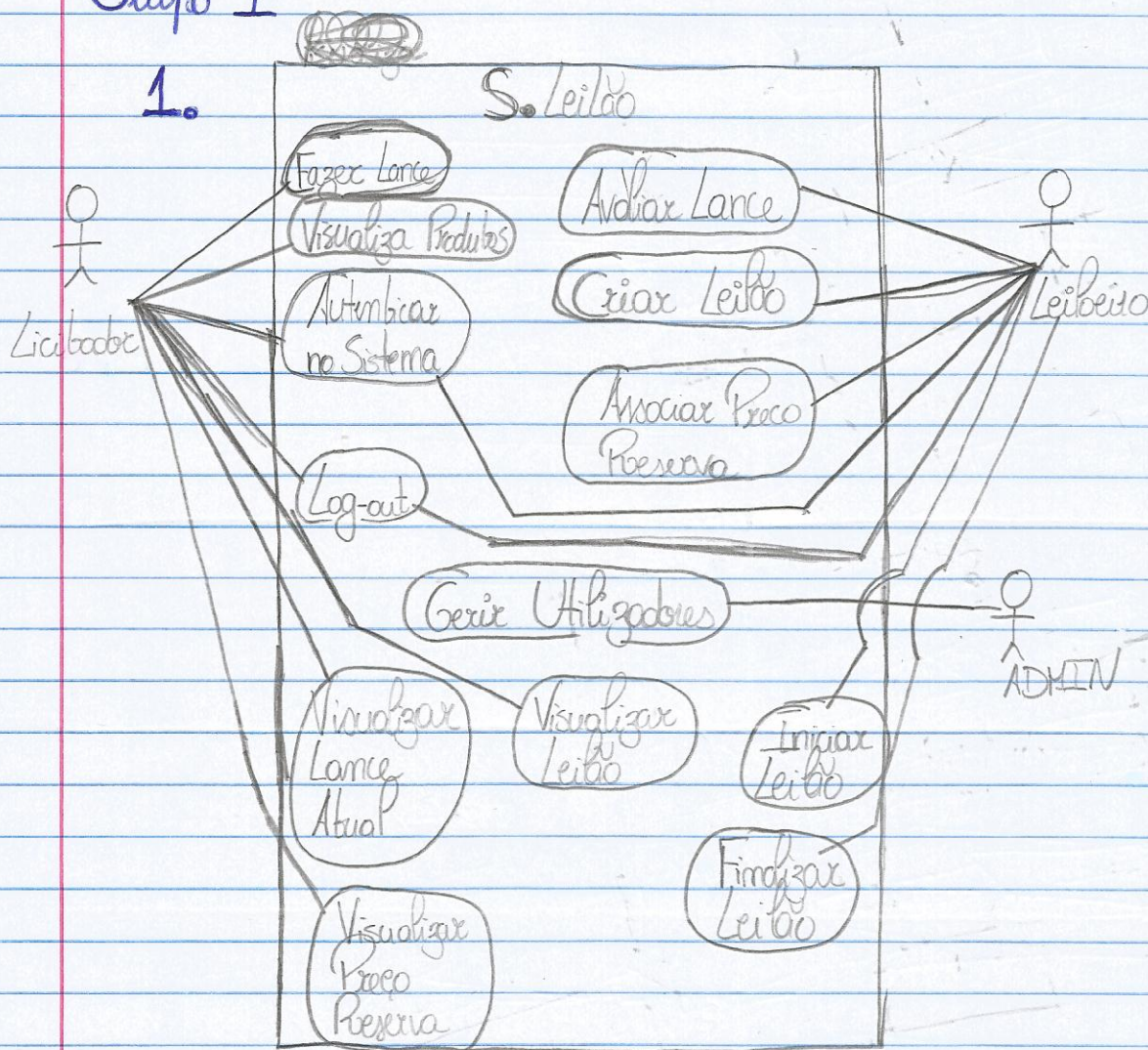
Seste

06/01/2016

03/01/2018

Grupo I

1.



2.

Use Case: Fazer Lance

Desc: Licitador faz lance

Pré: Licitador autenticado no sistema

Pós: Lance efetuado

Actor

Sistema

1. «include» Visualizar Leilão
2. Seleciona 1 leilão ativo

5. Apresenta novo lance

9. Indica que sim
10. Indica preço

3. Valida leilão e verifica preço lance atual
4. Apresenta preço lance atual

6. Valida novo lance
7. Registra novo lance no leilão e limitador desse lance
8. Indica que lance foi efetuado com sucesso e pergunta se deseja associar um valor máximo até ao qual serão feitos lances

11. Valida preço
12. Registra preço limite do limitador para aquele lance
13. Informa que preço foi registrado com sucesso

3.1. Informa que leilão não existe

6. Informa que lance é inferior ao atual e não foi registrado

8.2. Informa que não será associado nenhum valor

11.1. Informa que valor é inferior ao preço definido para o lance e por isso não será associado

1.1. Informa não existência de leilões ativos


Exceção 1
[Leilão inexistente ou inativo]
(passo 3)

Exceção 2
[Lance inferior ao atual]
(passo 6.)

Alternativa 1
[Não deseja associar valor]
(passo 8)

Exceção 3
[Valor inferior ao preço do lance]
(passo 11)

Exceção 4
[Não existem leilões ativos]
(passo 1)

3. 
Leilador

UT

Sistema

ref | Visualizar Leilão

Seleciona leilão ativo

lance := getLance(leilão)

break | [leilão inexistente ou inativo]
Informa que leilão não existe

← Apresenta lance atual

← Apresenta novo lance

registraLance(novoLance, leilador, leilão)

break | [lance inferior ao atual]
Informa que lance é inferior
ao atual e não foi registrado

← Indica que lance foi efetuado
com sucesso ...

alt | [Não deseja associar valor]
Indica que não

← Informa que não será associado nenhum valor

[Sim]

Indica que sim

Indica preço

registraPreco(preco, leilão, leilador)

break | [Valor inferior ao preço de lance]
Informa que valor é inferior
ao preço definido para o lance...

← Informa preço registrado com sucesso

Grupo II

4.

```
public class SGE {
    private Calendar data;
    private Map<String, Eleicao> eleicoes; (clone aqui)
    private Map<String, Eleitor> eleitores; (clone aqui)

    public boolean podeVotar(String codEleicao, String numEleitor)
    {
        ...
    }
}
```

```
public class Eleicao {
```

```
    private Calendar dataInicio;
```

```
    // // dataFim;
```

```
    // String codEleicao;
```

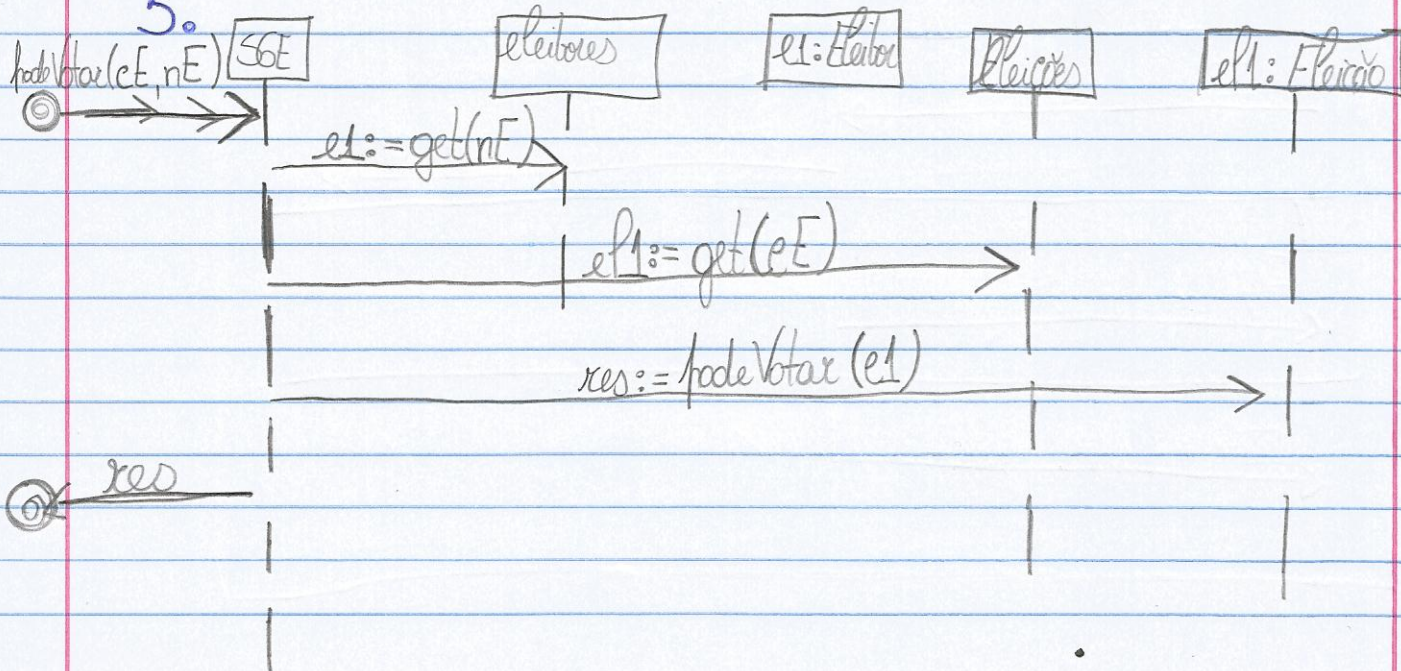
```
    private Collection<Eleitor> eleitores; (sem clone)
```

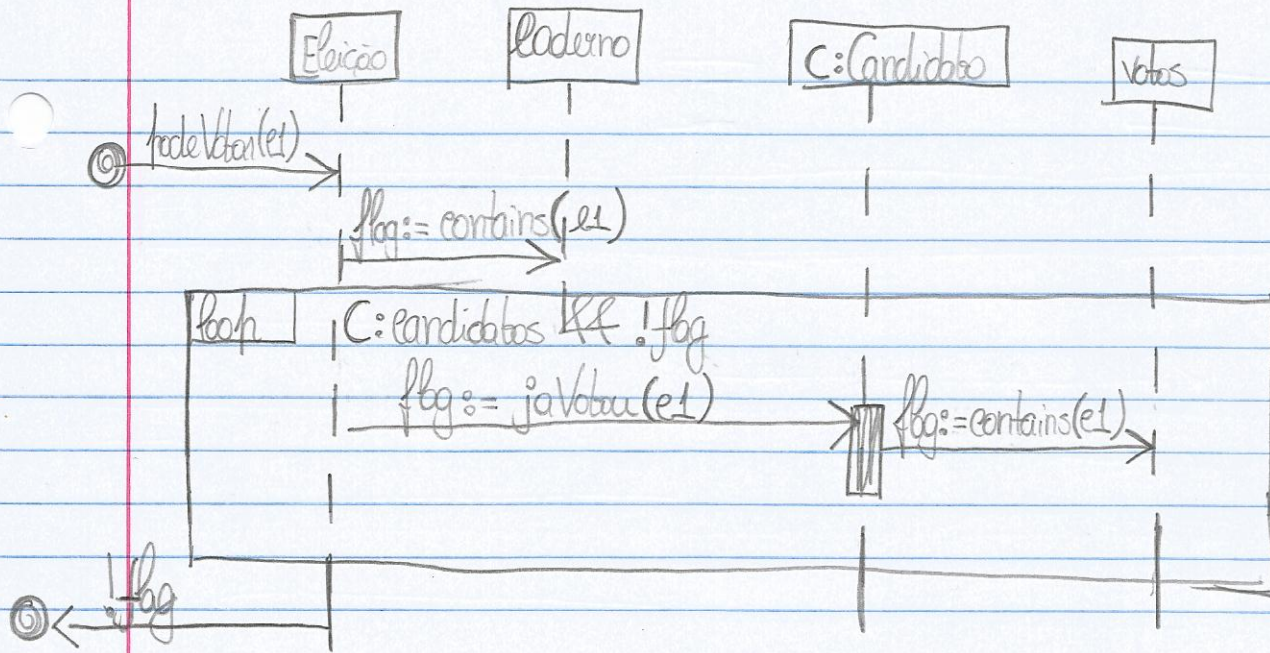
```
    private Collection<Candidato> candidatos; ( // )
```

```
    public boolean podeVotar(Eleitor e1)
```

```
    {
        ...
    }
}
```

5.





6.

