

**UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI**  
**SISTEMAS DE INFORMAÇÃO**

**Tales Félix**

Padrão de Projeto - MVC

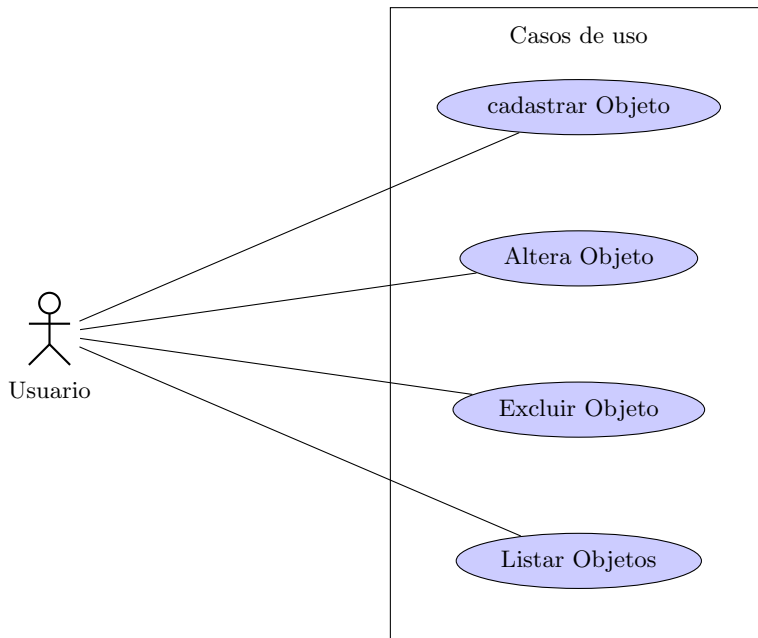
Docente: Eduardo Pelli.

Diamantina, 18 de Outubro de 2020

# 1 Introdução

Um colecionador precisa de um sistema para guardar informações de seus objetos. O sistema deverá fazer registro do nome e descrição geral de cada pertence. Ao ser solicitado pelo usuário o sistema deverá retonar uma lista contendo todos objetos cadastrados.

## 2 Diagrama de Casos de Uso



O diagrama de casos de uso corresponde a uma visão externa do sistema e representa graficamente os atores, os casos de uso, e os relacionamentos entre estes elementos. Ele tem como objetivo ilustrar em um nível alto de abstração quais elementos externos interagem com que funcionalidades do sistema, ou seja, a finalidade de um diagrama de caso de uso é apresentar um tipo de diagrama de contexto que apresenta os elementos externos de um sistema e as maneiras segundo as quais eles as utilizam.

## 3 Fluxo de Eventos

### 3.1 Cadastrar Objeto

O usuário Deverá preencher os campos, "Nome" e "Descrição" e sobmeter clicando no botão Enviar. Aparecerá uma mensagem de sucesso, caso contrário mostrarar um erro e o usuário terá que repetir a operação.

### 3.2 Listar Objetos

Após clicar em "Listar" aparecerar uma lista de elementos.  
Caso não tenha nenum matrial cadrastado mostrarar a lista vazia.

### 3.3 Alterar Objetos

O usuário Deverá preencher os campos, "Nome", "Descrição" e "ID" e submeter  
Se os ID do objeto que deseja alterar não estiver devidadamente preenchido mostrarar uma mensagem de erro.  
Caso contrário aparecerá uma mensagem de sucesso

### 3.4 Excluir Objeto

O usuário Deverá preencher "ID" e submeter  
Se os ID do objeto que deseja excluir não estiver devidadamente preenchido mostrarar uma mensagem de erro.

Caso contrário aparecerá uma mensagem de sucesso

## 4 Classes

### Lista de Colecao

#### Registro

- colecao : Colecao[]

+ incluirColecao(Colecao: colecao): bool  
+ EnviarListaColecao(): String  
+ Excluir(String: id): bool  
+ Alterar(id: String, nome: String, descricao: String):bool

#### Routes

Salvar(): bool  
Listar(): String  
Alterar(id: String, nome: String, descricao: String): boll  
Excluir(id: String): boll

#### Colecao

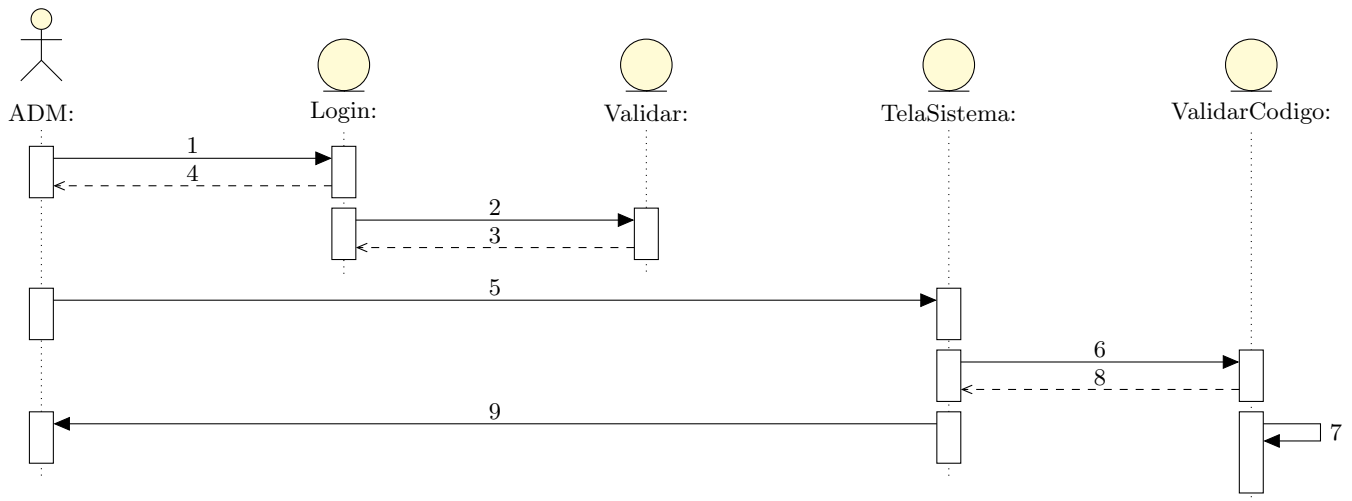
- nome : String  
- descricao : String  
- id : String

+ Colecao(nome: String, descricao: String)  
+ getNome(): String  
+ setNome(nome: String): void  
+ getDescricao(): String  
+ setDecriscao(descricao: String): void  
+ getId(): String  
+ setId(id: String): void  
+ toString(): String

#### Index

## 5 Diagrama de Sequência

### 5.1 Diagrama de Sequência Cadastrar Colaborador

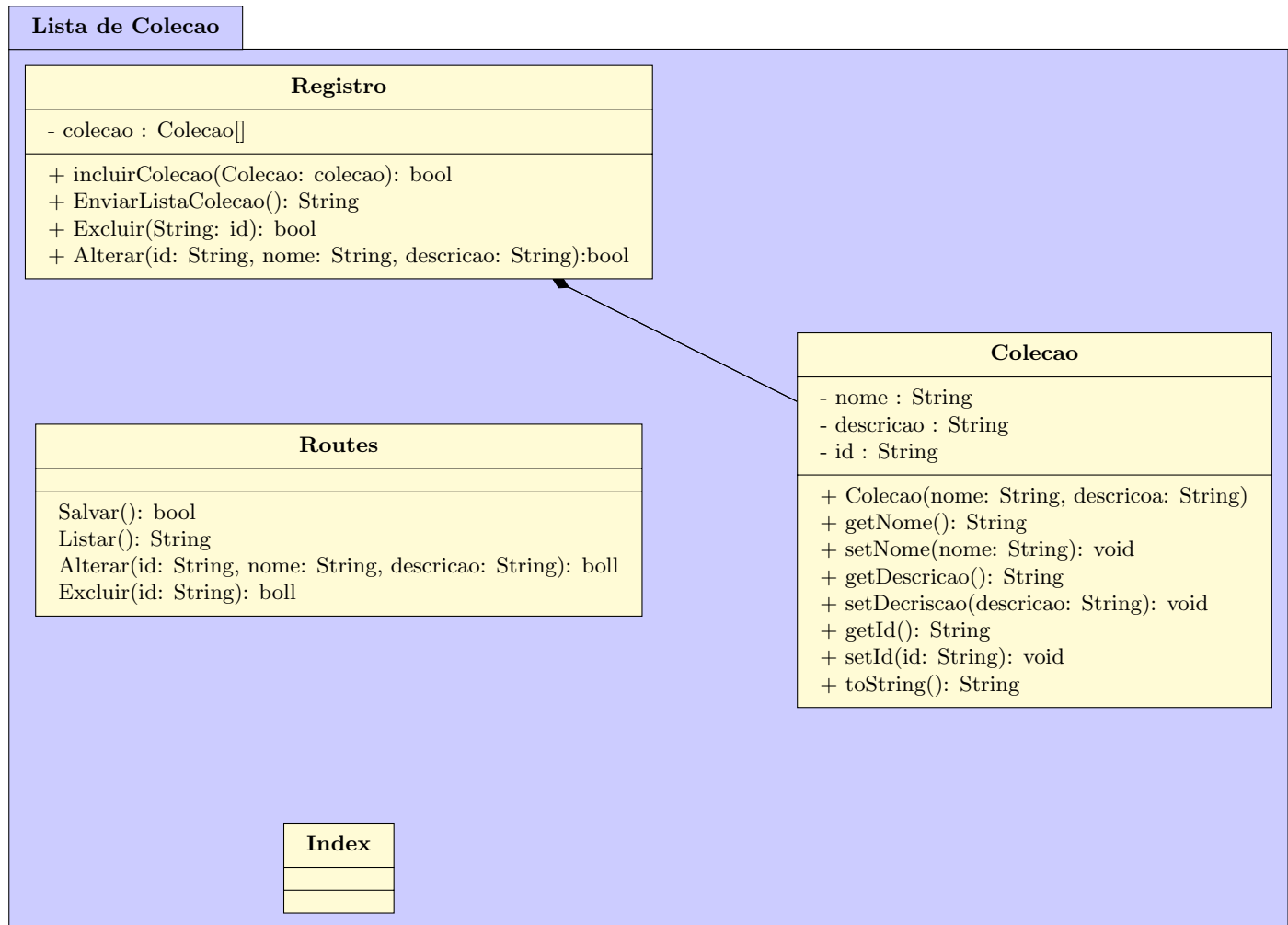


#### Descrição:

1. Logar()
2. Validar Login
3. Resposta validação
4. Login Efetuado
5. Formulário
6. CadastrarColaborador()
7. Validar Dados Inseridos
8. Mostrar mensagem de sucesso ou mostrar erro
9. Retonar Início

No que se refere ao diagrama de seqüência, preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em determinado processo, ou seja, quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processo. Dessa forma, determinar a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem entre si dentro de um determinado processo é o principal objetivo deste diagrama.

## 6 Diagrama de Classes



O diagrama de classes é considerado por muitos autores como o mais importante e o mais utilizado diagrama da UML. Seu principal enfoque está em permitir a visualização das classes que irão compor o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do sistema se relacionam, se complementam e transmitem informações entre si. Este diagrama apresenta uma visão estática de como as classes estão organizadas, preocupando-se em definir a estrutura lógica das mesmas.

## 7 Conclusão

A maior dificuldade estava em fazer um Diagrama e só na implementação do próximo perceber que faltava uma funcionalidade ou classe no sistema. Isso poderia desestruturar o sistema mudando o pensamento lógico de algumas partes.

Além disso, tivemos dificuldade na implementação do Diagrama de Sequência, uma vez que não encontramos artigos relacionados que abrangia o tema. Desse modo, percebemos a complexidade de abstrair um problema real, e implementação de tal projeto.

## References

- [1] DEVMEDIA.**O Que é UML e Diagramas de Caso de Uso:** Introdução Prática à UML. Recuperado em 17 de outubro de 2020, <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>
- [2] DEVMEDIA.**Artigo SQL Magazine 64 - Utilizando UML.** Recuperado em 17 de outubro de 2020, <https://www.devmedia.com.br/artigo-sql-magazine-64-utilizando-uml/12665>
- [3] DEVMEDIA.**Diagrama de Classes UML.** Recuperado em 17 de outubro de 2020, <https://www.devmedia.com.br/diagrama-de-classes-uml/12251>