

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
SISTEMAS DE INFORMAÇÃO

Tales Félix

Padrão de Projeto - MVC

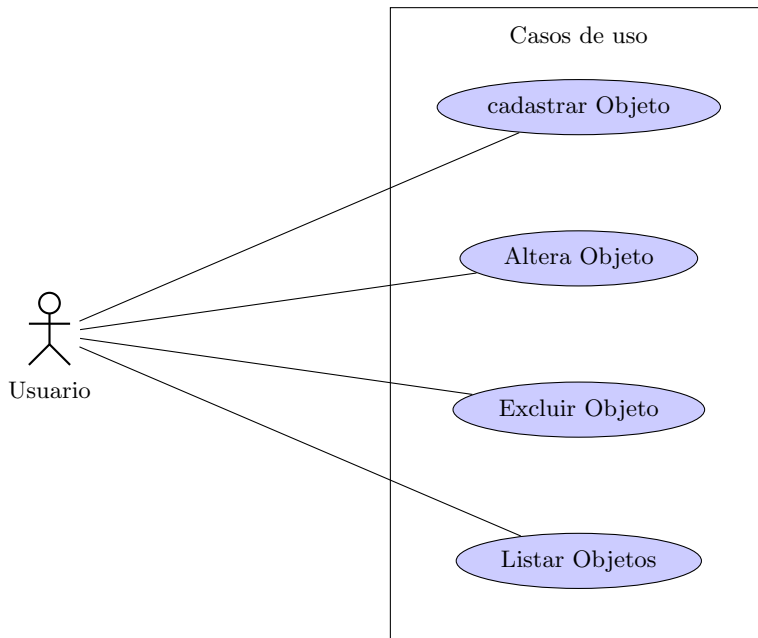
Docente: Eduardo Pelli.

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
```

Index

```
- nome: String  
- descricao: String  
- id: String  
- IdExcluir: String  
- IdAterar: Strig  
- nomeAlterar: String  
- descricaoAlterar: String  
  
+ EventEnviar()  
+ EventAterar()  
+ EventExcluir()  
+ EventListar()
```

Arquivo

```
- escrita: FileWriter  
- parser: JSONParser  
- scan: Scanner  
- gson: Gson
```

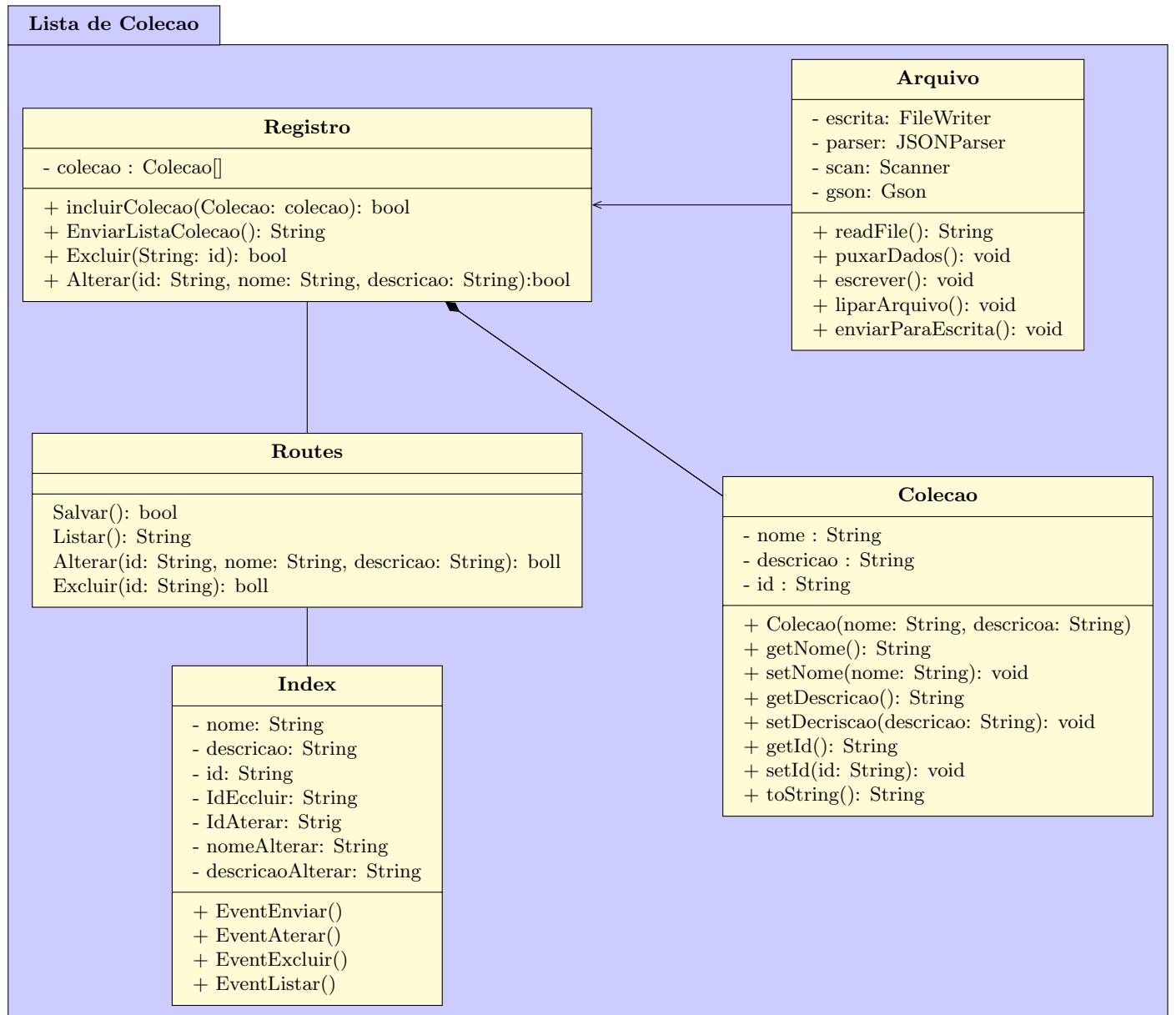
```
+ readFile(): String  
+ puxarDados(): void  
+ escrever(): void  
+ liparArquivo(): void  
+ enviarParaEscrita(): void
```

Colecao

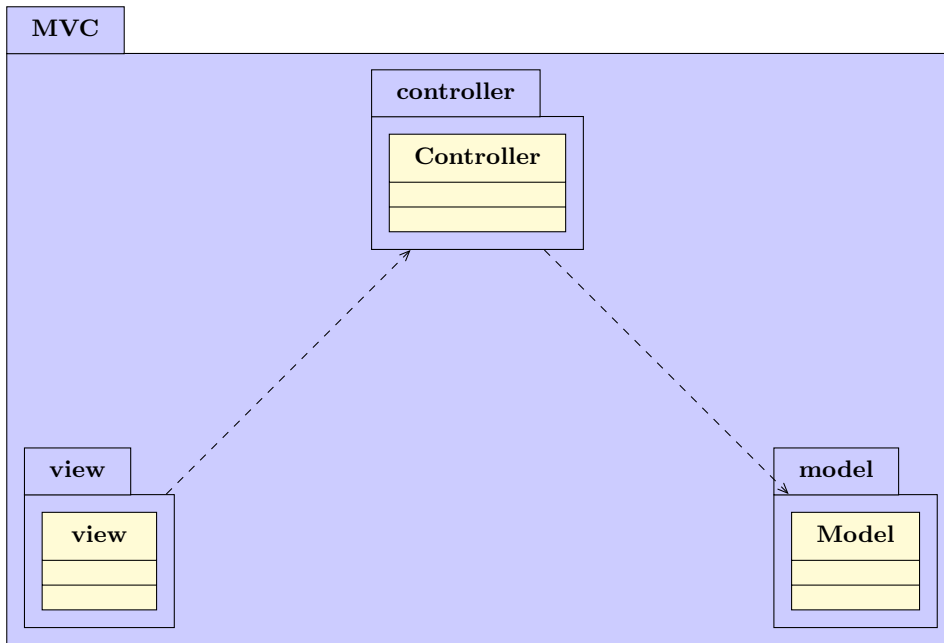
```
- nome : String  
- descricao : String  
- id : String  
  
+ Colecao(nome: String, descricao: String)  
+ getNome(): String  
+ setNome(nome: String): void  
+ getDescricao(): String  
+ setDecriscao(descricao: String): void  
+ getId(): String  
+ setId(id: String): void  
+ toString(): String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

5 Diagrama de Sequência

6 Diagrama de Classes Lista de Coleções



7 Diagrama de classe MVC



8 Conclusão

References