

# bookLoc

---

## Grupo:

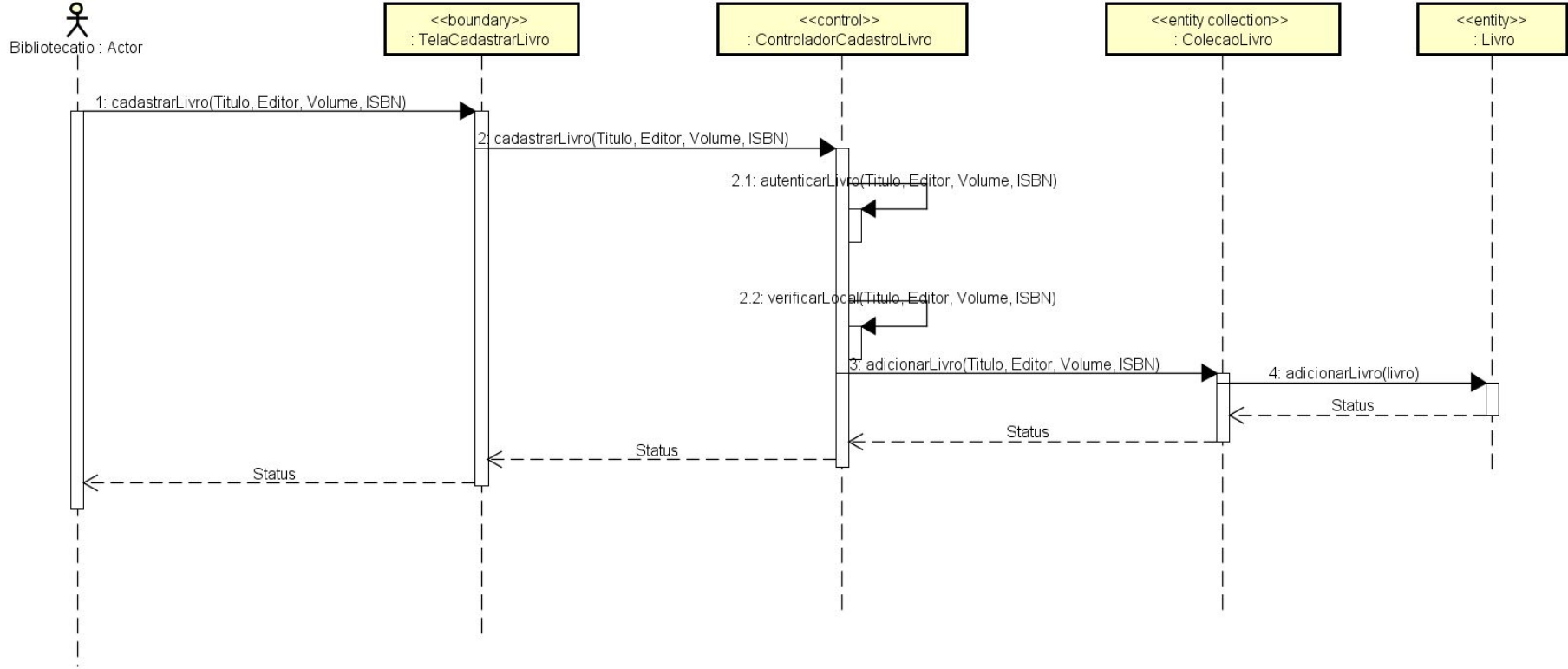
- José Antônio da Silva
- Moisés de Siqueira Campos Neto
- Rebeca Vasconcelos de Sa Alencar
- Walter Sobral Andrade

# Correções

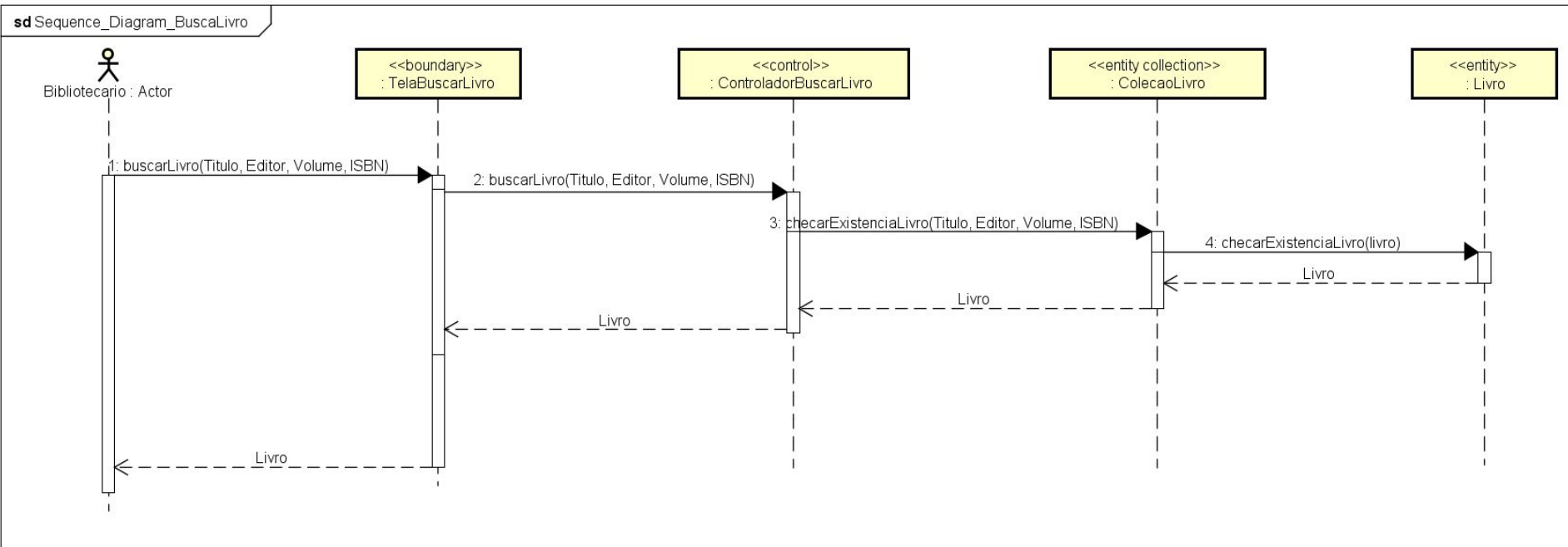
---

# Diagrama de Sequência (Cadastrar Livro)

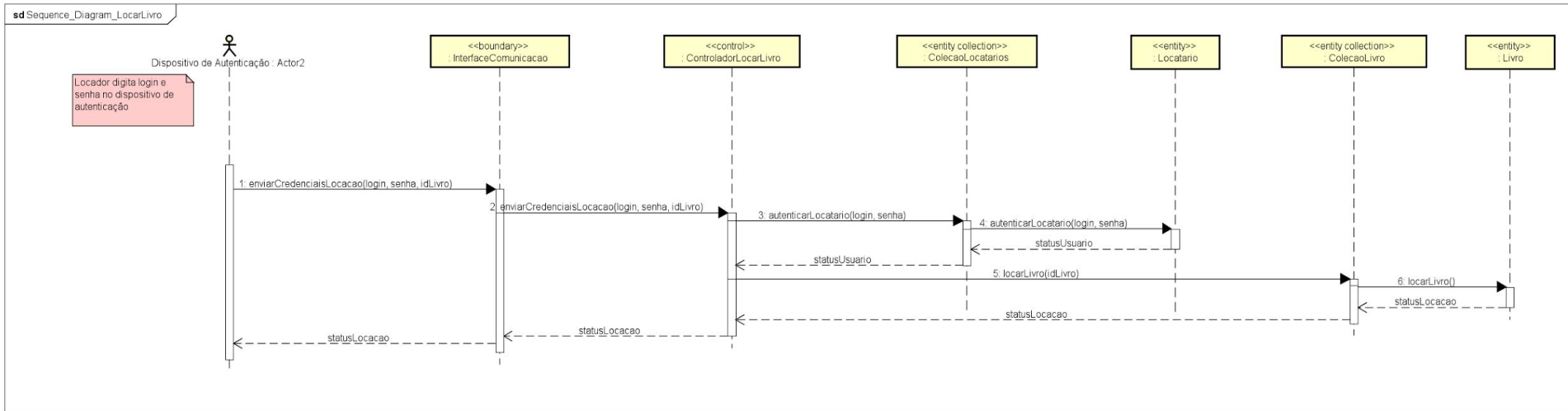
sd Sequence\_Diagram\_CadastrarLivro



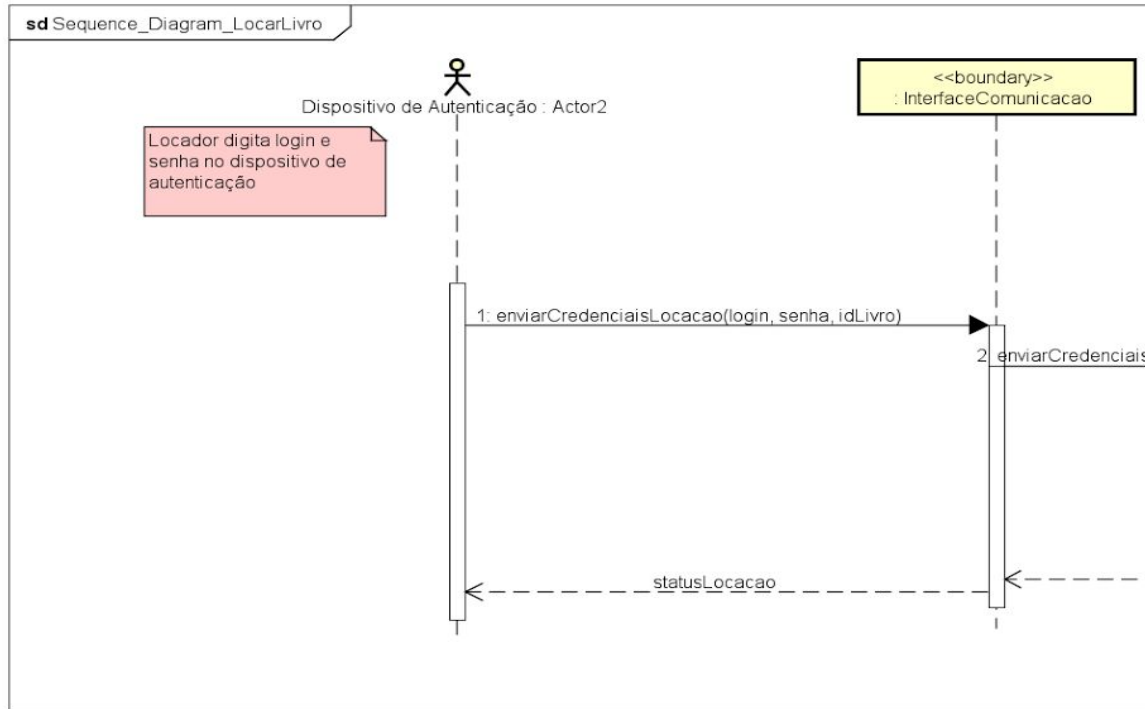
# Diagrama de Sequência (Buscar Livro)



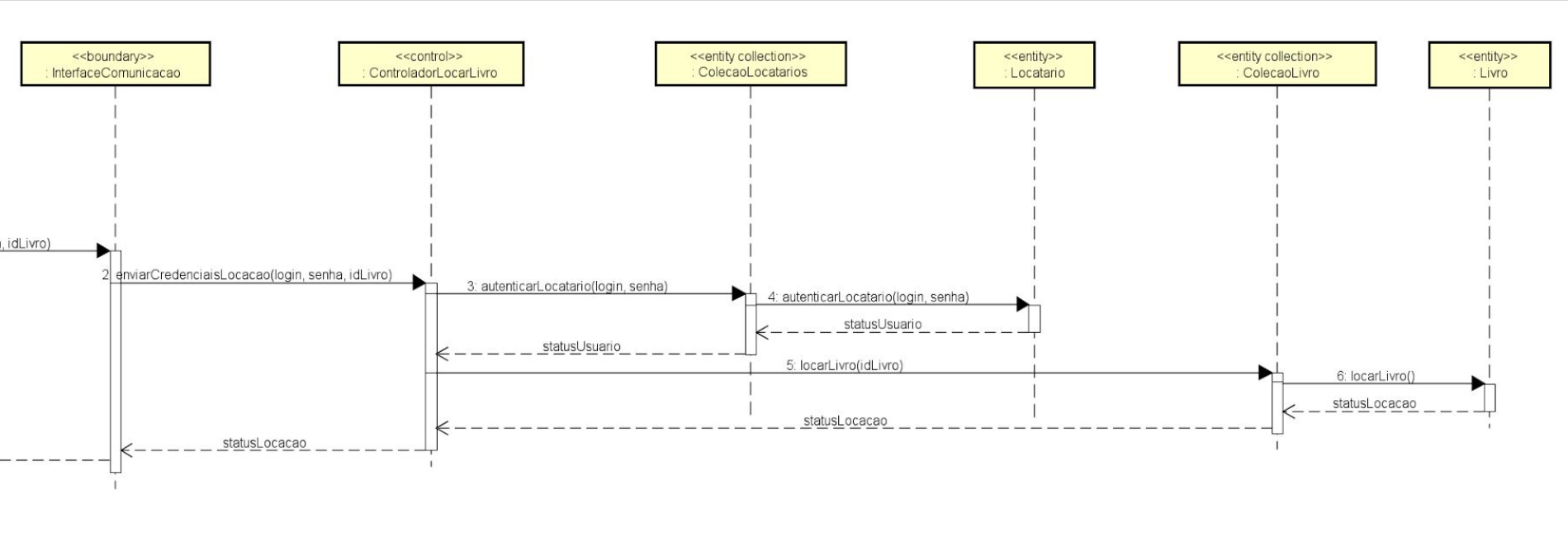
# Diagrama de Sequência (Locar Livro)



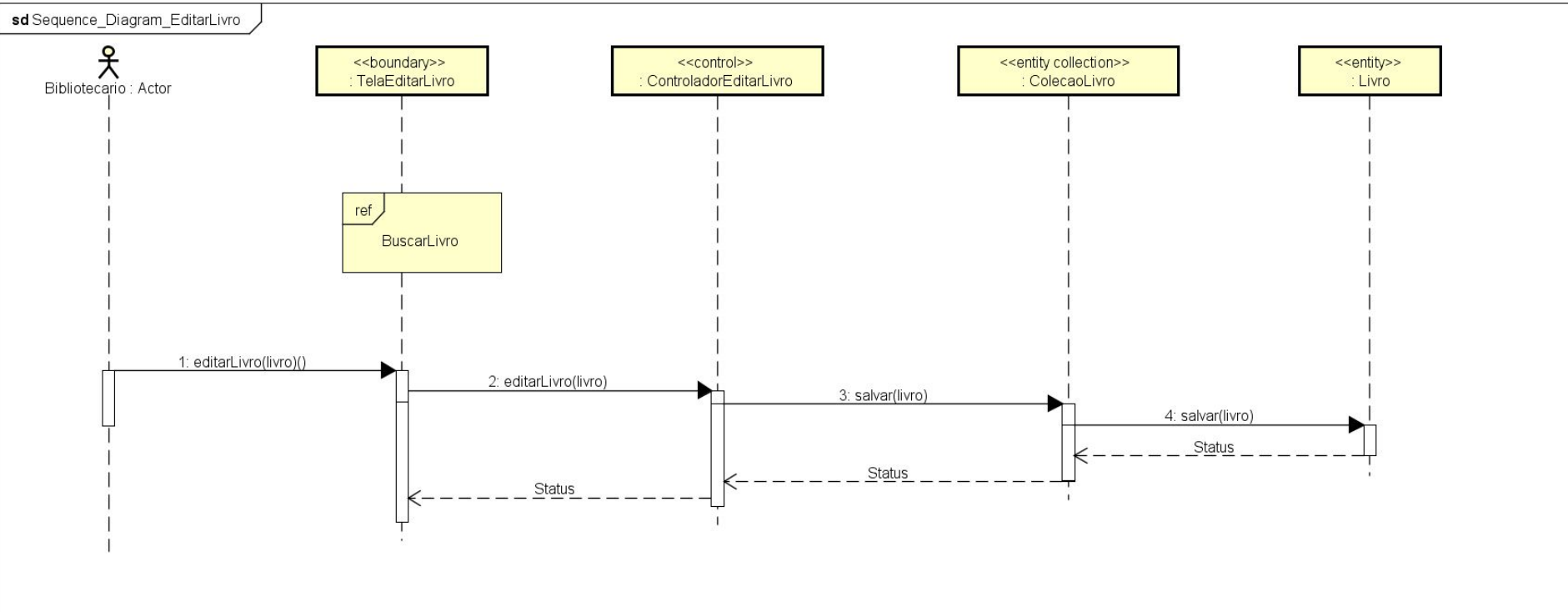
# Diagrama de Sequência (Locar Livro)



# Diagrama de Sequência (Locar Livro)

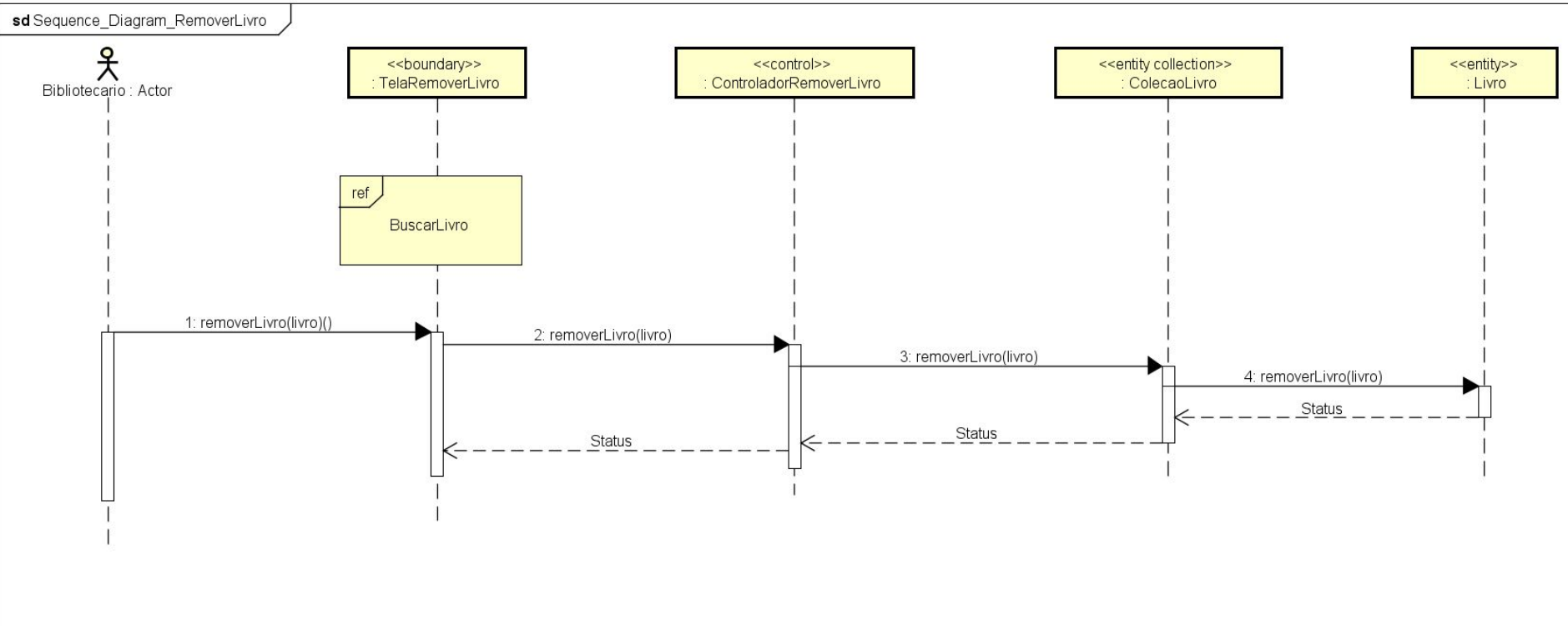


# Diagrama de Sequência (Editar Livro)





# Diagrama de Sequência (Remover Livro)



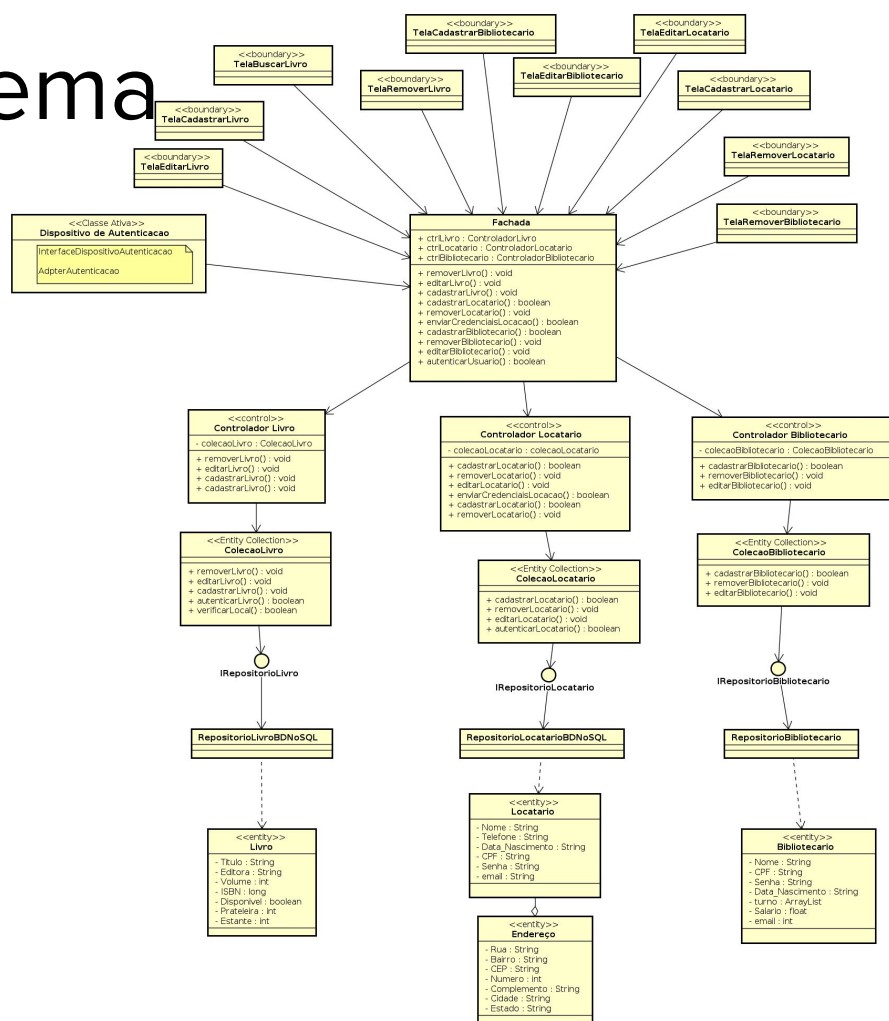
# Arquitetura do Sistema

---

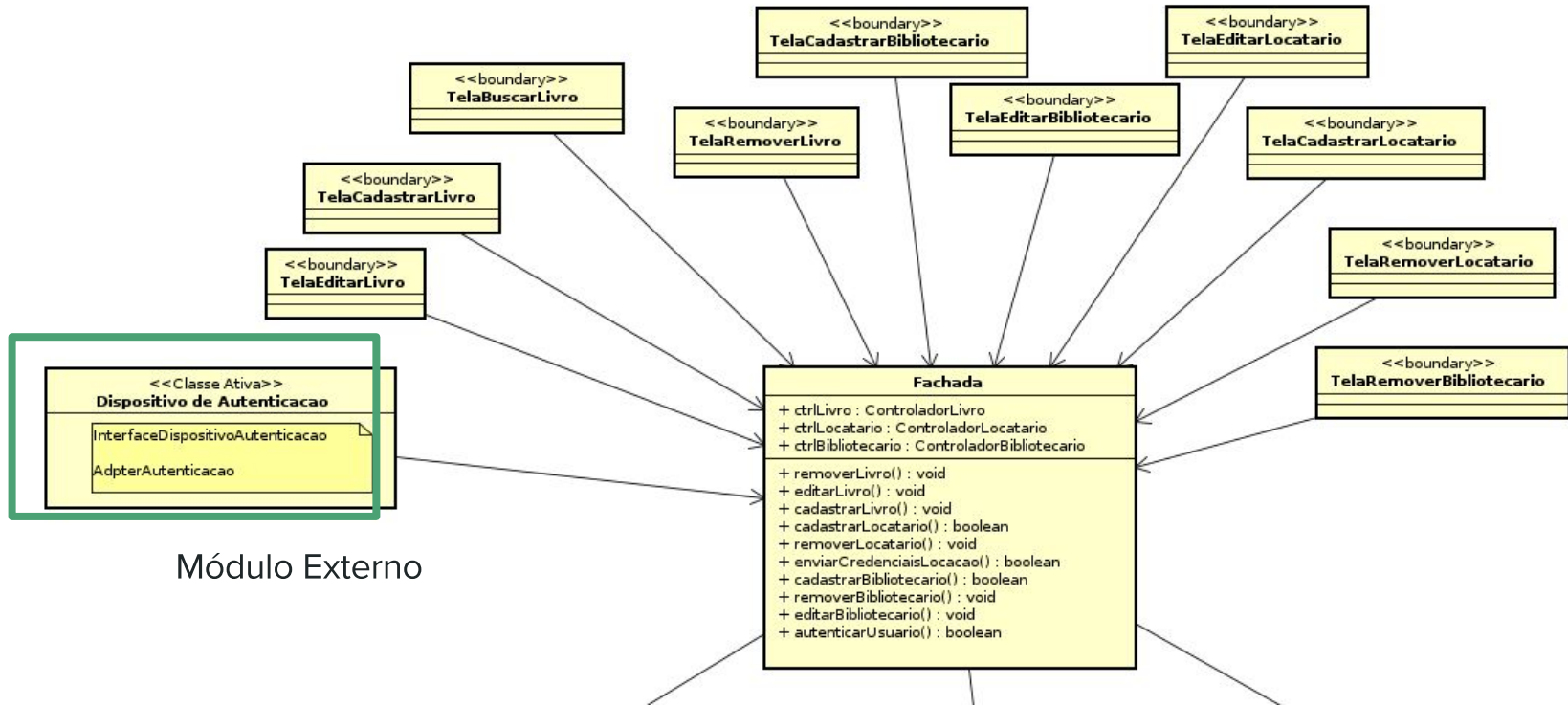
# Mapeamento das Classes de Análise

Classes de Análise	Elementos de Projeto
	Fachada
Tela Cadastrar Livro	Tela Cadastrar Livro
Tela Remover Livro	Tela Remover Livro
Tela Buscar Livro	Tela Buscar Livro
Tela Editar Livro	Tela Editar Livro
Controlador Cadastro Livro Controlador Buscar Livro Controlador Locar Livro Controlador Ediar Livro Controlador Remover Livro	Controlador Livro
Cadastrar Livro	ColecaoLivro IRepositorioLivro RepositorioLivroBDNoSql
Cadastrar Locatario	ColecaoLocatario IRepositorioLocatario RepositorioLocatarioBDNoSql
Cadastrar Bibliotecario	ColecaoBibliotecario IRepositorioBibliotecario RepositorioBibliotecarioBDNoSql

# Arquitetura do Sistema

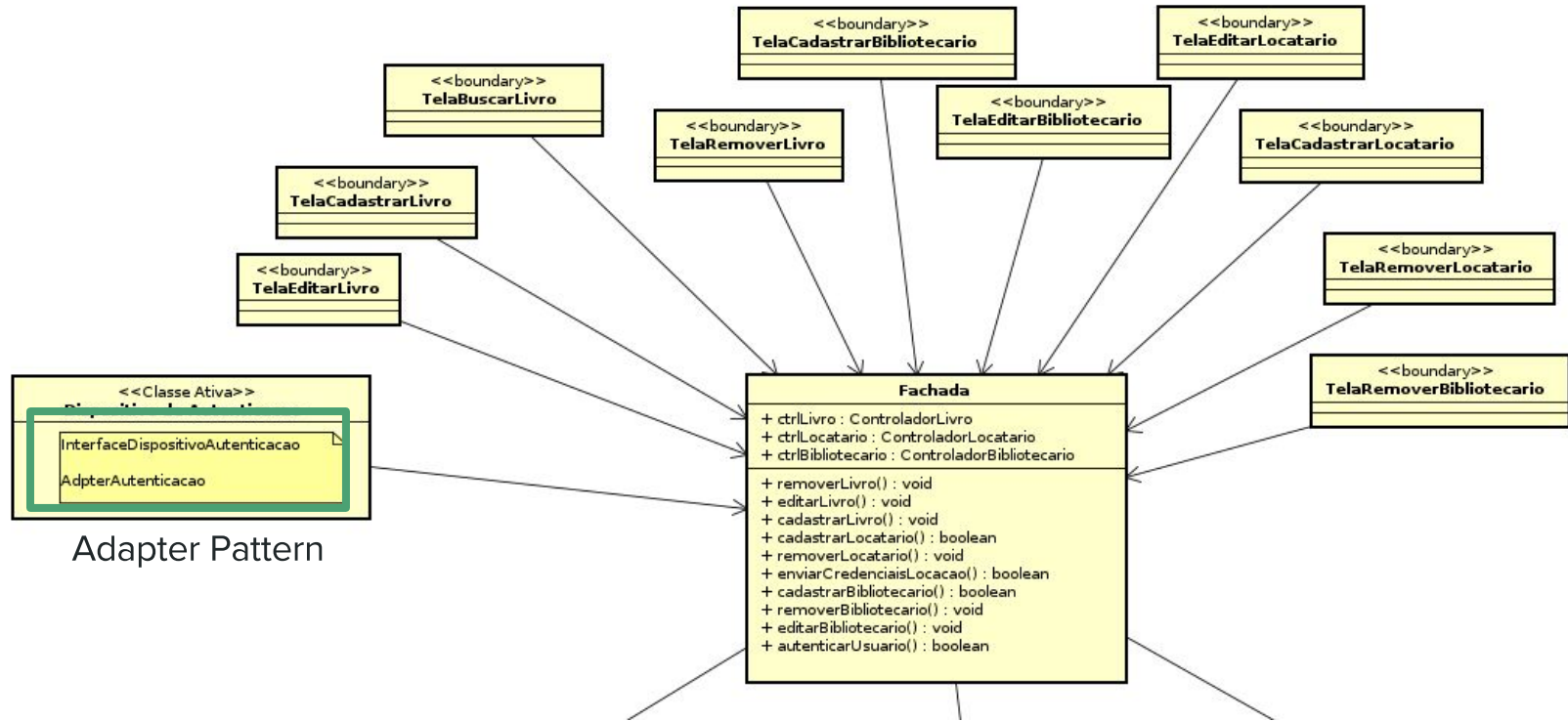


# Arquitetura do Sistema

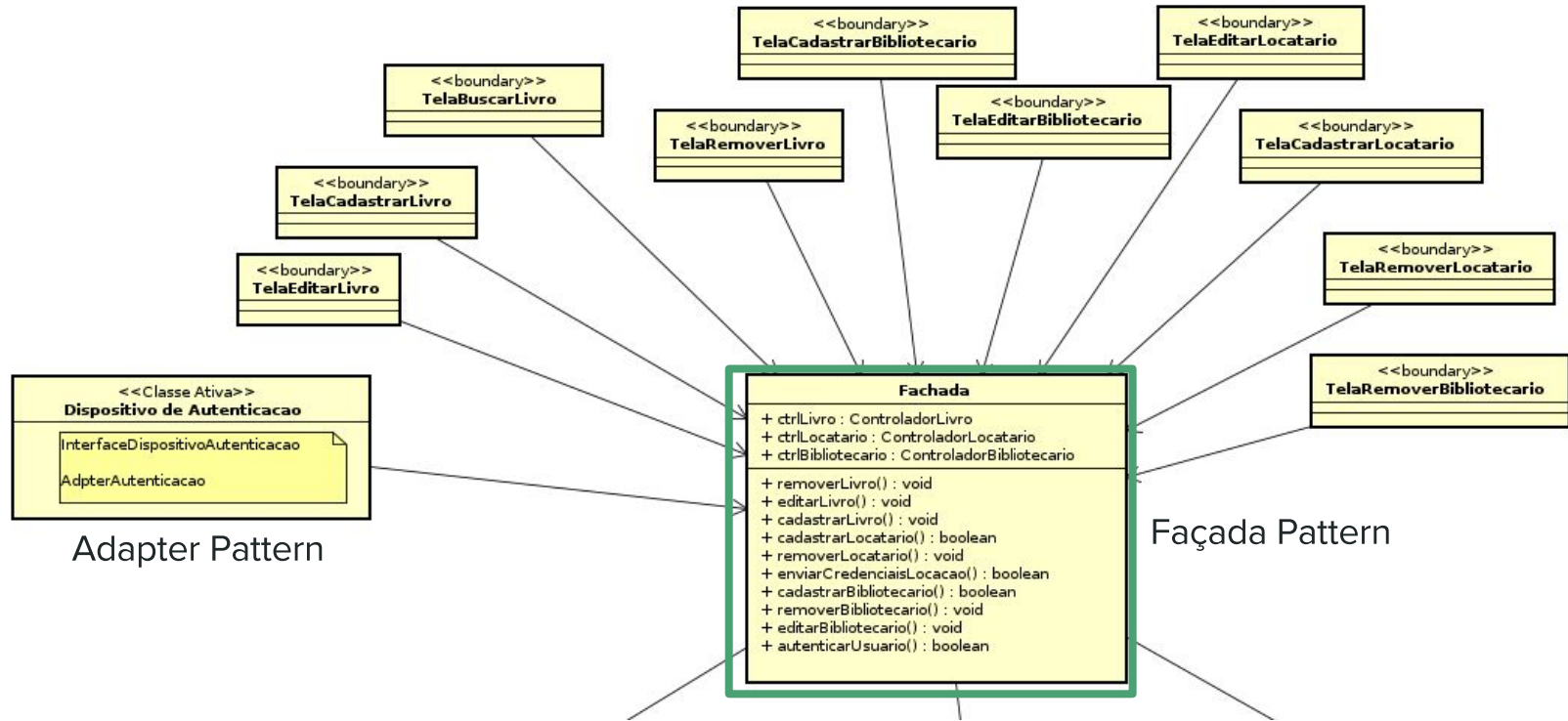


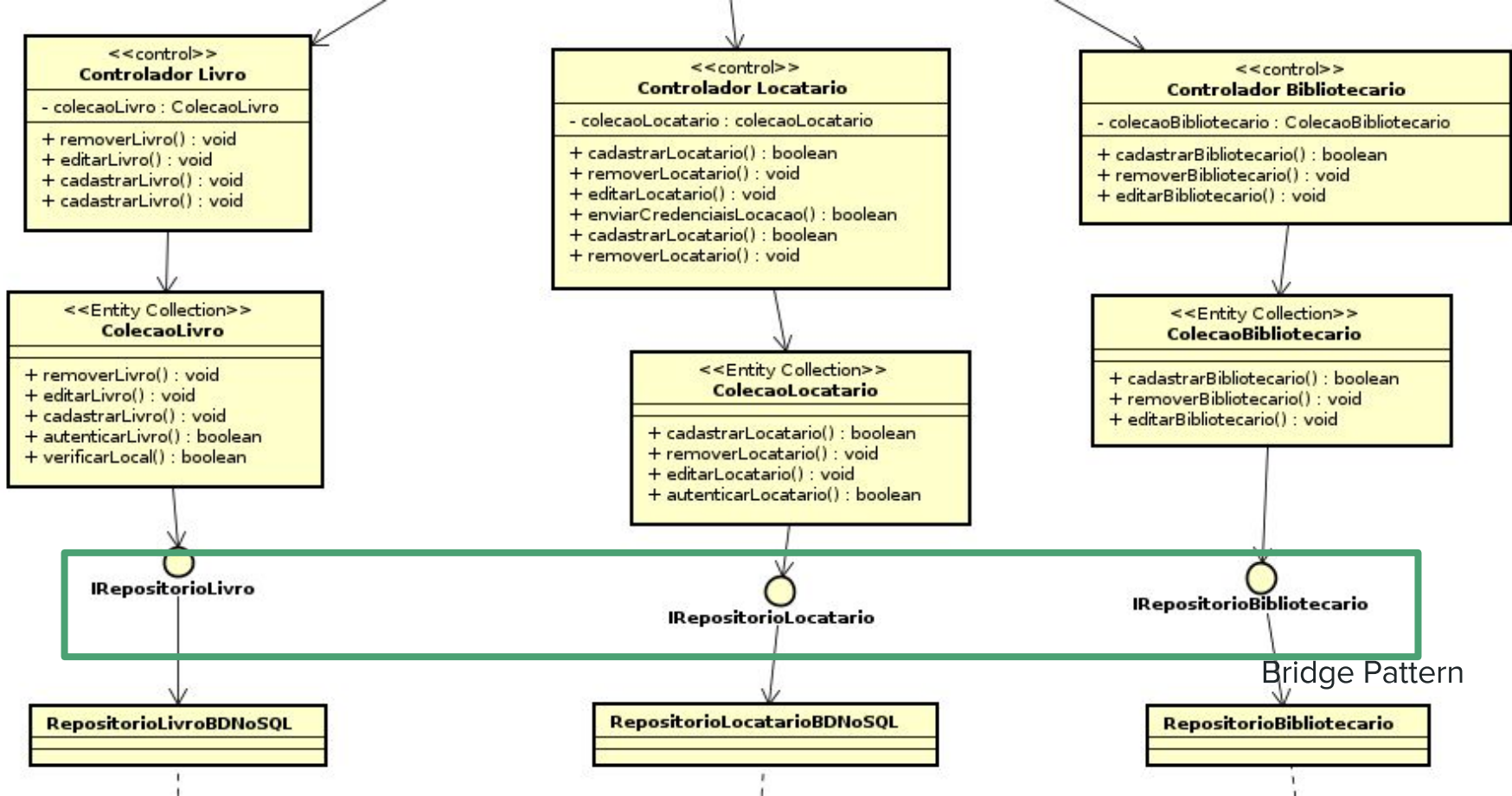
Módulo Externo

# Arquitetura do Sistema

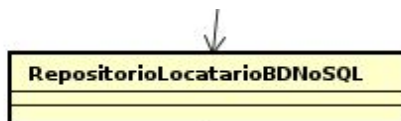


# Arquitetura do Sistema

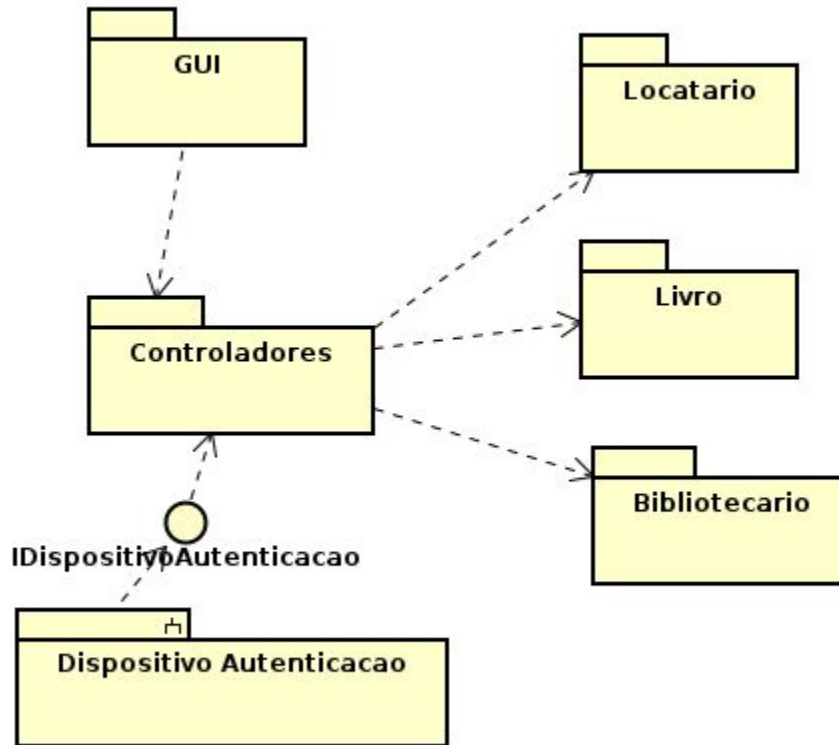








# Arquitetura do Sistema (Diagrama de Pacotes)



# Implementação

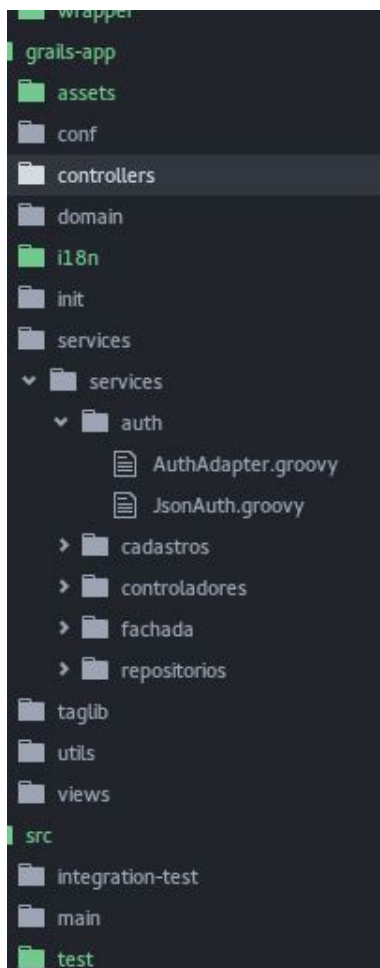
---

```
1 package bookloc
2
3 import services.fachada.Fachada
4
5 import static org.springframework.http.HttpStatus.*
6 import grails.transaction.Transactional
7
8 @Transactional(readOnly = true)
9 class BookController {
10
11     static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]
12
13     Fachada fachada = Fachada.getInstance()
14
15     def index(Integer max) {
16         params.max = Math.min(max ?: 10, 100)
17         respond Book.list(params), model:[bookCount: Book.count()]
18     }
19
20     def show(Book book) {
21         respond fachada.buscarBook(book)
22     }
23
24     def create() {
25         respond fachada.criarBook(params)
26     }
27
28     @Transactional
29     def save(Book book) {
30         try{
31             fachada.salvarBook(book)
```

```
class Fachada {  
  
    static constraints = {  
    }  
    private static Fachada instance;  
    private ControladorBook controladorBook  
    private ControladorUser controladorUser  
    private JsonAuth autenticador  
  
    private Fachada(){  
        controladorBook = new ControladorBook()  
        controladorUser = new ControladorUser()  
        autenticador = new JsonAuth()  
    }  
  
    public static Fachada getInstance(){  
        if(instance == null){  
            instance = new Fachada();  
        }  
        return instance;  
    }  
  
    // book  
    public Book buscarBook(Book book){  
        return controladorBook.buscarBook(book);  
    }  
    public Book criarBook(params){  
        controladorBook.criarBook(params);  
    }  
    public Book salvarBook(Book book){  
        controladorBook.salvarBook(book);  
    }  
    public void removerBook(Book book){  
        controladorBook.removerBook(book);  
    }  
    public List<Book> books(){  
        return controladorBook.books();  
    }  
    public HashMap autenticarUser(JSON json){  
        return autenticador.authuser(ison)
```

AuthController.groovy

```
1 package bookloc
2
3 import grails.converters.JSON
4 import grails.transaction.Transactional
5 import services.auth.JsonAuth
6 import services.fachada.Fachada
7
8 @Transactional(readOnly = true)
9 class AuthController {
10
11     Fachada fachada = Fachada().getInstance()
12
13     def authuser() {
14         JSON json = request.JSON
15         render fachada.autenticarUser(json) as JSON
16     }
17 }
18
```



```
1 package services.auth
2
3 import bookloc.Book
4 import bookloc.User
5 import grails.converters.JSON
6 import groovy.json.JsonSlurper
7 import services.repositorios.RepBook
8 import services.repositorios.RepUser
9
10 class JsonAuth implements AuthAdapter{
11
12     RepBook repBook = new RepBook()
13     RepUser repUser = new RepUser()
14
15     final HashMap authuser(JSON json){
16         def slurper = new JsonSlurper()
17         def jsonMap = slurper.parseText(json.toString())
18         def rentedBooks = []
19
20         Book book = repBook.findByISBN(jsonMap.isbn)
21         if(book == null)
22             return [status:"FAILED",message:"No such book with isbn $jsonMap.isbn"]
23
24         User user = repUser.findByName(jsonMap.username)
25         if(user == null)
26             return [status:"FAILED",message:"No such user with name $jsonMap.username"]
27
28         if(!isNotNull(user.rentedBooks)){
```

```
class ControladorBook {  
  
    CadastroBook cadastroBook  
  
    public ControladorBook() {  
        cadastroBook = new CadastroBook();  
    }  
  
    public Book buscarBook(Book book){  
        cadastroBook.recuperarBook(book)  
    }  
  
    public Book criarBook(params){  
        cadastroBook.criarBook(params)  
    }  
  
    public Book salvarBook(Book book){  
        cadastroBook.salvarBook(book)  
    }  
  
    public void removerBook(Book book){  
        cadastroBook.removerBook(book)  
    }  
  
    public List<Book> books(){  
        cadastroBook.recuperarBooks()  
    }  
}
```



ers

s

ices

auth

cadastros

CadastroBook.groovy

CadastroUser.groovy

controladores

fachada

repositorios

ion-test

```
7 class CadastroBook {
8
9     public IRep<Book> repBook;
10
11     public CadastroBook(){
12         repBook = new RepBook();
13     }
14
15     public Book salvarBook(Book book){
16         if (book == null)
17             throw new NoSuchElementException()
18
19         if (book.hasErrors())
20             throw new IllegalArgumentException()
21
22         repBook.save(book);
23     }
24
25     public Book criarBook(params){
26         repBook.create(params);
27     }
28
29     public Boolean removerBook(Book book){
30         if (book == null)
31             throw new NoSuchElementException()
32         repBook.delete(book);
33     }
34
35     public Book recuperarBook(Book book){
36         return repBook.find(book);
37     }
```

```
public class RepBook implements IRep<Book>{

    public List<Book> all(){
        Book.all
    }

    public findByISBN(String bookISBN){
        return Book.find{isbn == bookISBN}
    }

    public Book find(Book book) {
        return book
    }

    public Book create(params){
        new Book(params)
    }

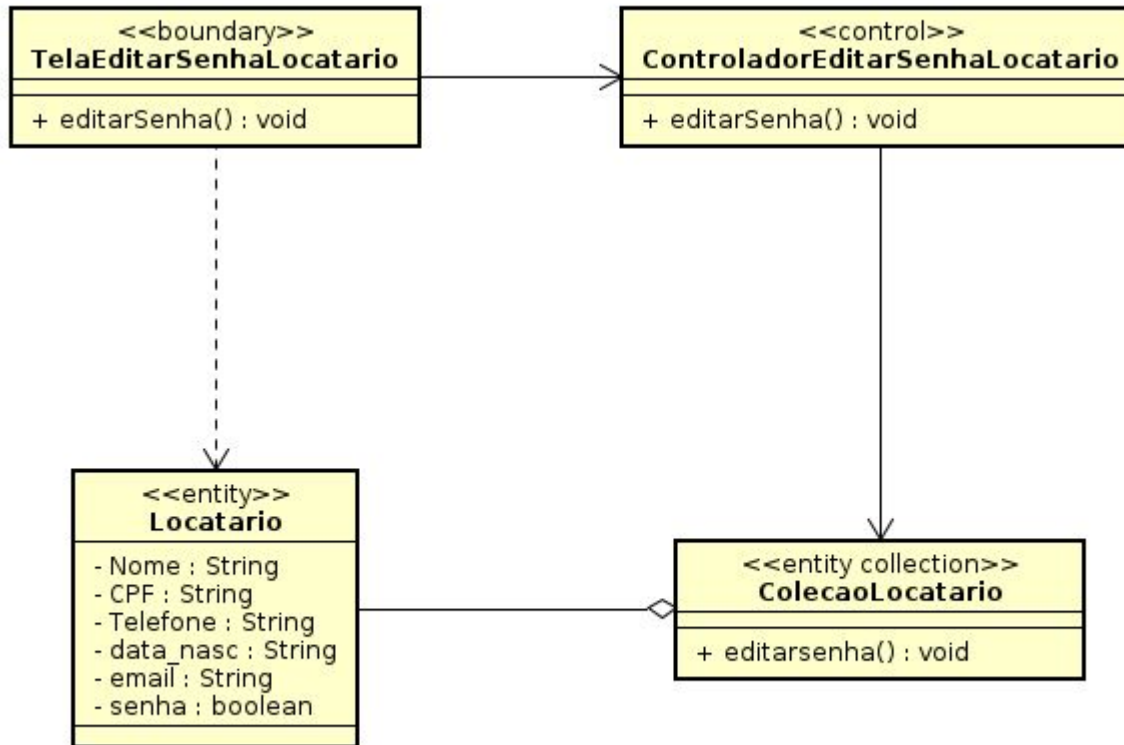
    public Book save(Book book){
        book.save flush:true
    }

    public void delete(Book book){
        book.delete flush:true
    }
}
```

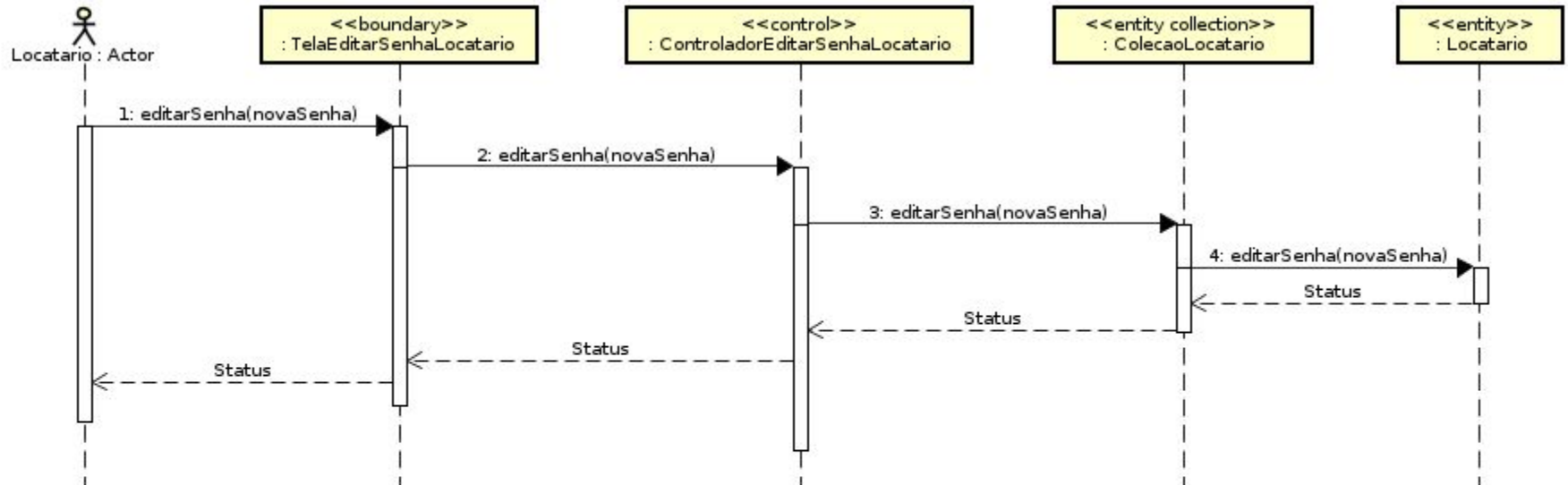
# Projeto de um caso de uso

---

# Diagrama de Classe (Editar Senha)



# Diagrama de Sequência (Editar Senha)



SOA

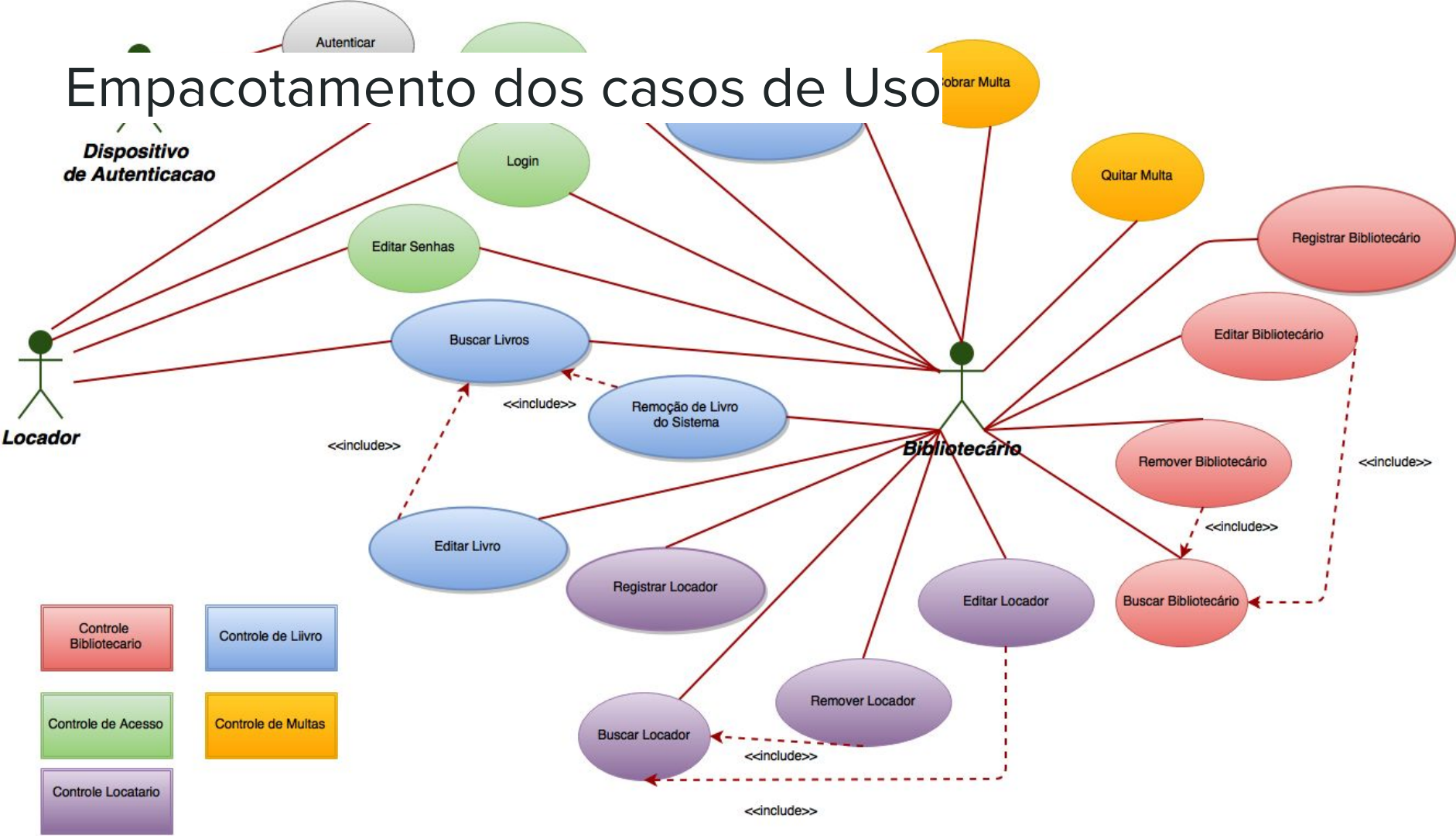
---

# Roteiro

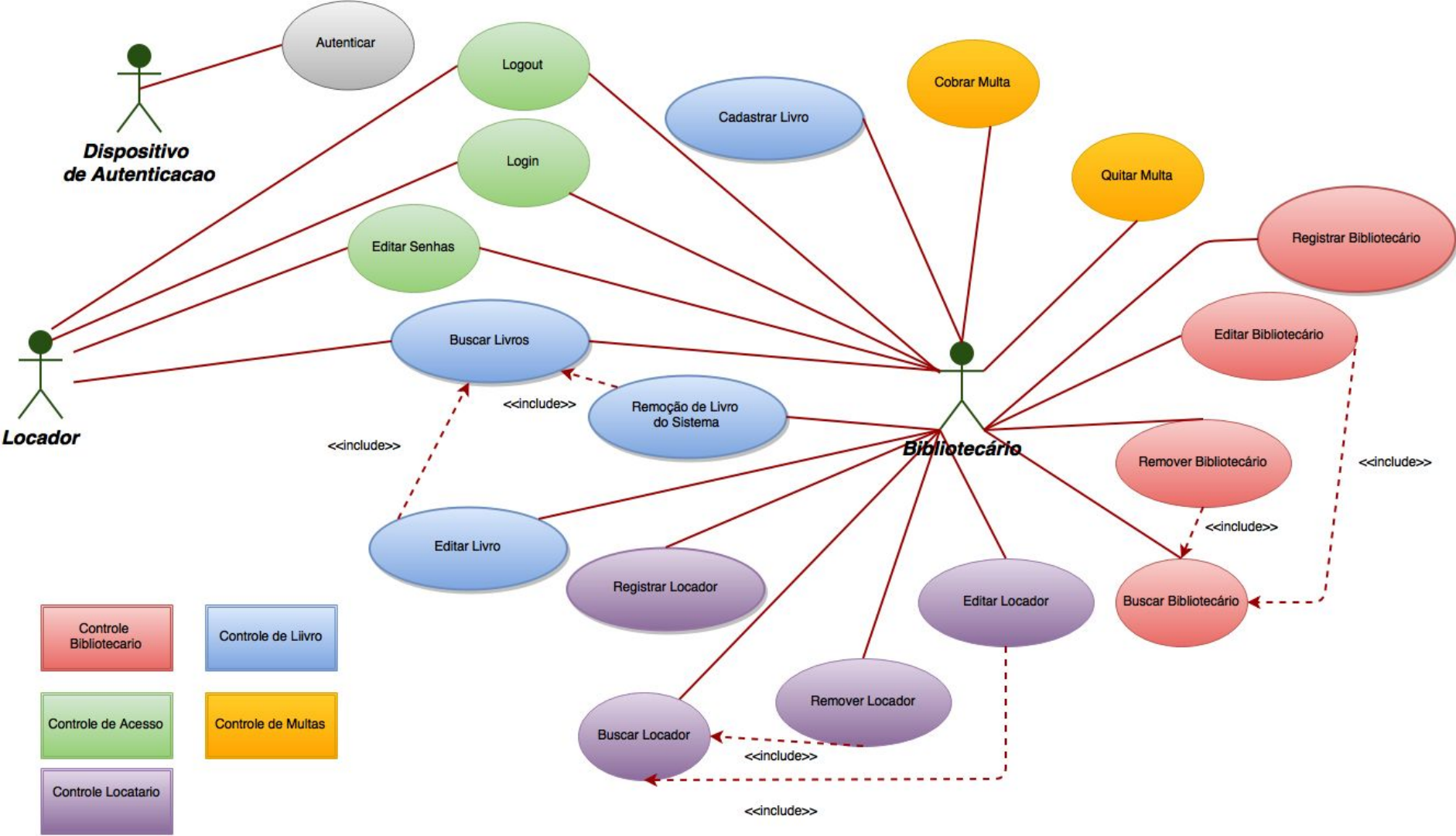
- Empacotamento dos casos de Uso
- Modelo de Informação do Negócio
- Modelo Navegacional
- Protótipo de Interface Gráfica
- Arquitetura dos Serviços
- Modelo de Interação dos Serviços
- Modelo de Informação Refinado
- Diagrama de Componentes dos Serviços (nível de análise)
- Diagrama de Componentes do Sistema
- Projeto detalhado de 3 telas Front-end e 3 componentes Back-end

# Empacotamento dos casos de Uso

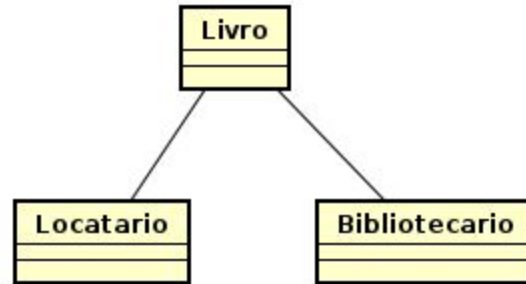
Dispositivo de Autenticacao



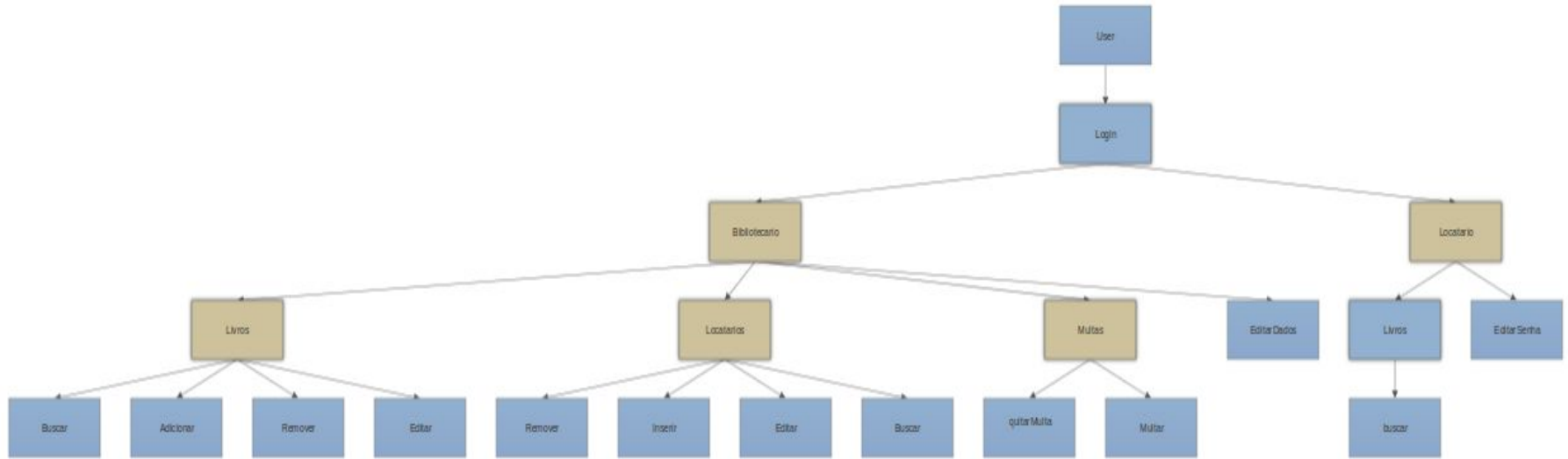




# Modelo de Informação



# Modelo Navegacional



# Protótipo de Interface Gráfica

Cadastro

Inserir livro

Remover livro

Locar livro

Buscar livro

## Cadastrar novo livro

Insira as informações do novo livro

Nome

**Editora**

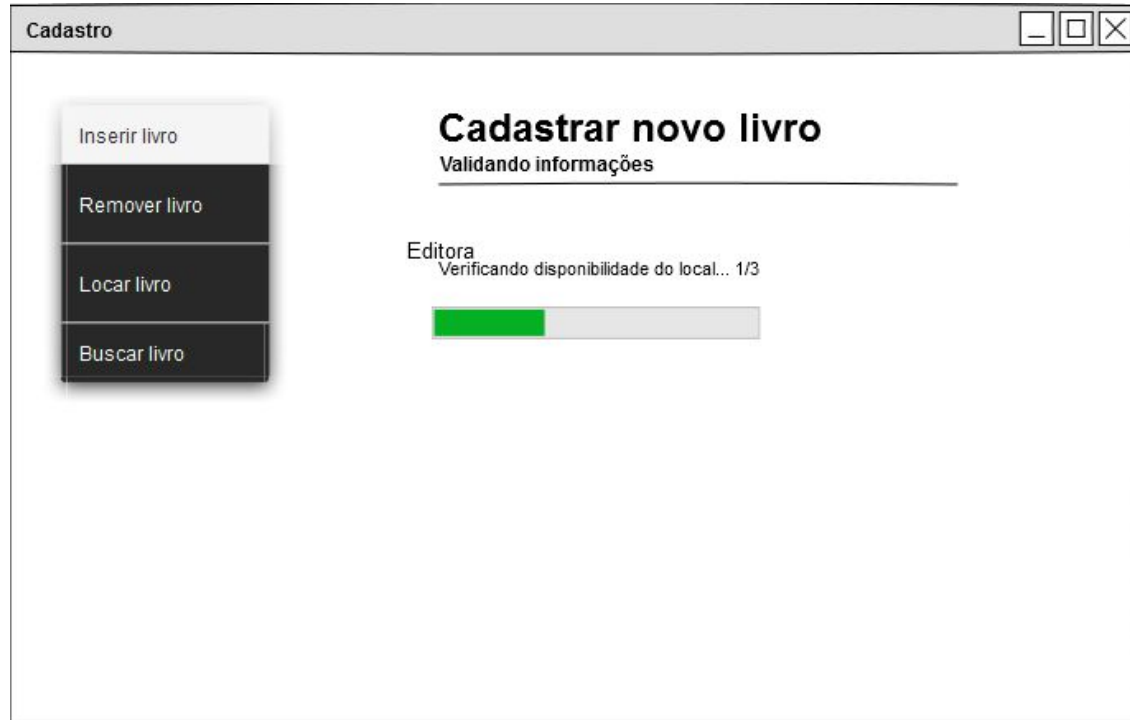
Volume

ISBN

Prateleira

Estante

# Protótipo de Interface Gráfica



# Protótipo de Interface Gráfica



# Protótipo de Interface Gráfica



# Protótipo de Interface Gráfica

Locar

Inserir livro

Remover livro

Locar livro

Buscar livro

## Locar livro

Credenciais

Locando "Signal Processing and Linear Systems"

Entre com suas credenciais:

Login:

Senha:

Login



# Protótipo de Interface Gráfica



# Protótipo de Interface Gráfica

Remover

Inserir livro

Remover livro

Locar livro

Buscar livro

## Remover livro

Busque pelo livro que deseja remover

Buscar livro por: ISBN

013168728X

Buscar

Detailed description: This is a wireframe of a graphical user interface for a library system. The window is titled 'Remover' and features a sidebar on the left with four menu items: 'Inserir livro', 'Remover livro', 'Locar livro', and 'Buscar livro'. The main content area is titled 'Remover livro' and includes a subtitle 'Busque pelo livro que deseja remover'. Below this, there is a search section with the label 'Buscar livro por:' followed by a dropdown menu currently set to 'ISBN'. A text input field contains the ISBN '013168728X', and a 'Buscar' button is positioned below it. The window also has standard minimize, maximize, and close window controls in the top right corner.

# Protótipo de Interface Gráfica

Remover

Inserir livro

Remover livro

Locar livro

Buscar livro

## Remover livro

Busque pelo livro que deseja remover

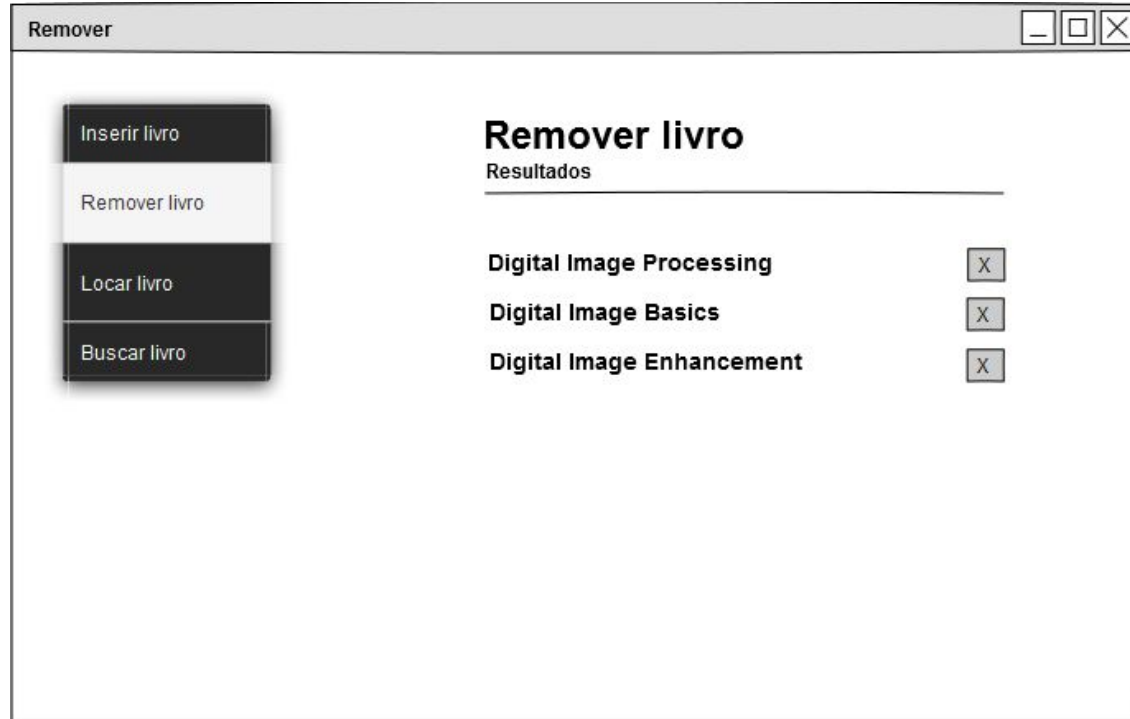
Buscar livro por: Nome

Digital Image

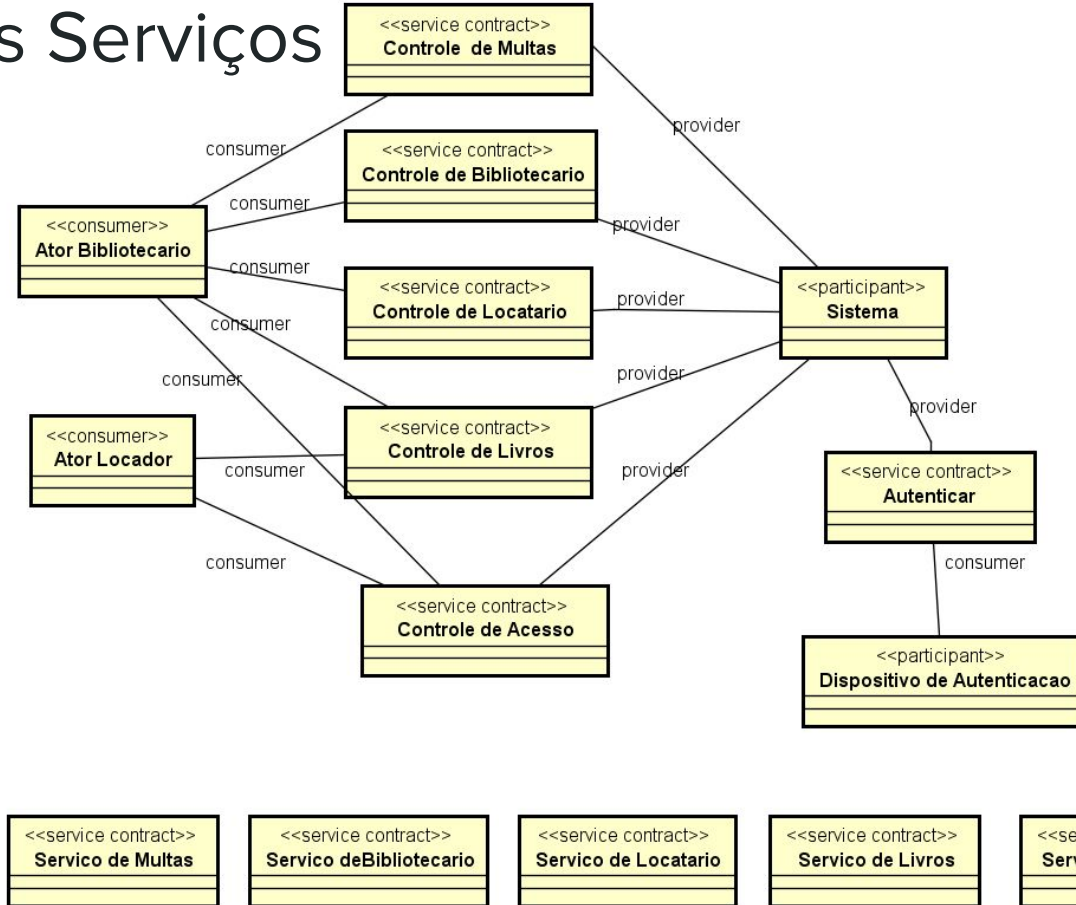
Buscar

Detailed description: This is a wireframe of a graphical user interface for a library system. The window is titled 'Remover' and has standard window controls (minimize, maximize, close) in the top right corner. On the left side, there is a vertical menu with four items: 'Inserir livro', 'Remover livro', 'Locar livro', and 'Buscar livro'. The 'Remover livro' item is highlighted with a white background. The main content area on the right is titled 'Remover livro' and contains the instruction 'Busque pelo livro que deseja remover'. Below this is a search form with a label 'Buscar livro por:' followed by a dropdown menu currently set to 'Nome'. Underneath the dropdown is a text input field containing the text 'Digital Image'. At the bottom of the search area is a 'Buscar' button.

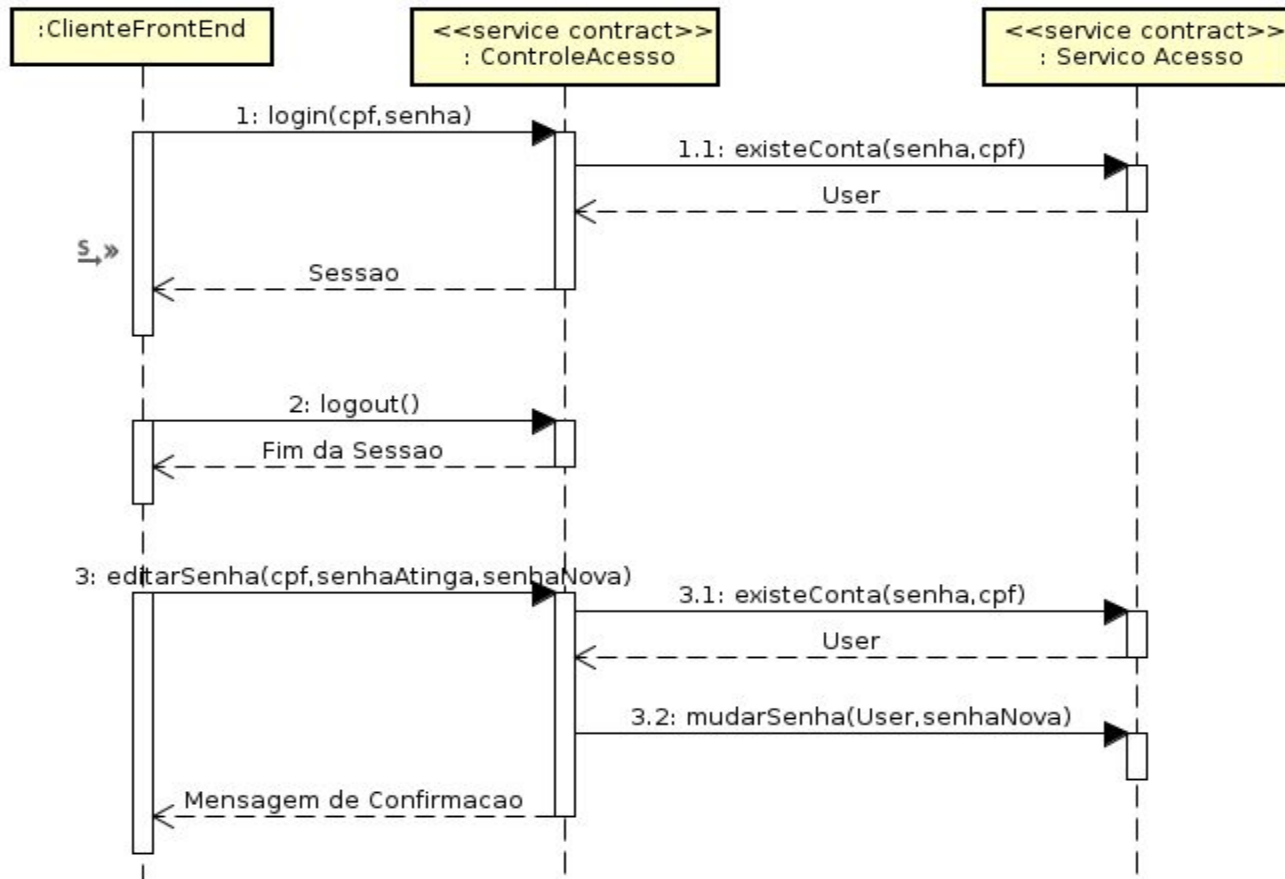
# Protótipo de Interface Gráfica



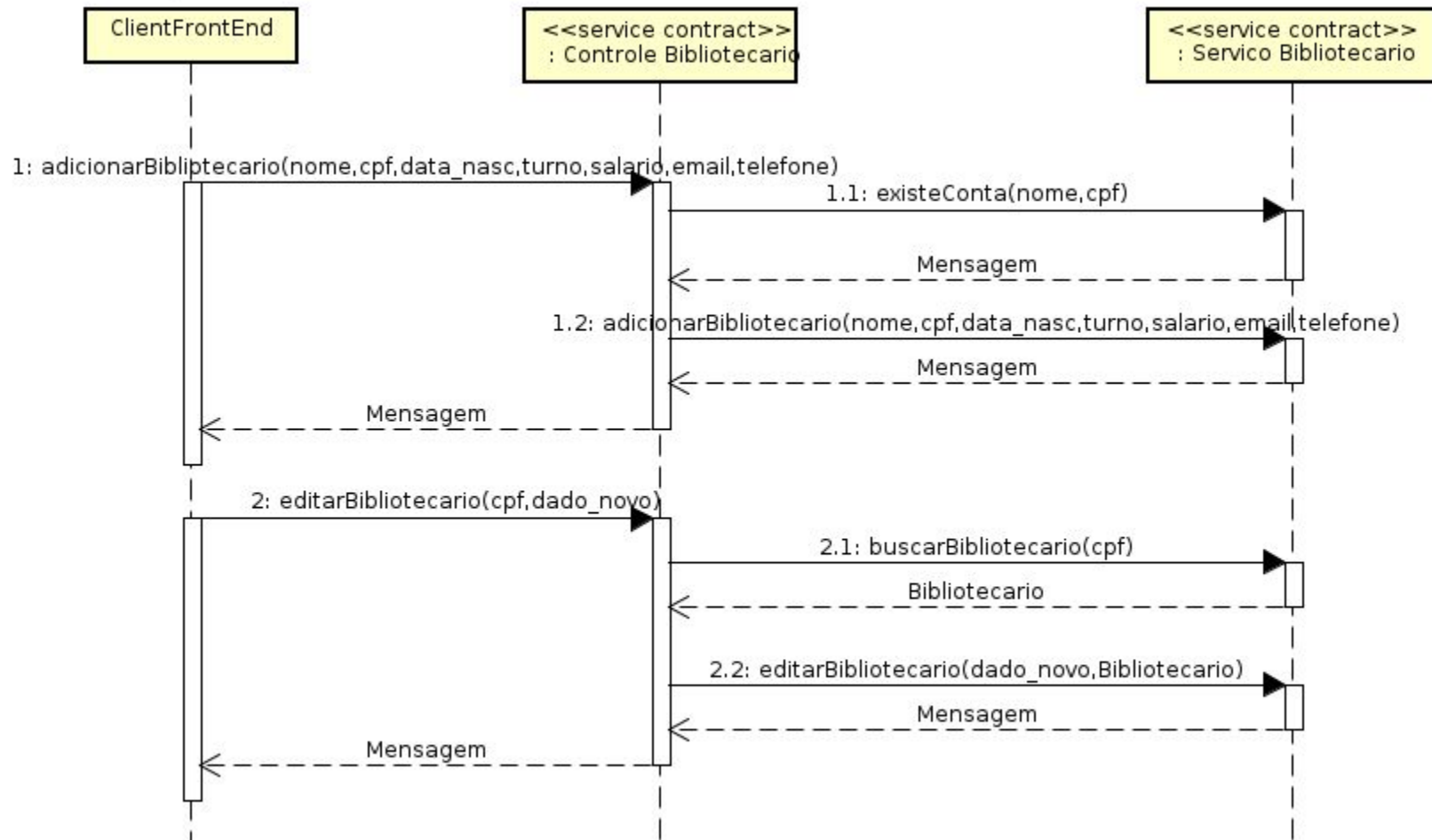
# Arquitetura dos Serviços



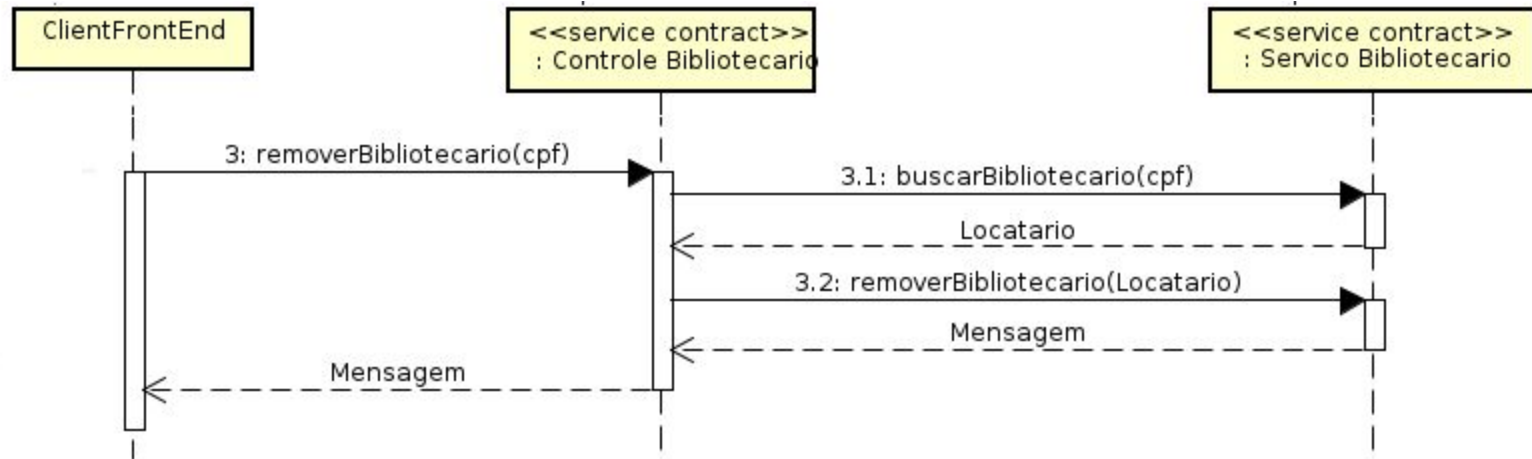
# Modelo de Interação dos Serviços - Acesso



# Modelo de Interação dos Serviços - Bibliotecario

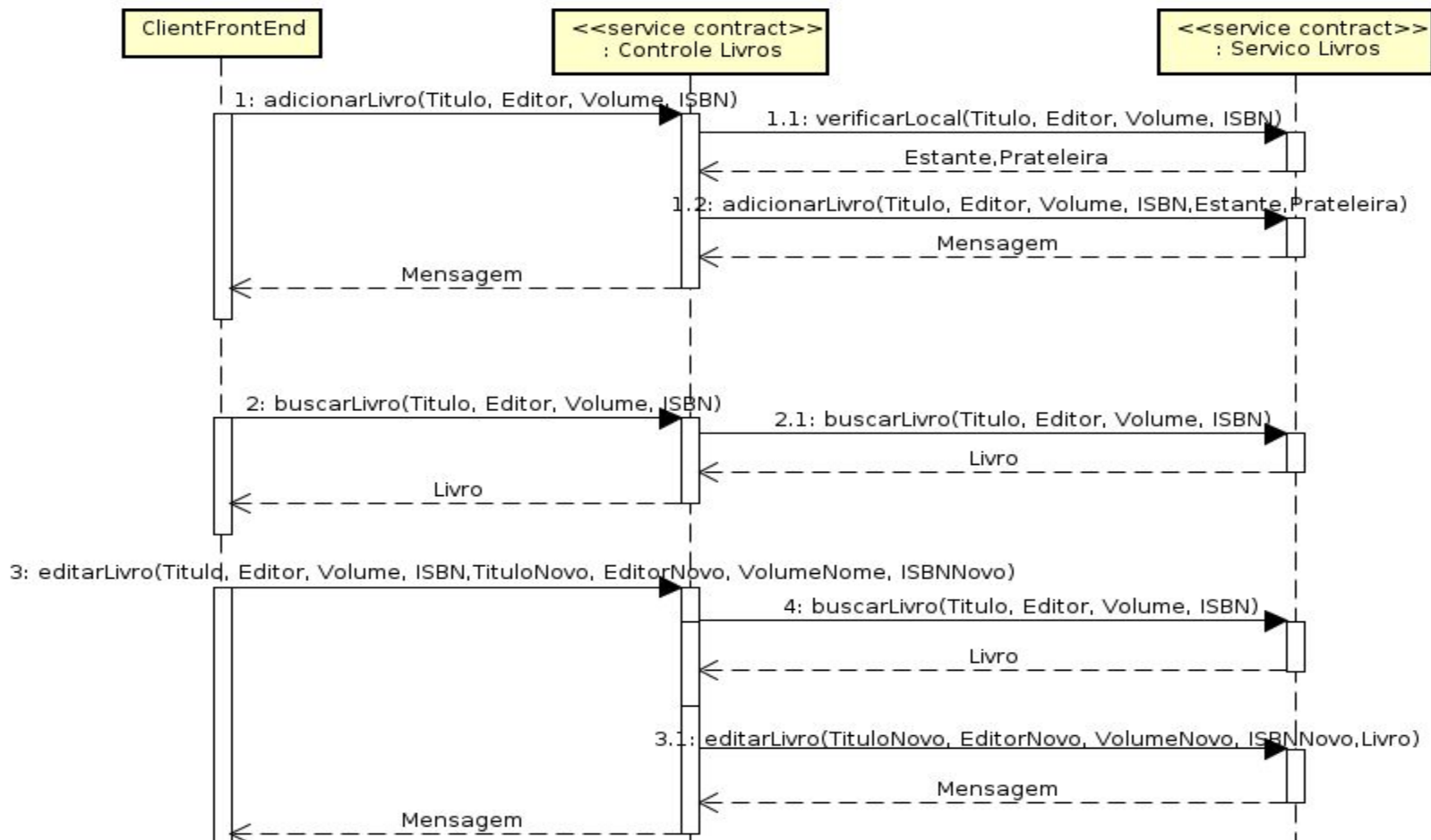


# Modelo de Interação dos Serviços - Bibliotecario

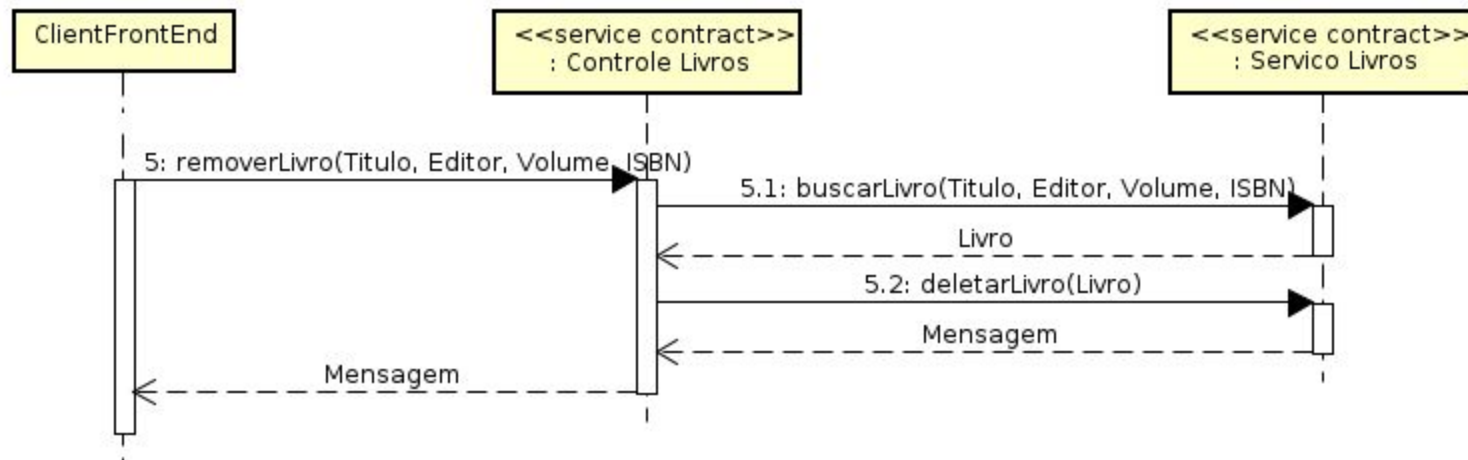




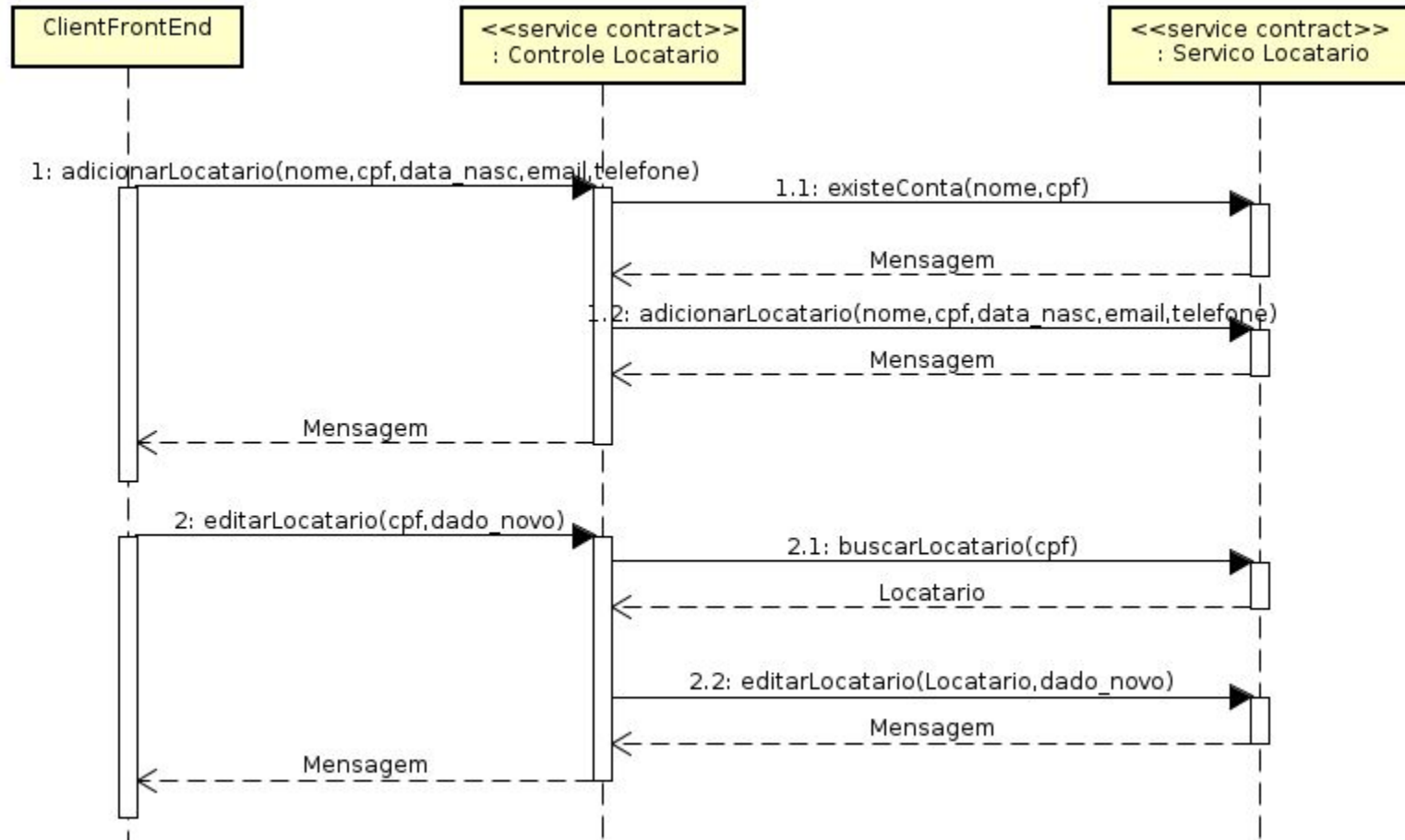
# Modelo de Interação dos Serviços - Livro



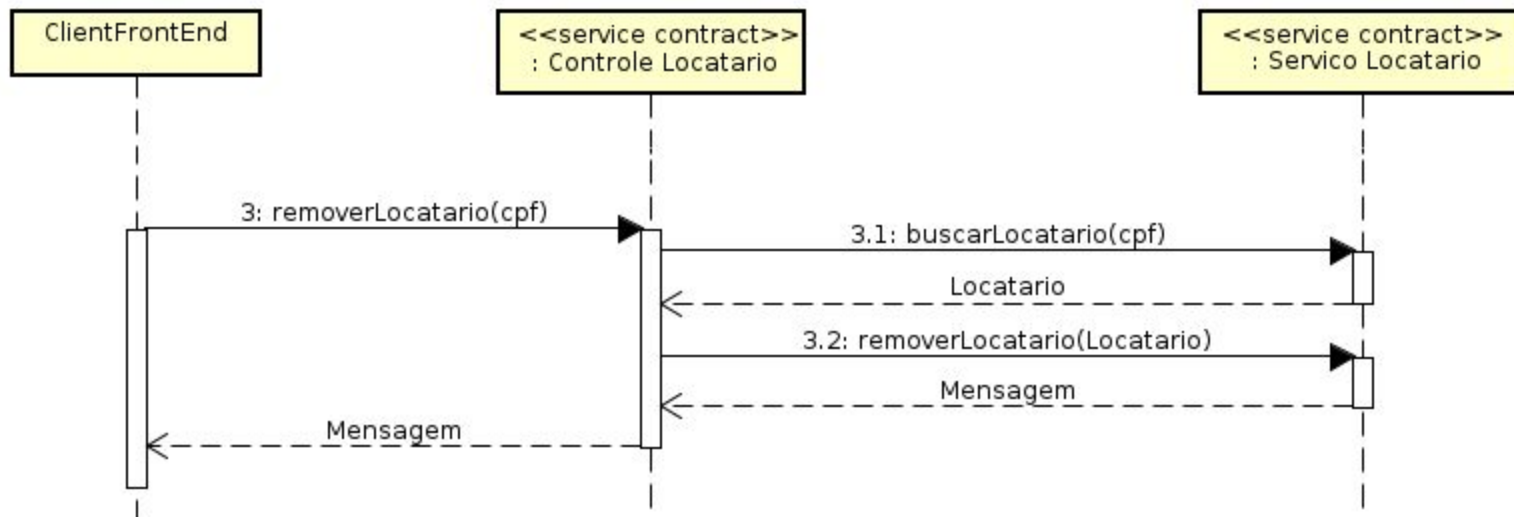
# Modelo de Interação dos Serviços - Livro



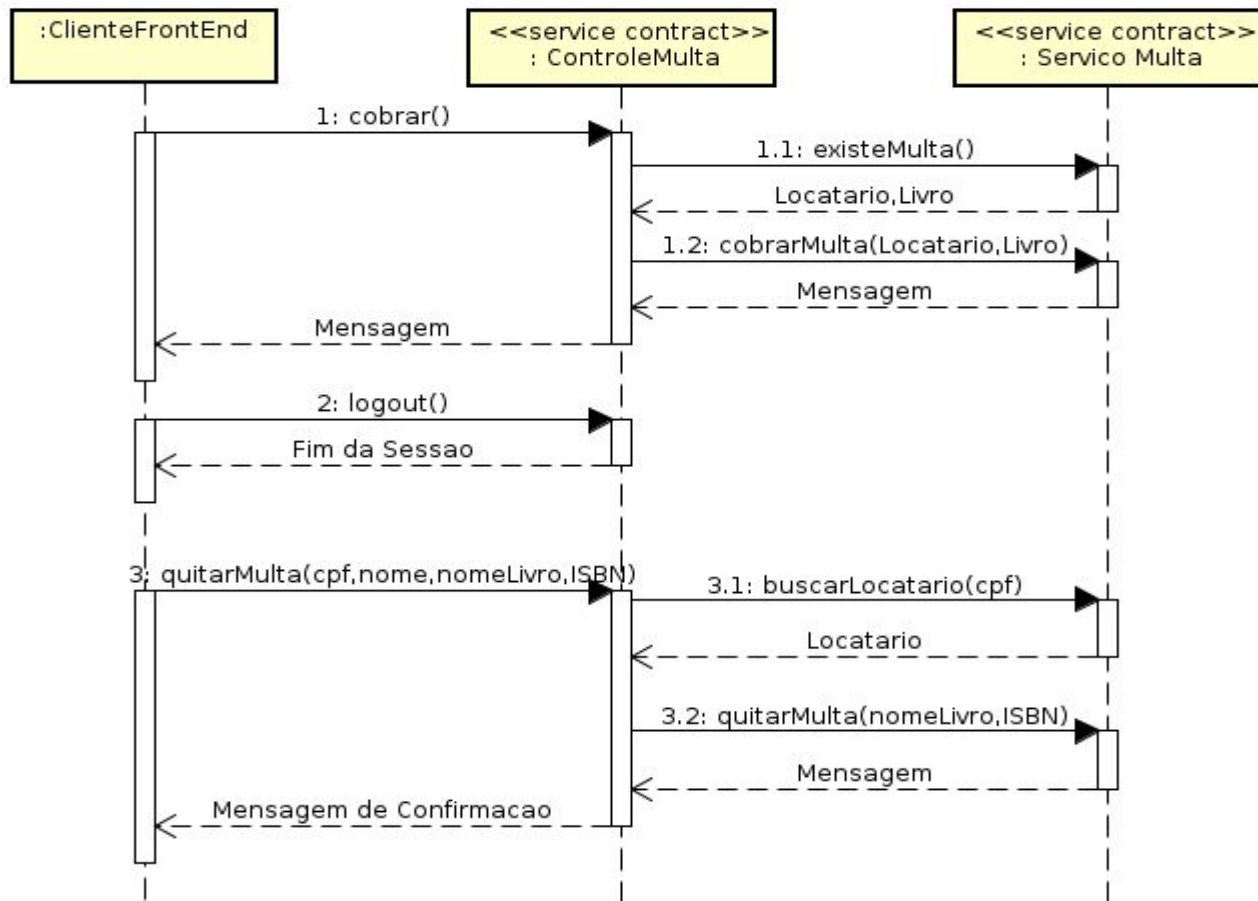
# Modelo de Interação dos Serviços - Locatario



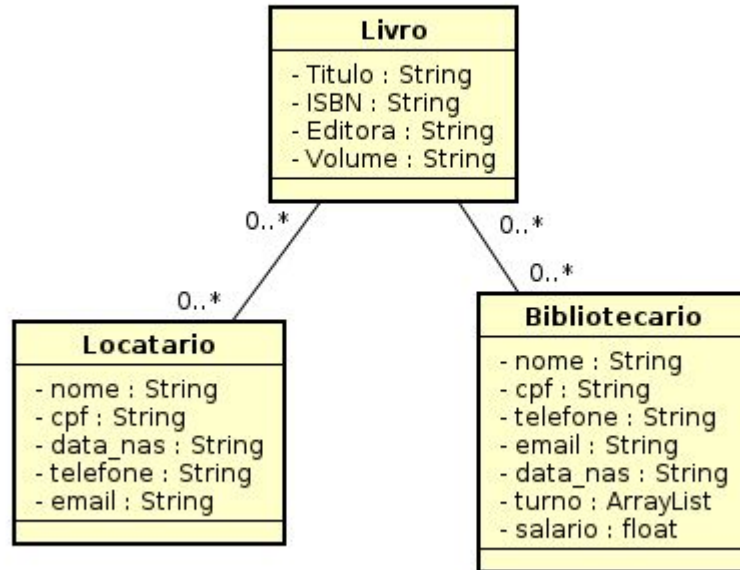
# Modelo de Interação dos Serviços - Locatario



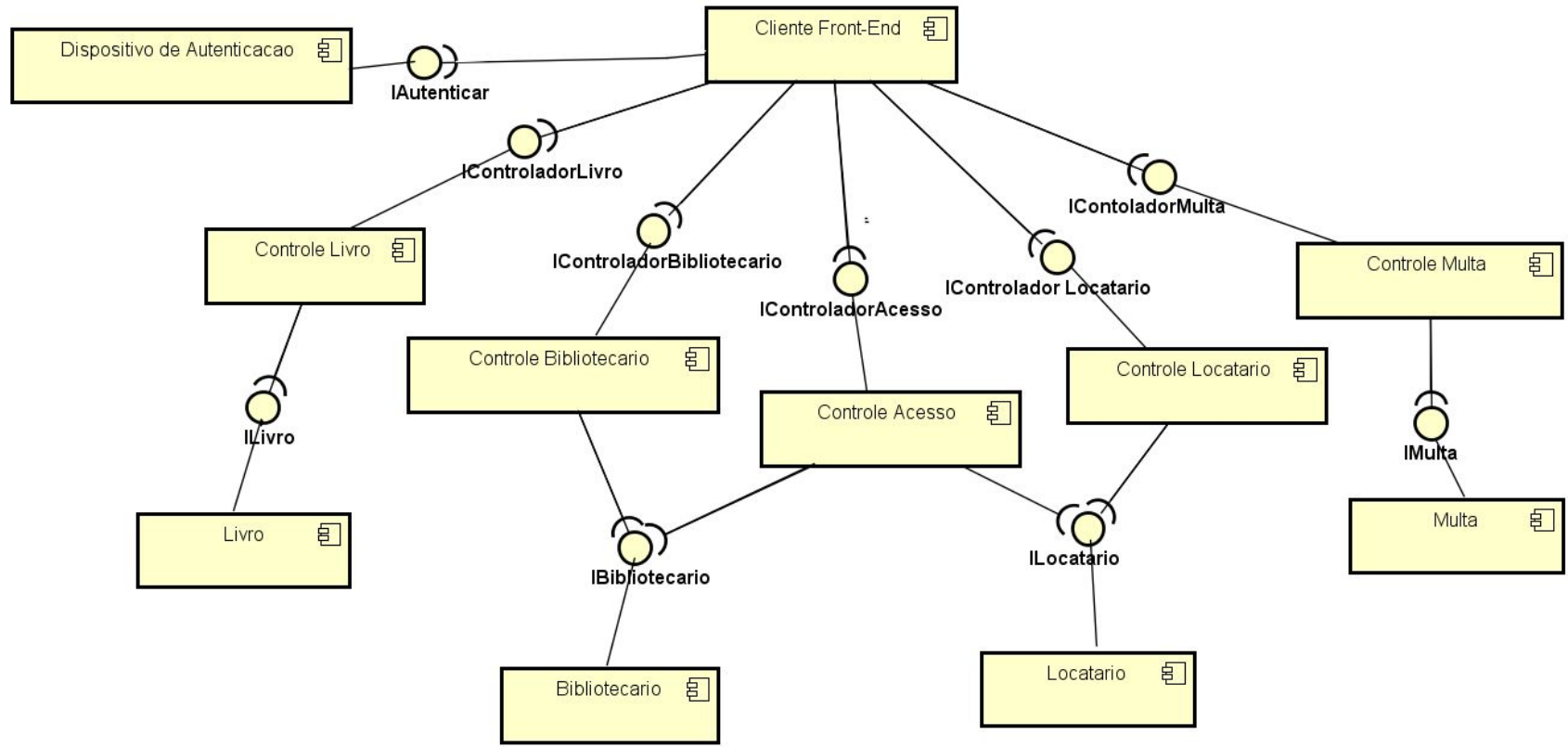
# Modelo de Interação dos Serviços - Multa



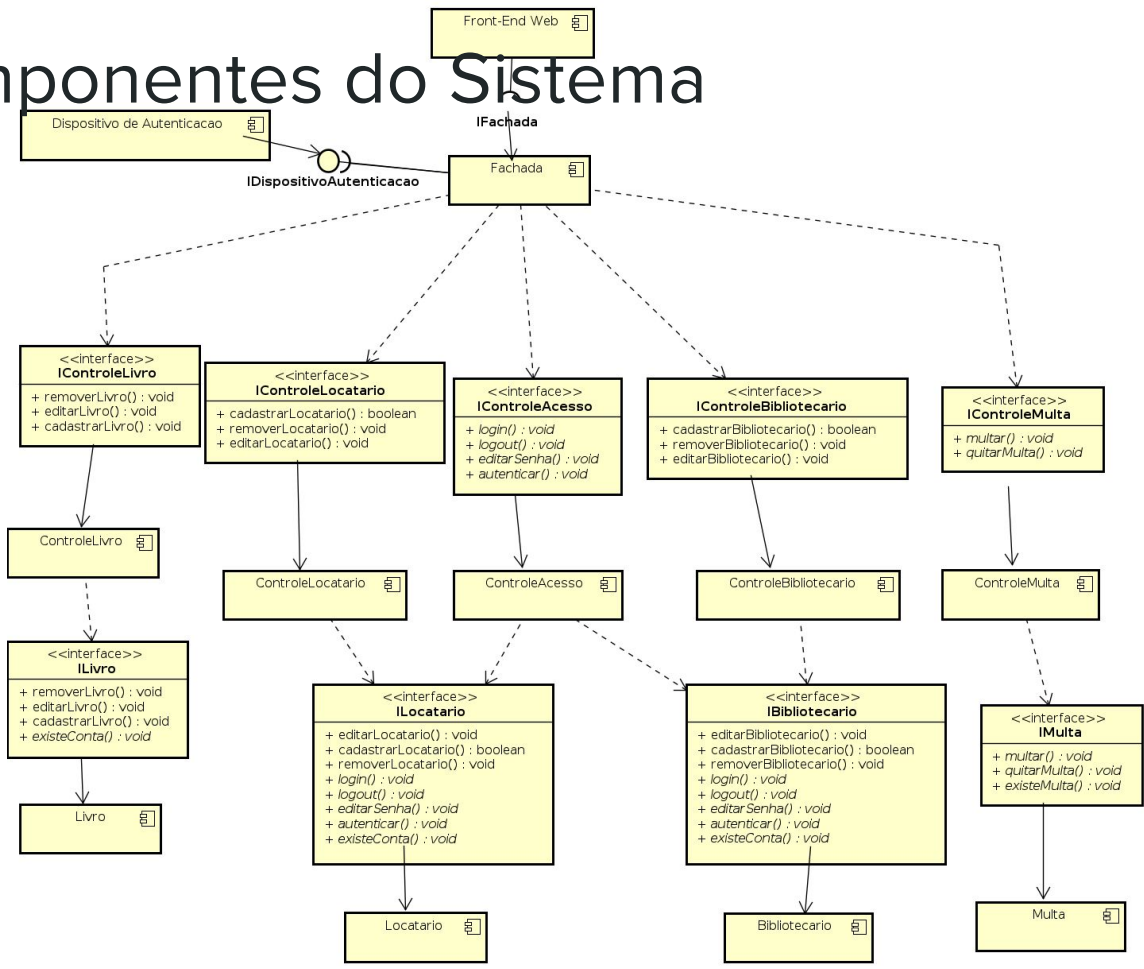
# Modelo de Informação Refinado



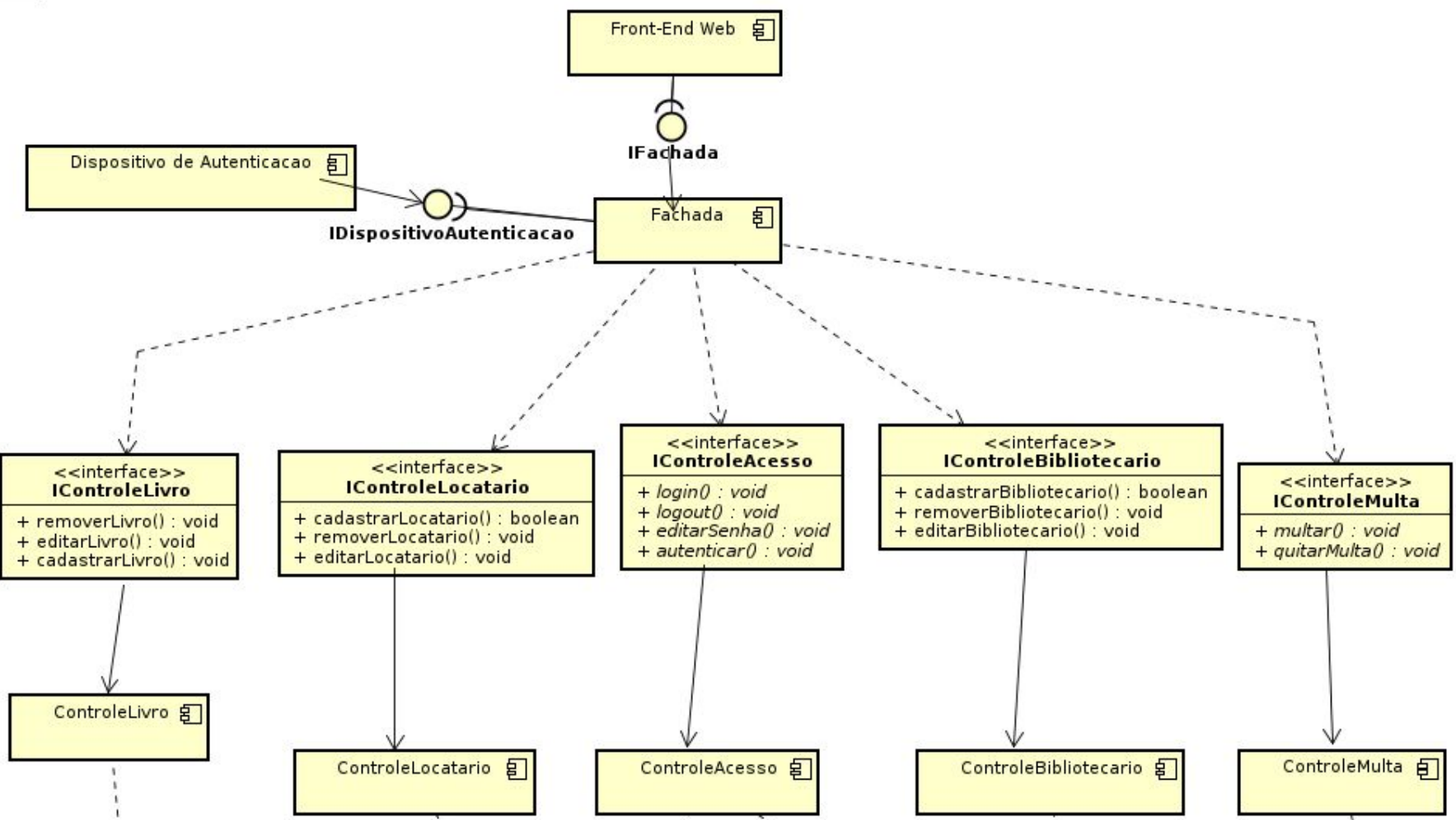
# Diagrama de Componentes dos Serviços (nível de análise)



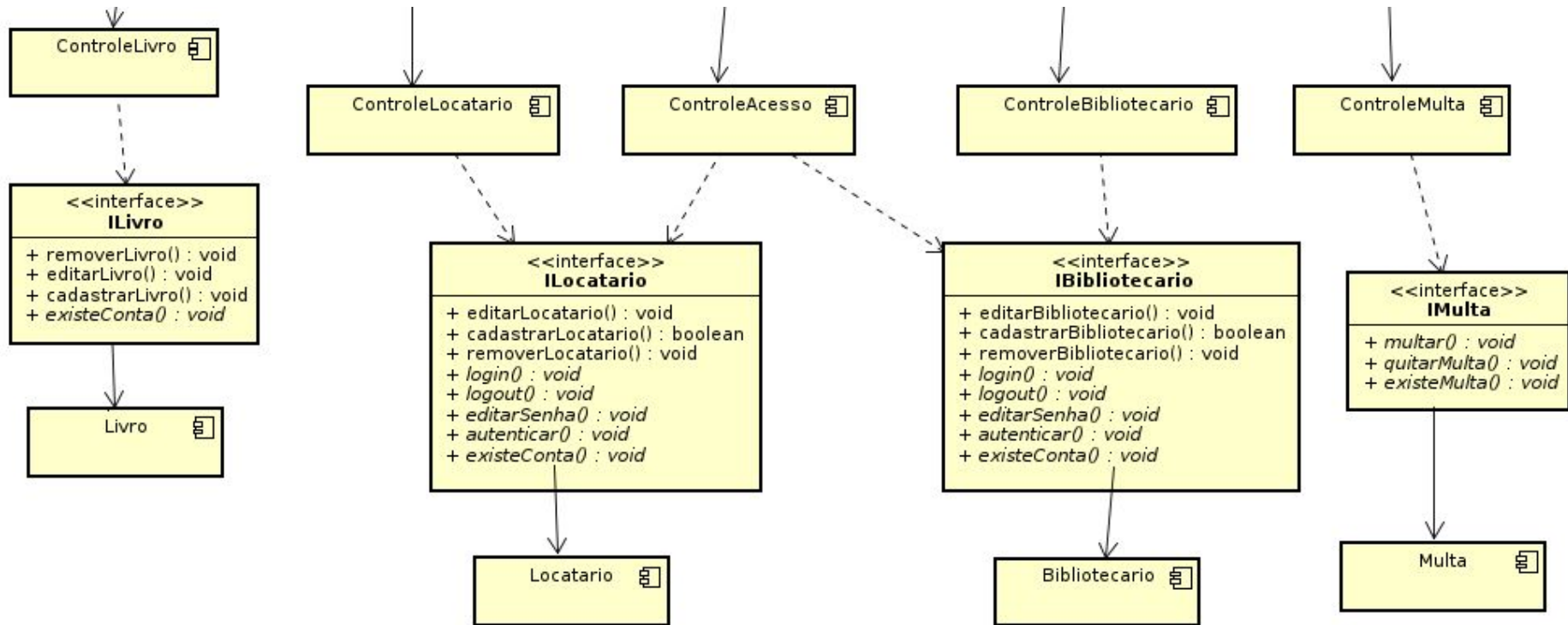
# Diagrama de Componentes do Sistema







# Diagrama de Componentes do Sistema



# Projeto de Front-end - Buscar Livro

**Buscar**

**Inserir livro**

**Remover livro**

**Locar livro**

**Buscar livro**

**Buscar livro**  
Busque pelo livro que deseja

Título

Editora

ISBN

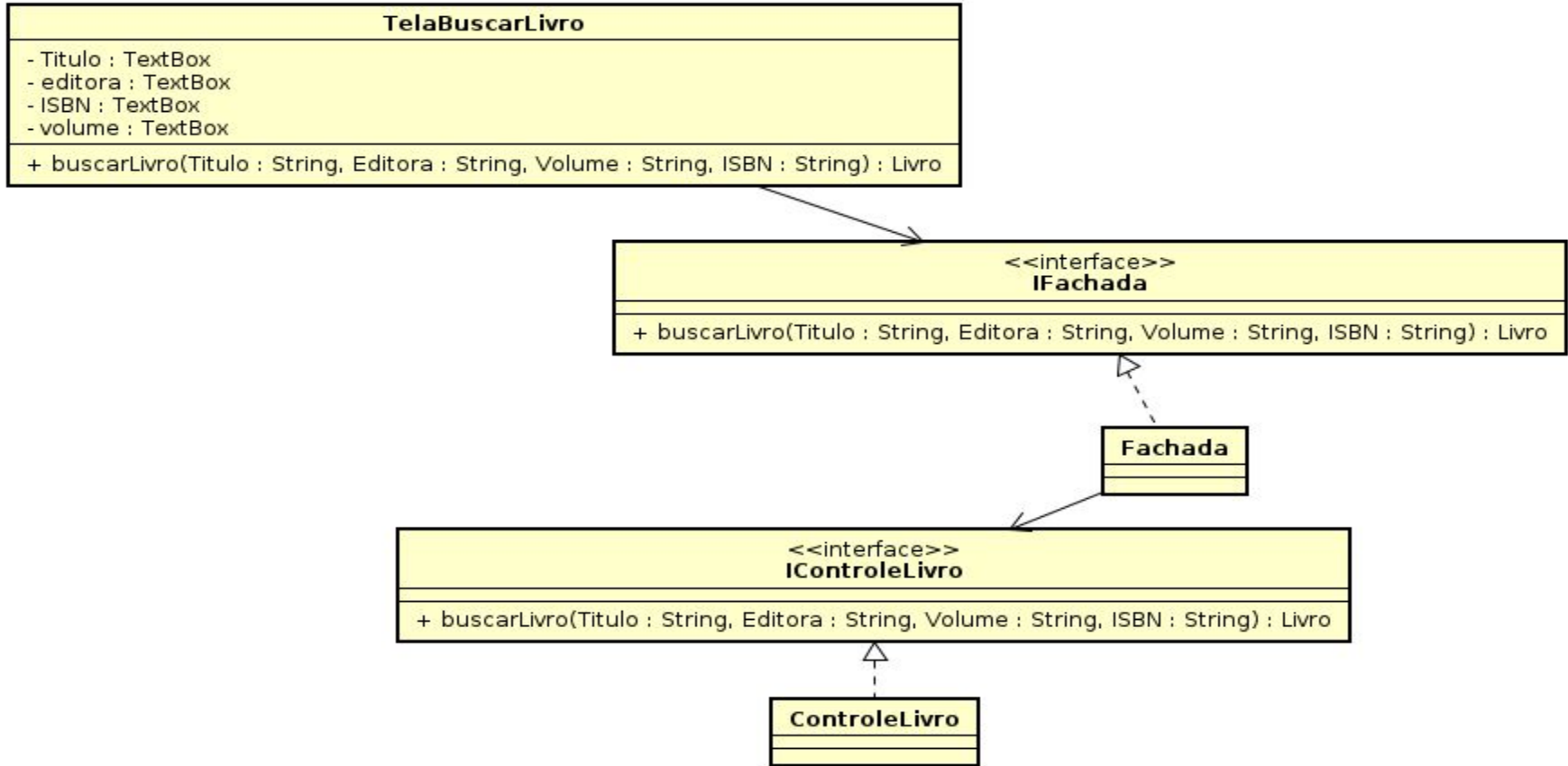
Volume

**TelaBuscarLivro**

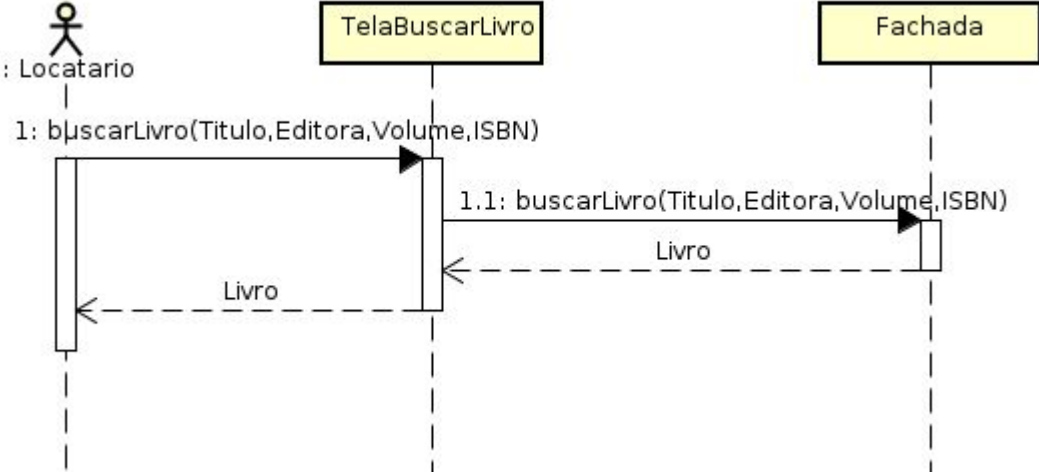
- Titulo : TextBox
- editora : TextBox
- ISBN : TextBox
- volume : TextBox

+ buscarLivro(Titulo : String, Editora : String, Volume : String, ISBN : String) : Livro

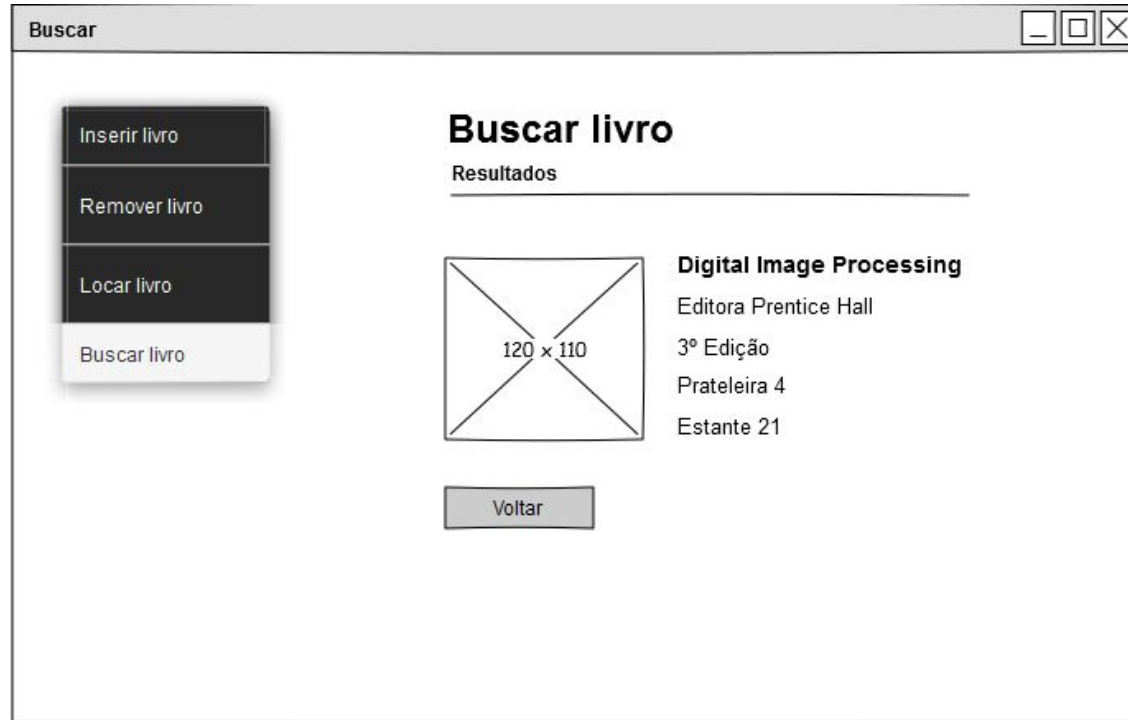
# Projeto de Front-end - Buscar Livro



# Projeto de Front-end - Buscar Livro



# Projeto de Front-end



# Projeto de Front-end - Cadastrar Bibliotecario

Buscar

## Cadastrar novo bibliotecário

Informe os dados do novo bibliotecário

Nome

CPF

Data de nascimento

Turno

Salário

Email

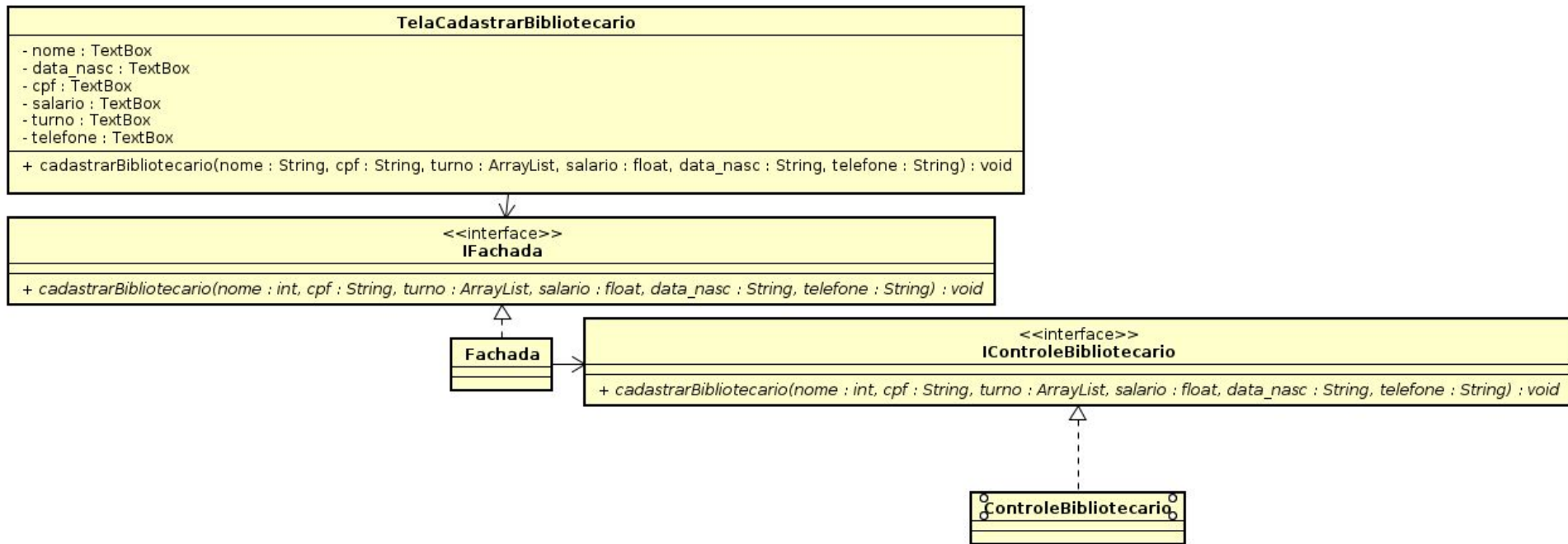
Telefone

**TelaCadastrarBibliotecario**

- nome : TextBox
- data\_nasc : TextBox
- cpf : TextBox
- salario : TextBox
- turno : TextBox
- telefone : TextBox

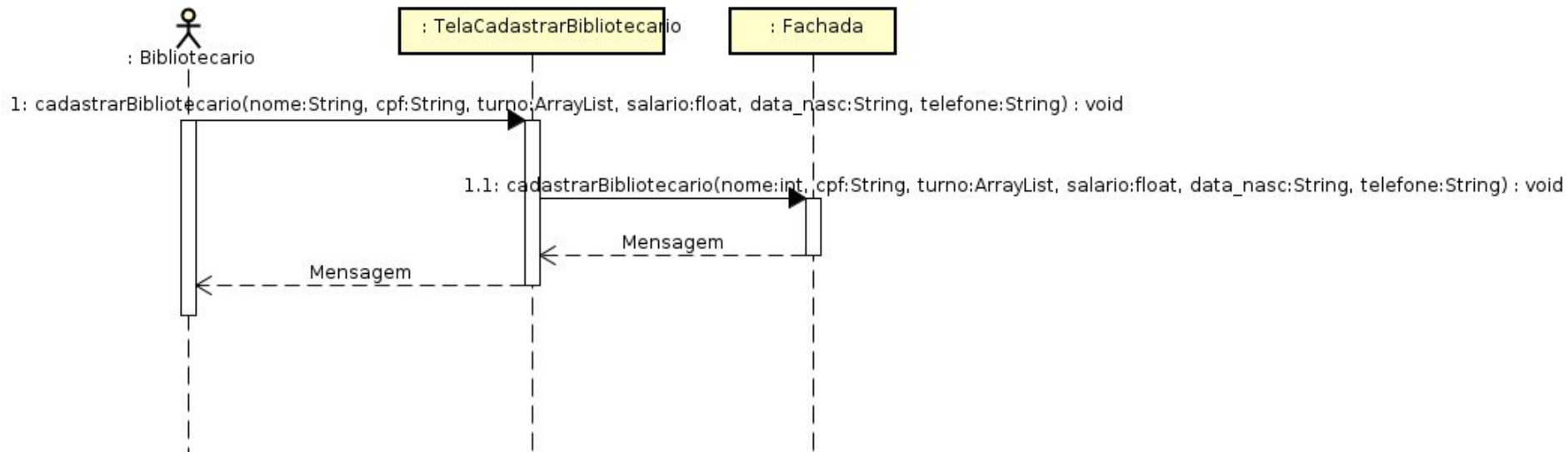
+ cadastrarBibliotecario(nome : String, cpf : String, turno : ArrayList, salario : float, data\_nasc : String, telefone : String) : void

# Projeto de Front-end - Cadastrar Bibliotecario





# Projeto de Front-end - Cadastrar Bibliotecario



# Projeto de Front-end

Login

## Login

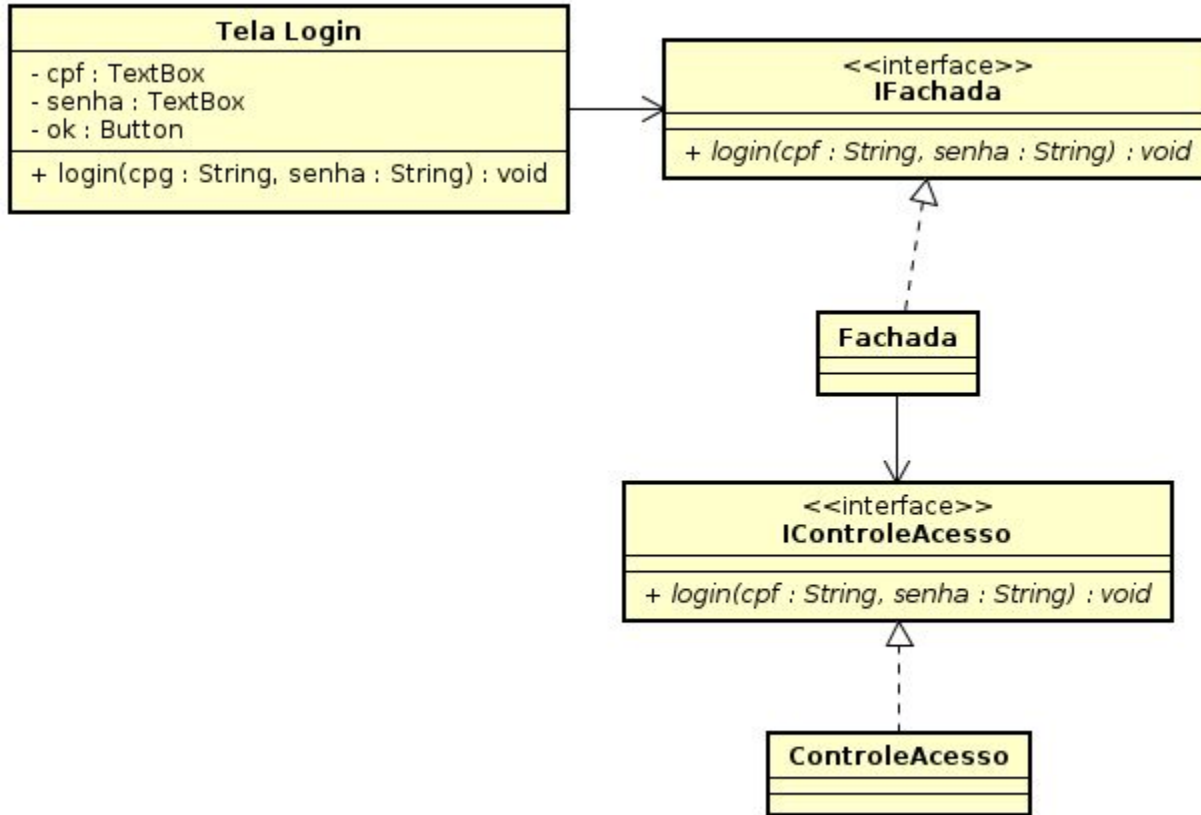
Informe suas credenciais

CPF

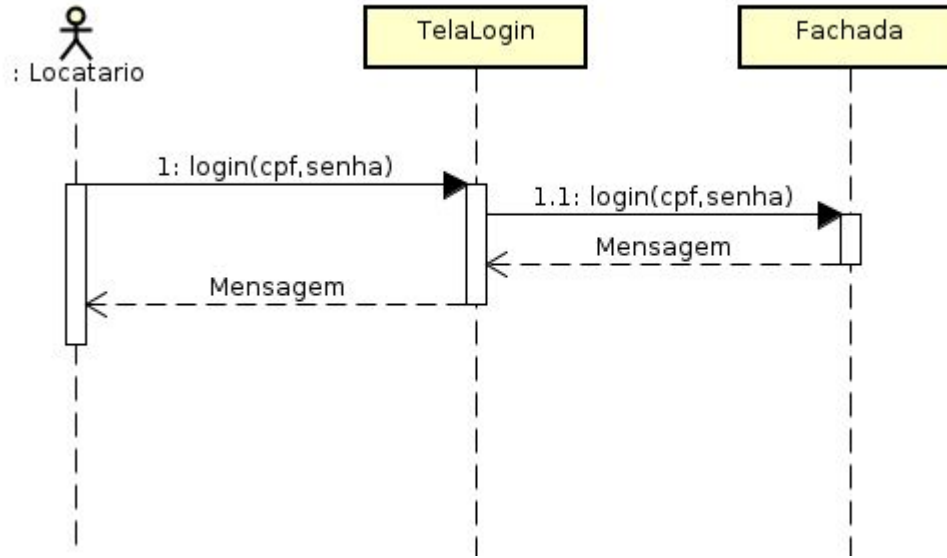
Senha

Tela Login
- cpf : TextBox - senha : TextBox - ok : Button
+ login(cpf : String, senha : String) : void

# Projeto de Front-end - Login

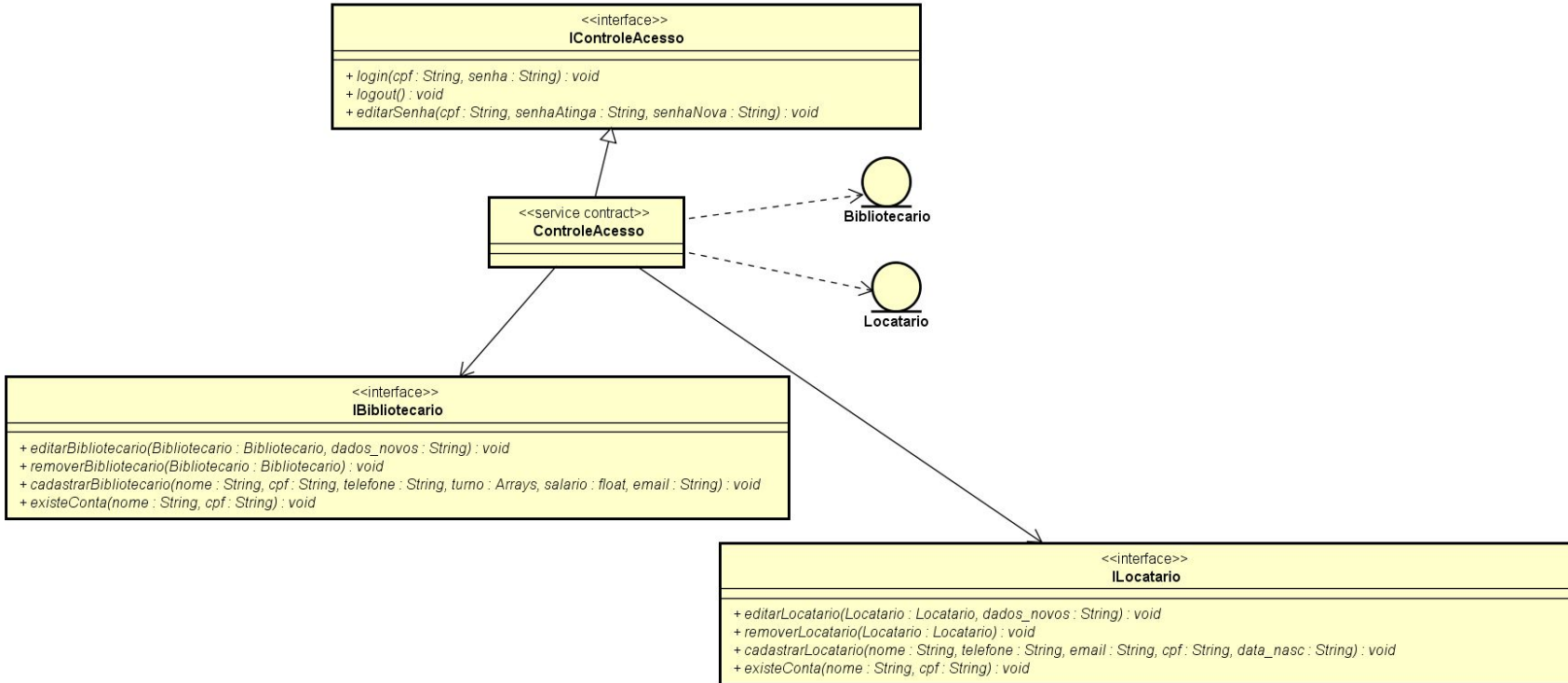


# Projeto de Front-end - Login

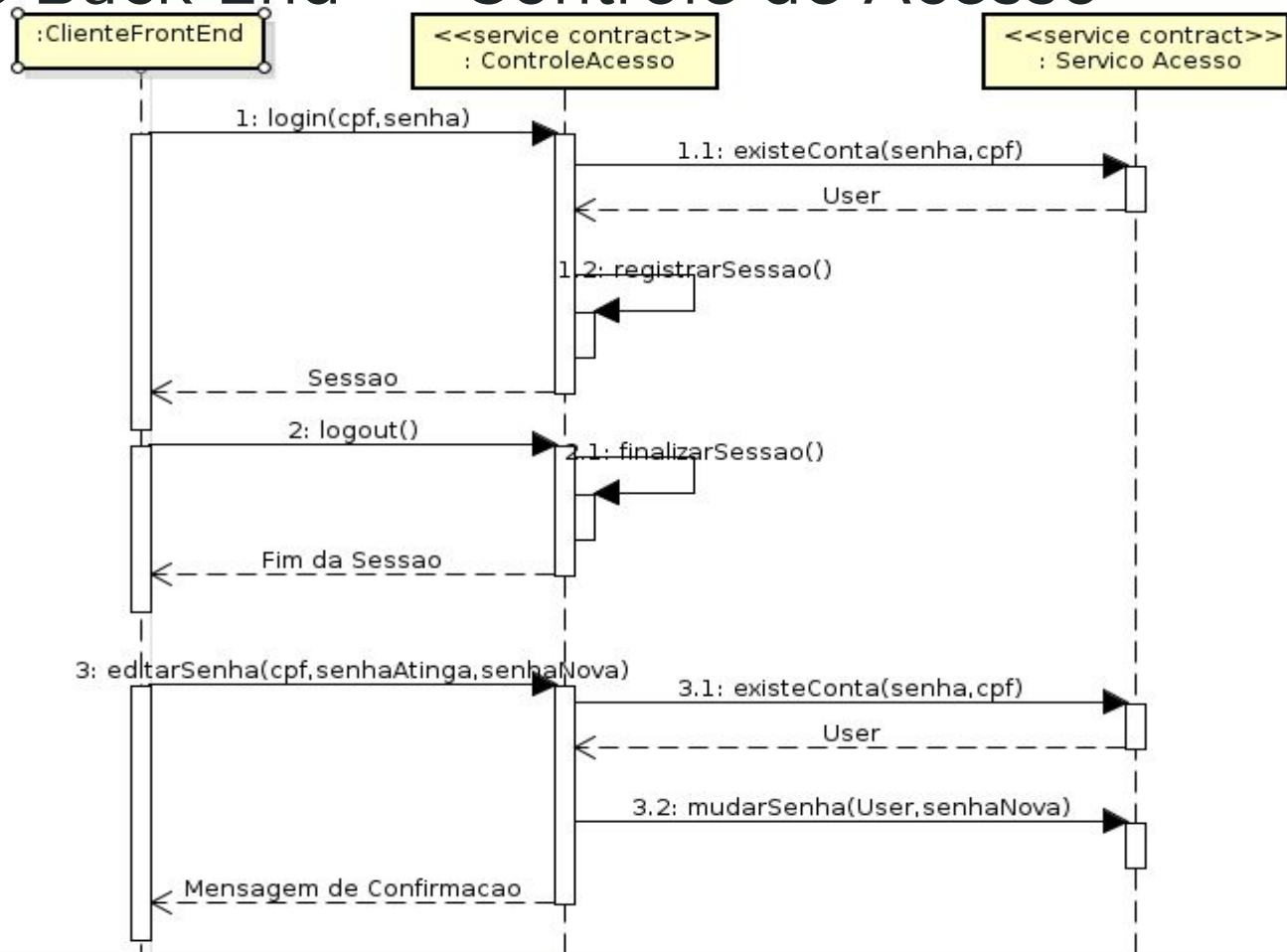


# Projeto Back-End - Controle de Acesso

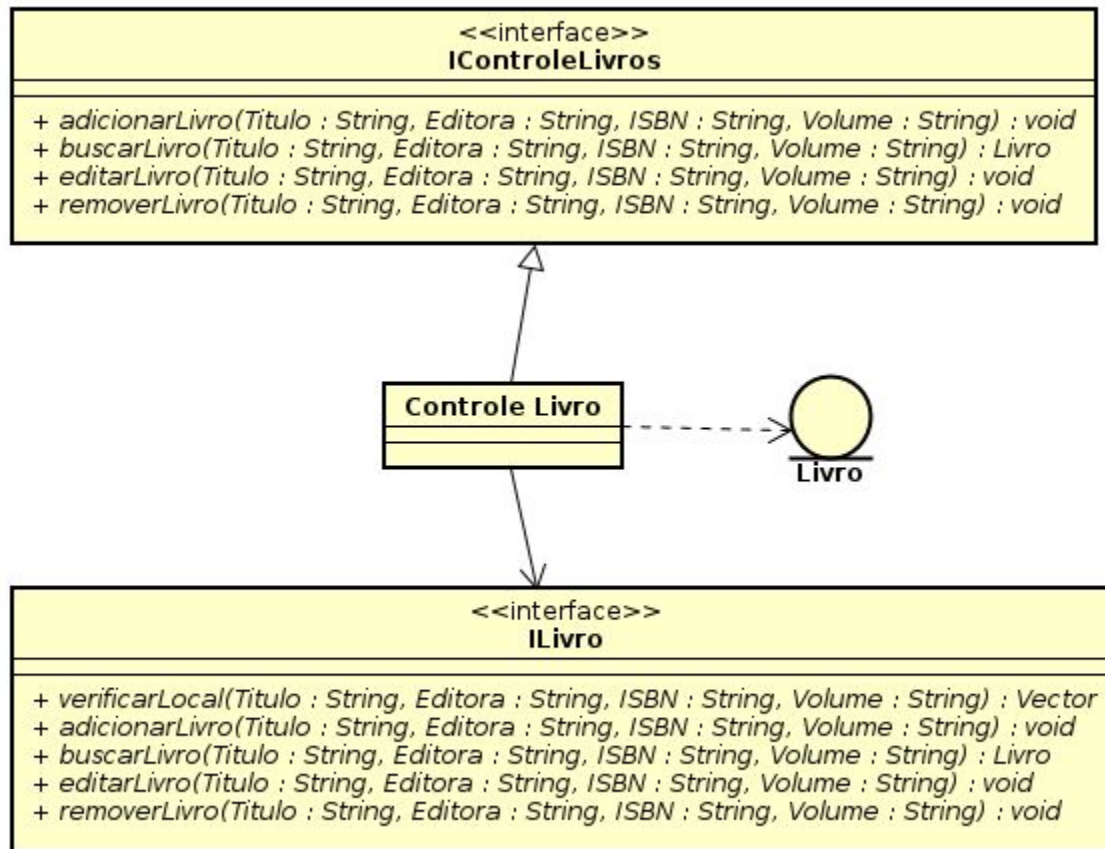
pkg



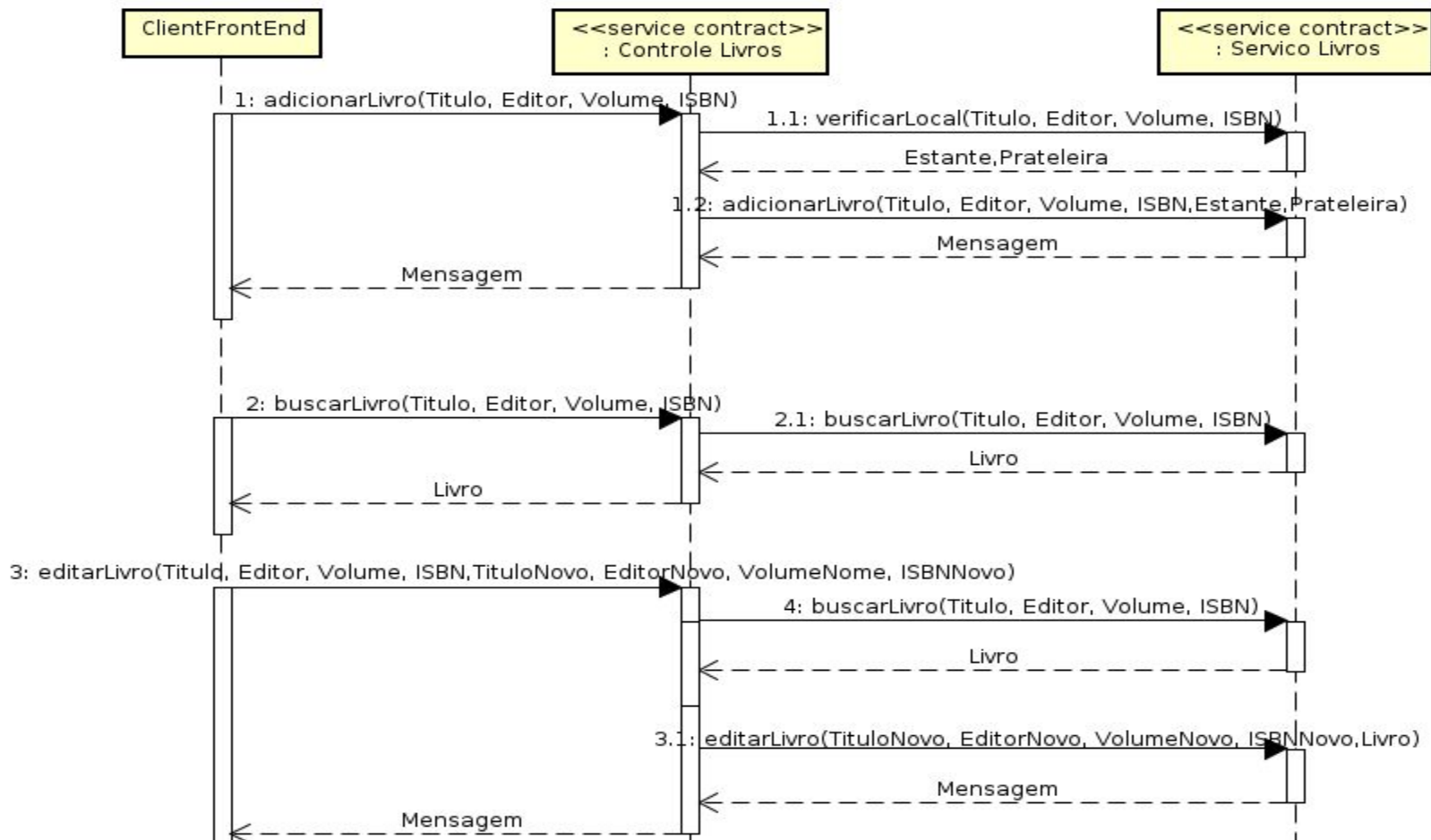
# Projeto Back-End - Controle de Acesso



# Projeto Back-End - Livro

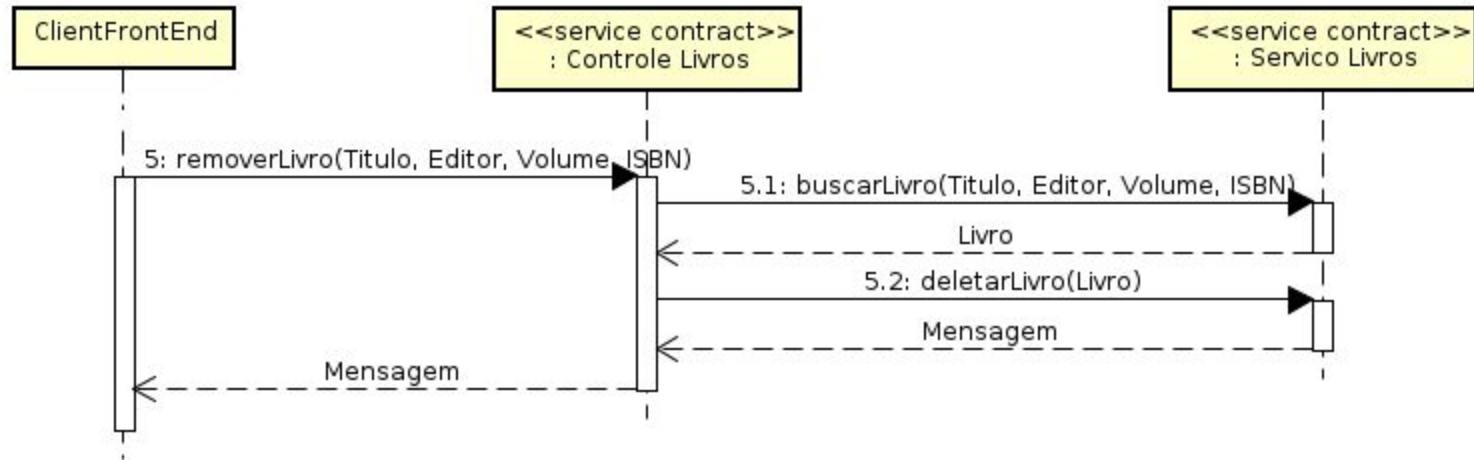


# Projeto Back-End - Livro

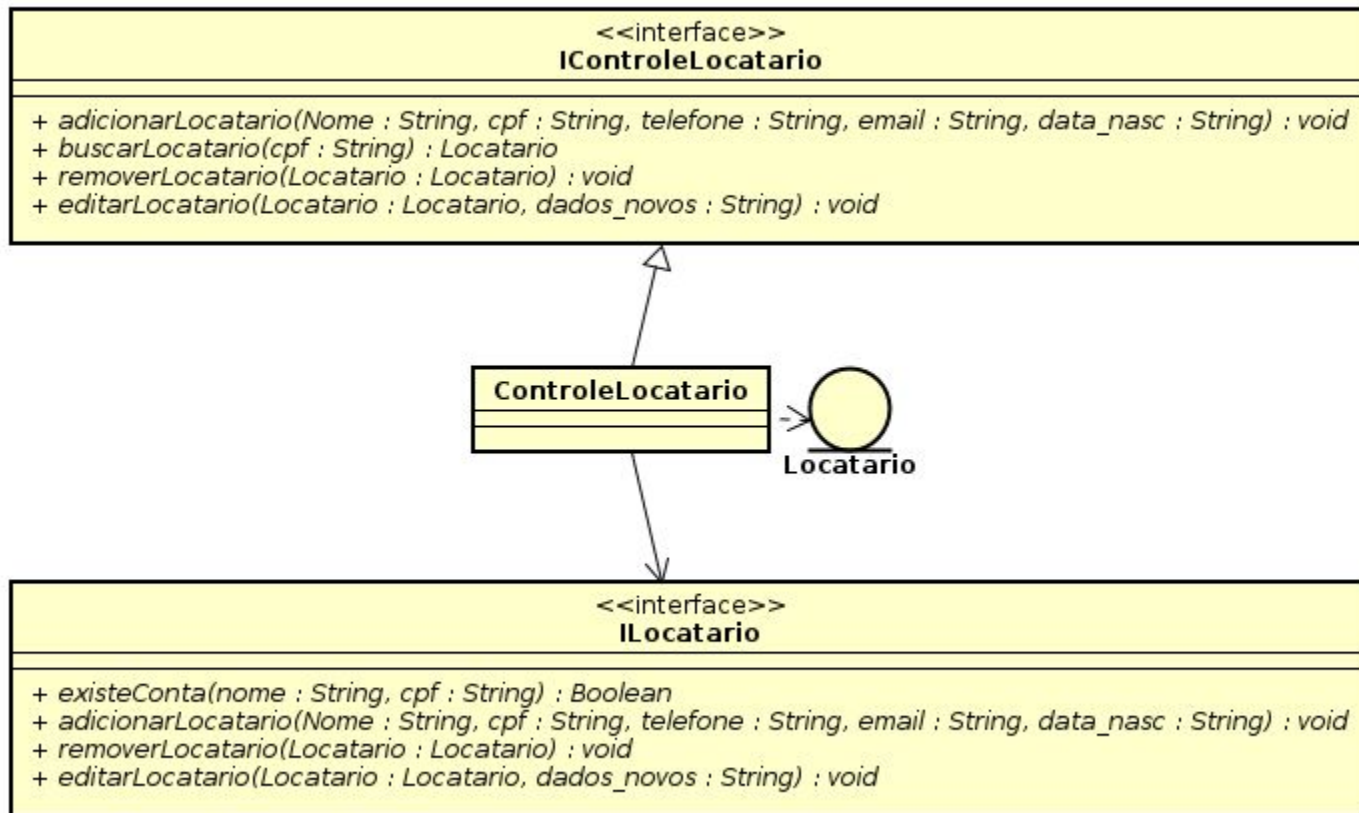




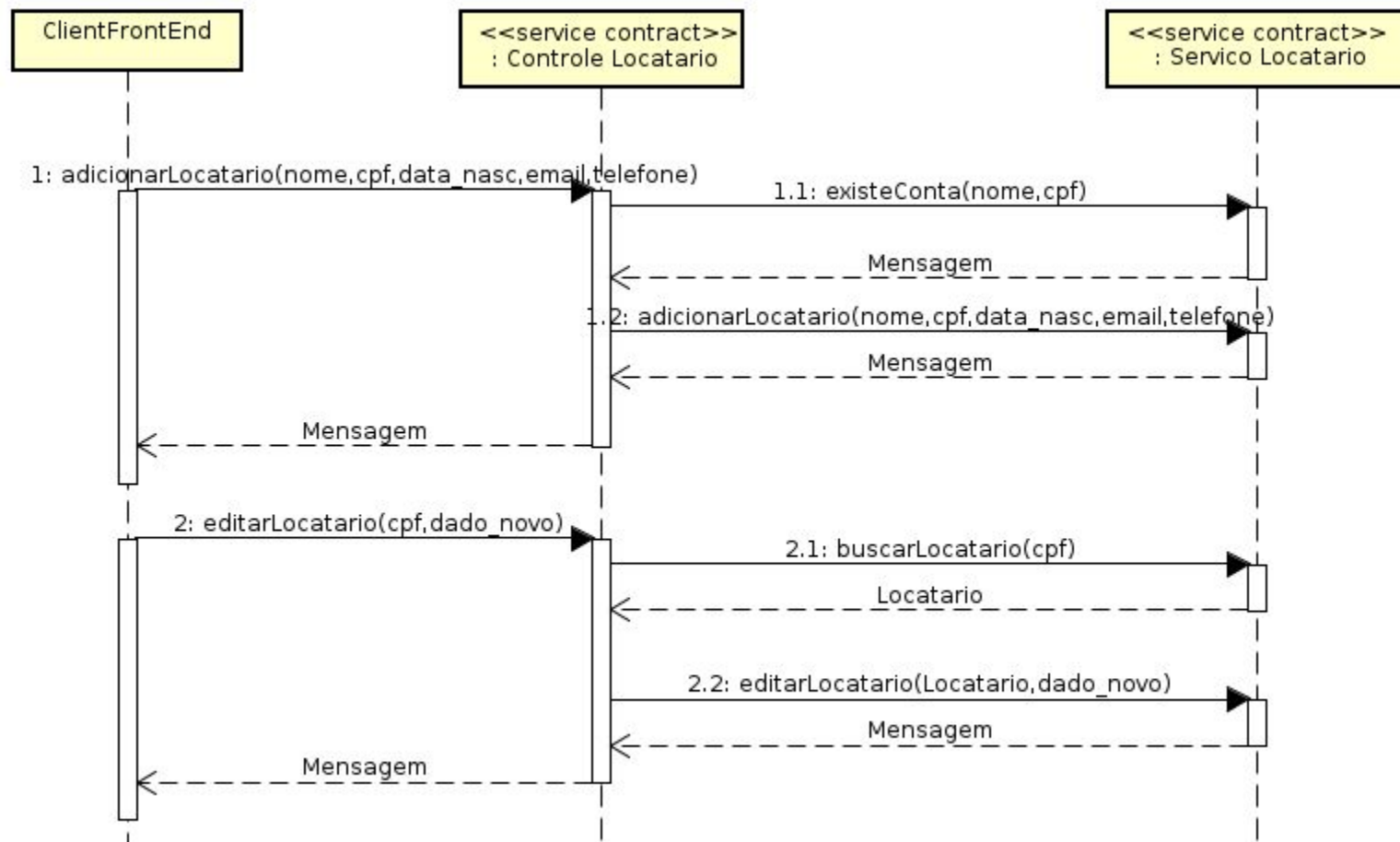
# Projeto Back-End - Livro



# Projeto Back-End - Locatario



# Projeto Back-End - Locatario



# Projeto Back-End - Locatario

