

# Cargos



&



# Salários

## Grupo 2

Guilherme Ezequiel

Leonardo Silva

Lucas Ivan

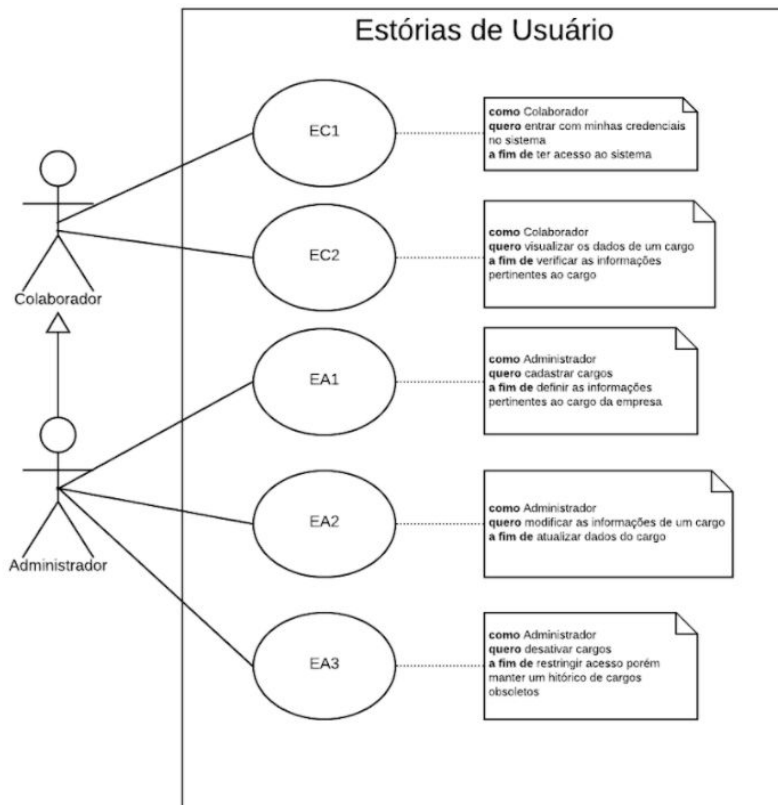
Sabrina Schmidt

Vitor Gehrke

feat. Setores

# Backlog do Produto

## Sprint 1



# Backlog do Produto

## Sprint 2

### Cargo

Cadastrar  
Alterar  
Remover  
Visualizar

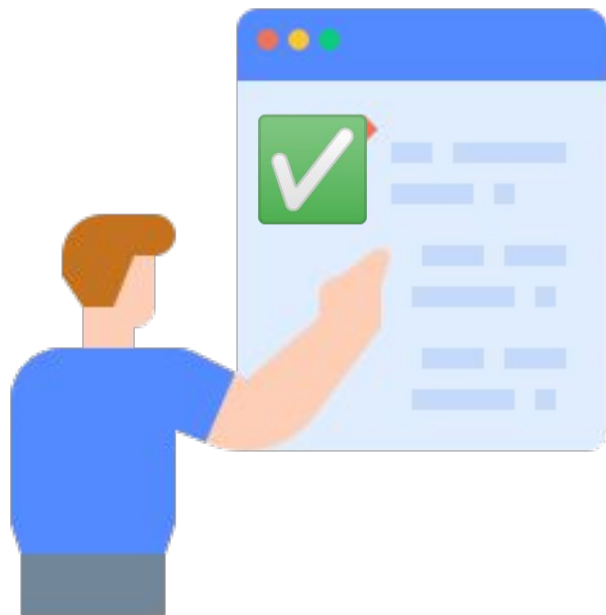
idSetor

### Setor

Cadastrar  
Alterar  
Remover  
Visualizar



## O que foi feito



- ✓ Refatoração - Classe Cargo
  - Cadastrar, consultar, alterar e deletar
- ✓ Criação - Classe Setor
  - Cadastrar, consultar alterar e deletar
- ✓ Interface
  - Métodos para assinatura
- ✓ Testes
  - Testes de todos os métodos implementados

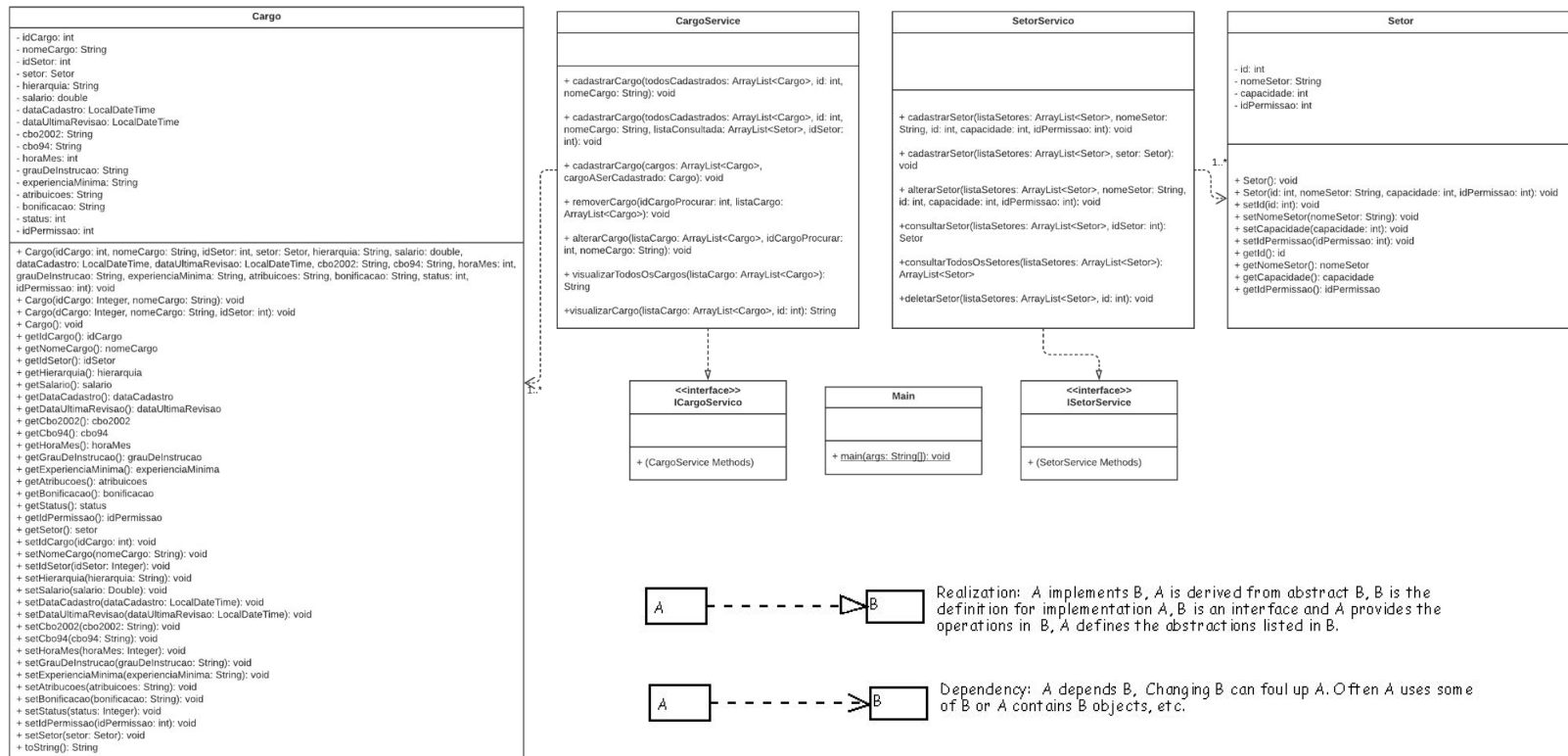
# Interface - Cargo

```
public interface ICargoService {  
  
    public void cadastrarCargo(ArrayList<Cargo> todosCadastrados, int id, String nomeCargo);  
  
    public void cadastrarCargo(ArrayList<Cargo> todosCadastrados, int id, String nomeCargo, ArrayList<Setor>  
listaConsultada, int idSetor);  
  
    public void cadastrarCargo(ArrayList<Cargo> cargos, Cargo cargoASerCadastrado);  
  
    public void removerCargo(int idCargoProcurar, ArrayList<Cargo> listaCargo);  
  
    public void alterarCargo(ArrayList<Cargo> listaCargo, int idCargoProcurar, String nomeCargo);  
  
    public String visualizarTodosOsCargos(ArrayList<Cargo> listaCargo);  
  
    public String visualizarCargo(ArrayList<Cargo> listaCargo, int id);  
  
}
```

# Interface - Setor

```
public interface ISetorServico {  
  
    public void cadastrarSetor(ArrayList<Setor> listaSetores, String nomeSetor, int id, int capacidade, int idPermissao);  
  
    public void cadastrarSetor(ArrayList<Setor> listaSetores, Setor setor);  
  
    public void alterarSetor(ArrayList<Setor> listaSetores, String nomeSetor, int id, int capacidade, int id Permissão);  
  
    public Setor consultarSetor(ArrayList<Setor> listaSetores, int id Setor);  
  
    public ArrayList<Setor> consultarTodosOsSetores(ArrayList<Setor> listaSetores);  
  
    public void deletarSetor(ArrayList<Setor> listaSetores, int id);  
}
```

# Diagrama de Classes (UML)



# Debug

```
ArrayList<Setor> listaSetores = new ArrayList<Setor>();
ArrayList<Cargo> listaCargos = new ArrayList<Cargo>();
SetorService setorSvc = new SetorService();
CargoService cargoSvc = new CargoService();

setorSvc.cadastrarSetor(listaSetores, "Recursos humanos", 3521, 20, 1524);
setorSvc.cadastrarSetor(listaSetores, "Comercial", 3522, 20, 1525);
setorSvc.cadastrarSetor(listaSetores, "Financeiro", 3523, 20, 1526);
setorSvc.cadastrarSetor(listaSetores, "Contabilidade", 3524, 20, 1527);
setorSvc.cadastrarSetor(listaSetores, "Administracao", 3525, 20, 1528);

cargoSvc.cadastrarCargo(listaCargos, 1, "Auxiliar de Administração", listaSetores, 3525);
cargoSvc.cadastrarCargo(listaCargos, 2, "Recrutador", listaSetores, 3521);
cargoSvc.cadastrarCargo(listaCargos, 3, "Auxiliar de Contabilidade", listaSetores, 3524);
cargoSvc.cadastrarCargo(listaCargos, 4, "Representante Comercial", listaSetores, 3522);
















setorSvc.alterarSetor(listaSetores, "RH", 3521, 20, 1529);
setorSvc.deletarSetor(listaSetores, 3524);

cargoSvc.alterarCargo(listaCargos, 2, "Recrutador de RH");
cargoSvc.removerCargo(4, listaCargos);
```



# Debug

```
setorSvc.cadastrarSetor(listaSetores, "Recursos humanos", 3521, 20, 1524);  
setorSvc.cadastrarSetor(listaSetores, "Comercial", 3522, 20, 1525);  
setorSvc.cadastrarSetor(listaSetores, "Financeiro", 3523, 20, 1526);  
setorSvc.cadastrarSetor(listaSetores, "Contabilidade", 3524, 20, 1527);  
setorSvc.cadastrarSetor(listaSetores, "Administracao", 3525, 20, 1528);
```

Name	Value
 cadastrarSetor() returned	(No explicit return value)
 args	String[0] (id=20)
 listaSetores	ArrayList<E> (id=21)
 [0]	Setor (id=41)
 capacidade	20
 id	3521
 idPermissao	1524
 nomeSetor	"Recursos humanos" (id=57)
>  [1]	Setor (id=47)
>  [2]	Setor (id=50)
>  [3]	Setor (id=53)
>  [4]	Setor (id=56)
 listaCargos	ArrayList<E> (id=31)
 setorSvc	SetorService (id=32)
 cargoSvc	CargoService (id=36)

# Debug

```
cargoSvc.cadastrarCargo(listaCargos, 1, "Auxiliar  
de Administração", listaSetores, 3525);  
  
cargoSvc.cadastrarCargo(listaCargos, 2,  
"Recrutador", listaSetores, 3521);  
  
cargoSvc.cadastrarCargo(listaCargos, 3, "Auxiliar  
de Contabilidade", listaSetores, 3524);  
  
cargoSvc.cadastrarCargo(listaCargos, 4,  
"Representante Comercial", listaSetores, 3522);
```















▪ salario	null
▼ ▪ setor	Setor (id=60)
▪ capacidade	20
▪ id	3521
▪ idPermissao	1524
> ▪ nomeSetor	"Recursos humanos" (id=65)
▪ status	0



Name	Value
↳ cadastrarCargo() returned	(No explicit return value)
▪ args	String[0] (id=20)
> ▪ listaSetores	ArrayList<E> (id=21)
▼ ▪ listaCargos	ArrayList<E> (id=31)
> ▪ [0]	Cargo (id=61)
▼ ▪ [1]	Cargo (id=65)
▪ atribuiçoes	null
▪ bonificacao	null
▪ cbo2002	null
▪ cbo94	null
▪ dataCadastro	null
▪ dataUltimaRevisao	null
▪ experienciaMinima	null
▪ grauDeInstrucao	null
▪ hierarquia	null
▪ horaMes	0
▪ idCargo	2
▪ idPermissao	0
▪ idSetor	3521
> ▪ nomeCargo	"Recrutador" (id=73)
▪ salario	null
> ▪ setor	Setor (id=74)
▪ status	0
> ▪ [2]	Cargo (id=68)
> ▪ [3]	Cargo (id=71)
▪ setorSvc	SetorServico (id=32)
▪ cargoSvc	CargoService (id=36)

# Debug

```
setorSvc.alterarSetor(listaSetores,  
"RH", 3521, 20, 1529);  
setorSvc.deletarSetor(listaSetores,  
3524);
```

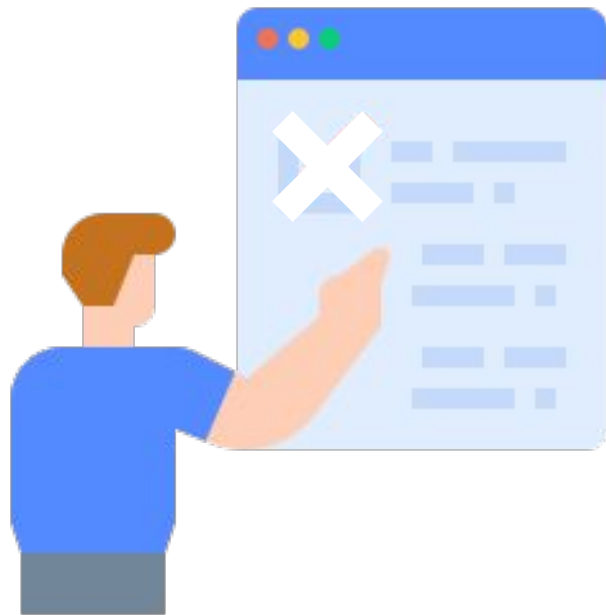
Name	Value
 deletarSetor() returned	(No explicit return value)
 args	String[0] (id=20)
 listaSetores	ArrayList<E> (id=21)
 [0]	Setor (id=41)
 capacidade	20
 id	3521
 idPermissao	1529
 nomeSetor	"RH" (id=80)
 [1]	Setor (id=47)
 [2]	Setor (id=50)
 [3]	Setor (id=56)
 listaCargos	ArrayList<E> (id=31)
 setorSvc	SetorService (id=32)
 cargoSvc	CargoService (id=36)

# Debug

```
cargoSvc.alterarCargo(listaCargos, 2,
"Recrutador de RH");
cargoSvc.removerCargo(4, listaCargos);
```

Name	Value
➡ removerCargo() returned	(No explicit return value)
• args	String[0] (id=20)
> • listaSetores	ArrayList<E> (id=22)
▼ • listaCargos	ArrayList<E> (id=33)
> • [0]	Cargo (id=47)
▼ • [1]	Cargo (id=50)
• atribuicoes	null
• bonificacao	null
• cbo2002	null
• cbo94	null
• dataCadastro	null
• dataUltimaRevisao	null
• experienciaMinima	null
• grauDeInstrucao	null
• hierarquia	null
• horaMes	0
• idCargo	2
• idPermissao	0
• idSetor	3521
> • nomeCargo	"Recrutador de RH" (id=63)
• salario	null
> • setor	Setor (id=60)
• status	0
> • [2]	Cargo (id=52)
• setorSvc	SetorService (id=34)
• cargoSvc	CargoService (id=37)

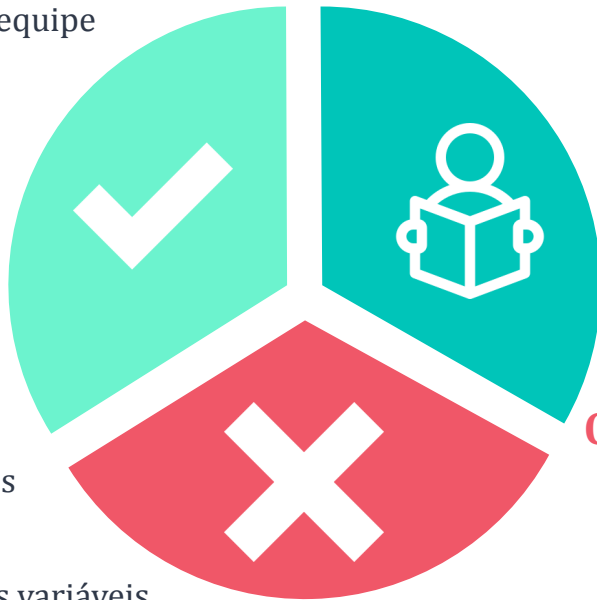
## O que não foi feito



- ☐ Padronizar alguns métodos e variáveis
- ☐ Validação de dados.

## O que deu certo

- ✓ Implementação dos métodos assinados da interface
- ✓ Pair programming/Comunicação de equipe
- ✓ Testes unitários
- ✓ Melhorias da documentação
- ✓ Criação da classe Setor
- ✓ Encapsulamento de todas as variáveis
- ✓ Refatoração da classe Cargo e de suas variáveis



## O que aprendemos



Avanço com Java

- Implementação de interface
- Overloading
- POO



Um pouco mais sobre GIT/Github



Documentação e Diagrama UML

## O que deu errado

- ☐ Controle de versionamento Git/Github
- ☐ Comunicação nos commits
- ☐ Perda de alterações por falta de commits
- ☐ Divisão de tarefas



## Fazer mais

Organização do trello

Conhecer mais o GIT

“git pull” antes de alterar código

## Continuar fazendo

Pair programming

Compartilhamento de conhecimentos

## Fazer menos

Commitar antes de revisar o código

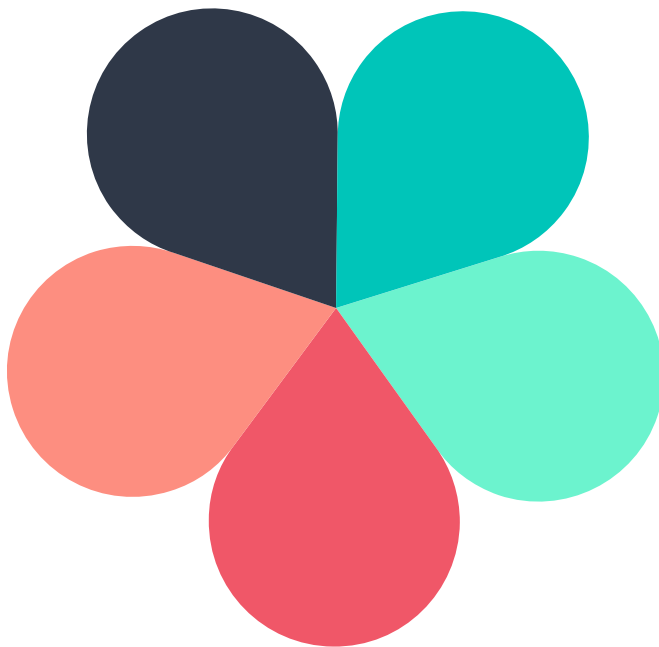
## Começar a fazer

Convenções

Documentar antes de programar

## Parar de fazer

Teste depois do método



# Obrigade pela atenção!

## Grupo 2

Guilherme Ezequiel

Leonardo Silva

Lucas Ivan

Sabrina Schmidt

Vitor Gehrke