bookLoc

Grupo:

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

Roteiro

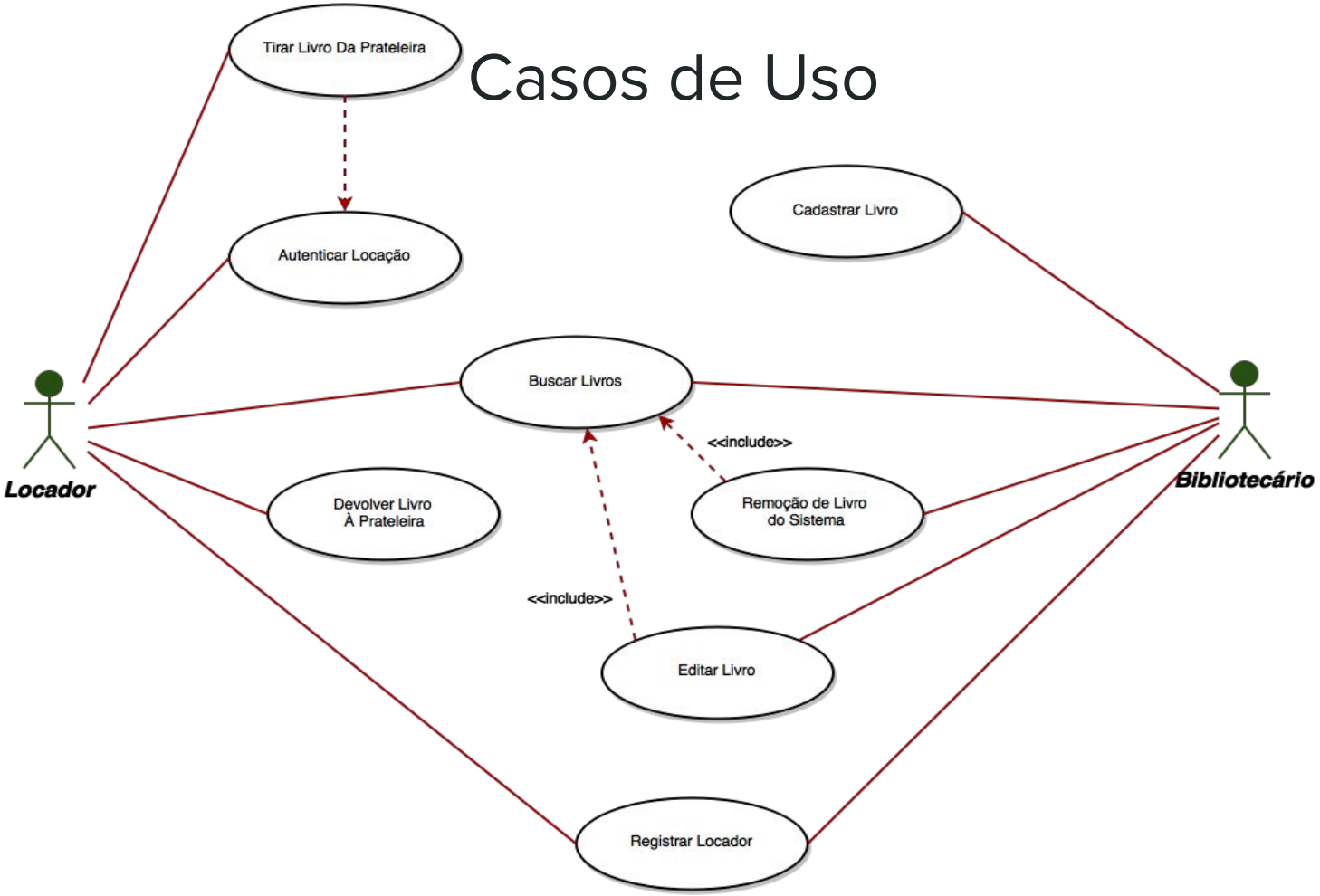
- Descrição do projeto
- Casos de Uso
- Arquitetura do sistema
- Implementação

Descrição do projeto

Trata-se de um sistema focado em usuários de serviços bibliotecários em geral, bem como usuários de sistemas de locação com caráter genérico sujeitos à uma validação em um ponto de atendimento central. É esperado com isso um aumento de qualidade no serviço prestado.

Casos de Uso

Casos de Uso


Tirar Livro Da Prateleira
Autenticar Locação
Cadastrar Livro
Buscar Livros
Locador
Bibliotecário
<<include>>
Devolver Livro
À Prateleira
Remoção de Livro
do Sistema
<<include>>
Editar Livro
Registrar Locador

UC01

[UC01]: Cadastrar Livro

Identificador: [UC 01]
Descrição: este caso de uso é responsável por cadastrar o livro no sistema da biblioteca.
Atores: bibliotecário
Prioridade: essencial
Pré-condições: verificar se a *tag* já foi cadastrada.
Pós-condições: o novo livro foi cadastrado no banco de dados do sistema.
Fluxo de Eventos Principal
1. O bibliotecário seleciona cadastrar livro no sistema; 2. O bibliotecário aplica a *tag* ao livro; 3. O bibliotecário entra com os dados da nova tag na interface de cadastro de livros; 4. O bibliotecário informa os dados: 5. Os dados serão cadastrado no banco de dados; 6. O sistema confirma o cadastro do novo livro. 7. O bibliotecário posiciona o livro na prateleira; 8. O sistema identifica a *tag* do livro e o estado do livro como “livro registrado”; 9. O livro é habilitado para locação.

Diagrama de Classe (Cadastrar Livro)

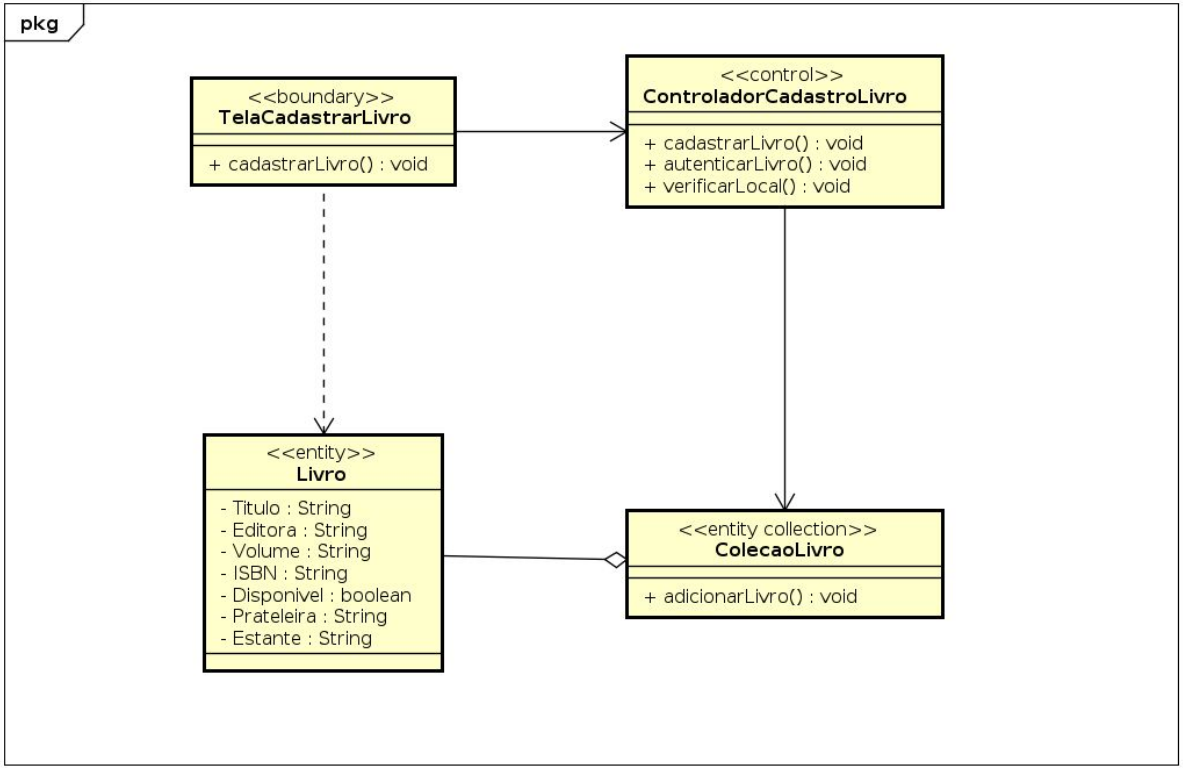
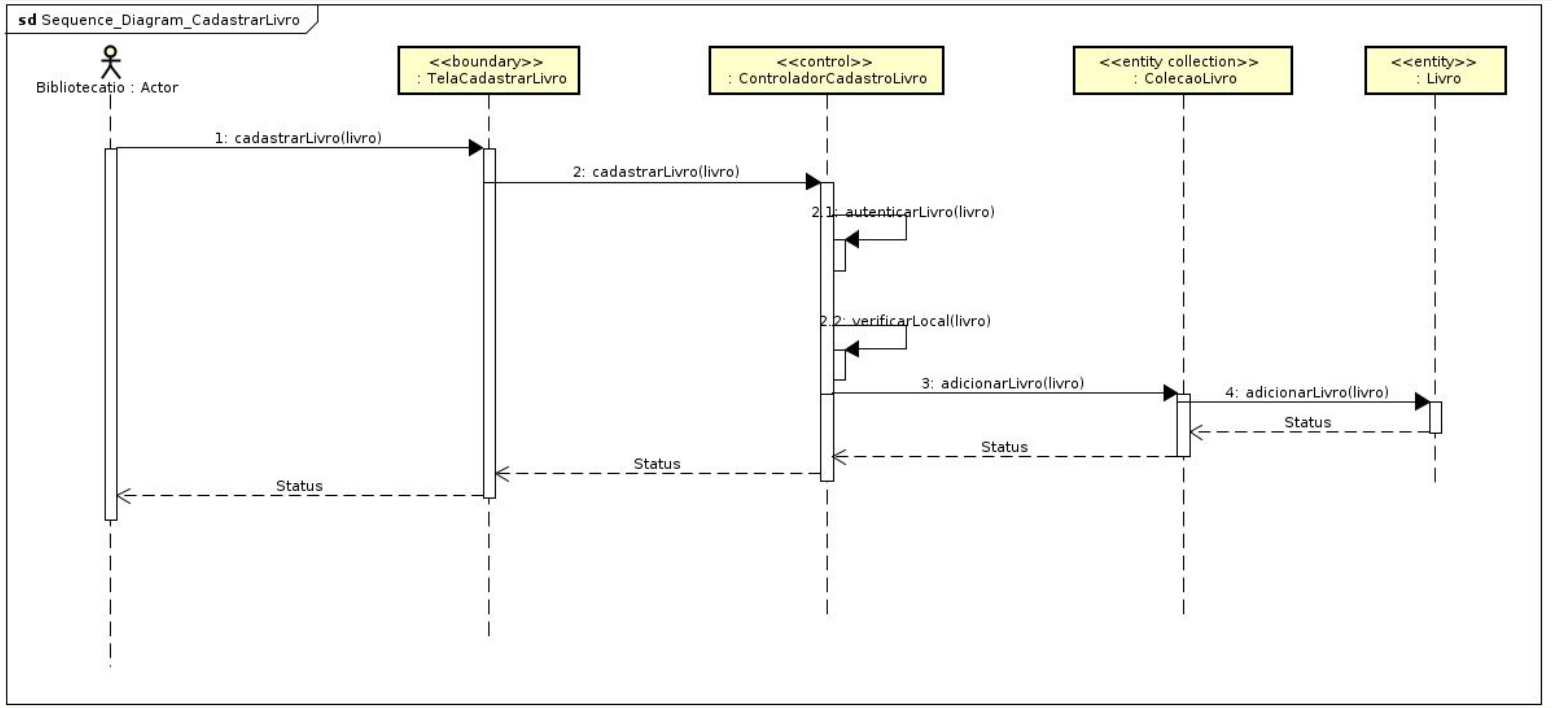


Diagrama de Sequência (Cadastrar Livro)



UC02

[UC 02]: Buscar Livro

Identificador: [UC 02]
Descrição: este caso de uso é responsável por buscar livros por características presentes no sistema da biblioteca.
Atores: Locador, Bibliotecário
Prioridade: importante
Pré-condições: usuário deve estar logado no sistema.
Pós-condições: uma seção com detalhes sobre o livro é exibida para usuário.
Fluxo de Eventos Principal
1. O usuário seleciona a opção de buscar por um livro específico no sistema; 2. O usuário insere os dados do livro à buscar; 3. O sistema exibe os detalhes do livro solicitado para o usuário.

Diagrama de Classe (Buscar Livro)

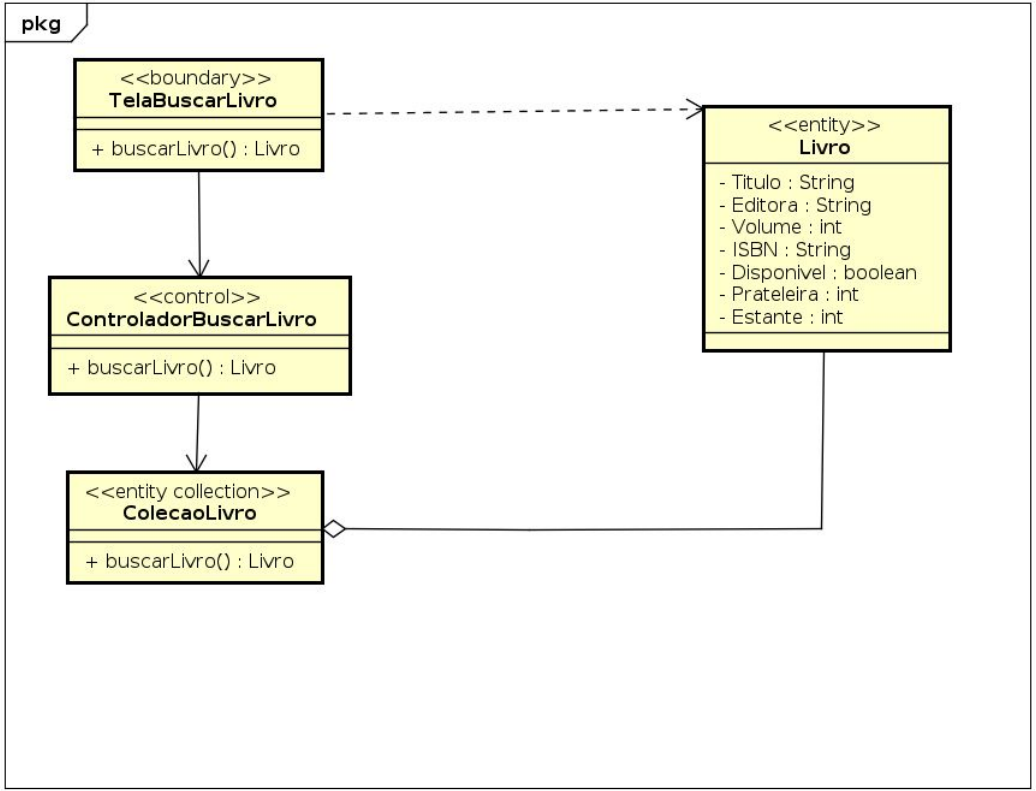
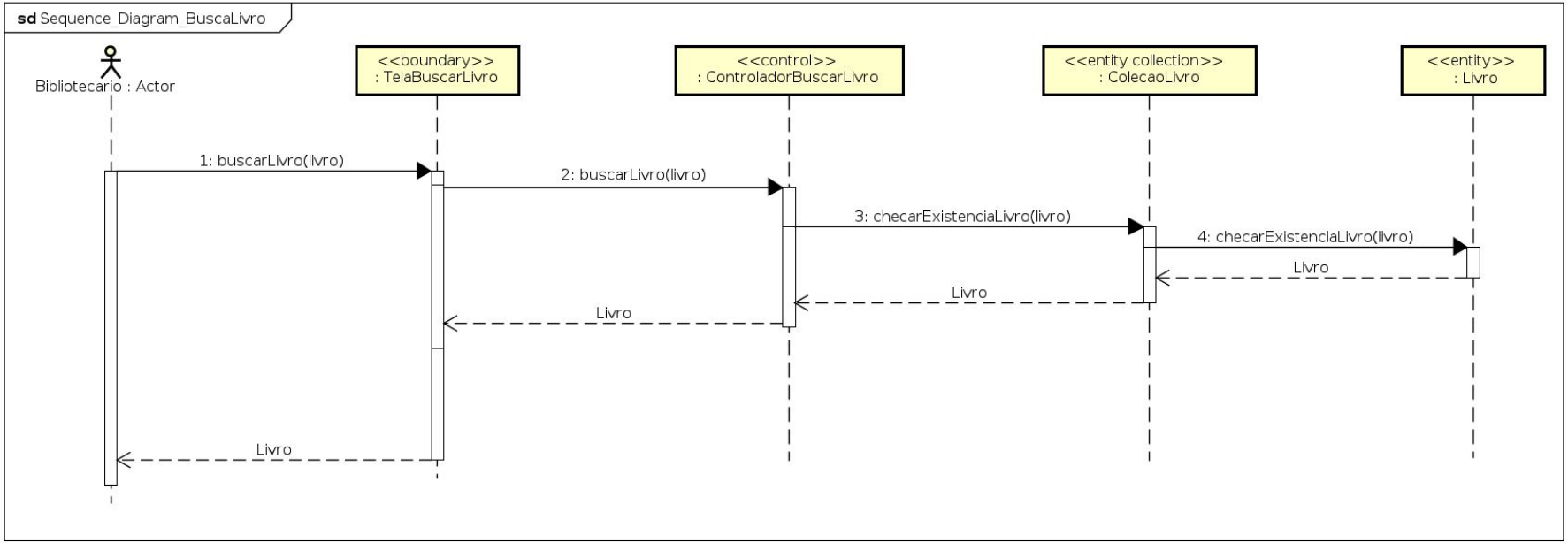


Diagrama de Sequência (Buscar Livro)



UC03

[UC 03]: Locar Livro

Identificador: [UC 03]
Descrição: este caso de uso é responsável pelo processo de retirada de um livro de uma prateleira e sua autenticação por parte do locador.
Atores: Locador
Prioridade: essencial
Pré-condições: o livro deve constar como “disponivel” no sistema.
Pós-condições: o livro deve constar como “locado” no sistema.
Fluxo de Eventos Principal
1. O locador retira o livro que deseja locar da prateleira. 2. Uma entrada referente ao livro retirado aparece no dispositivo de autenticação da prateleira. 3. O locador seleciona a entrada do seu livro e digita suas credenciais. 4. Caso a autenticação seja verdadeira, o livro é alocado para a conta do locador.

Diagrama de Classe (Locar Livro)

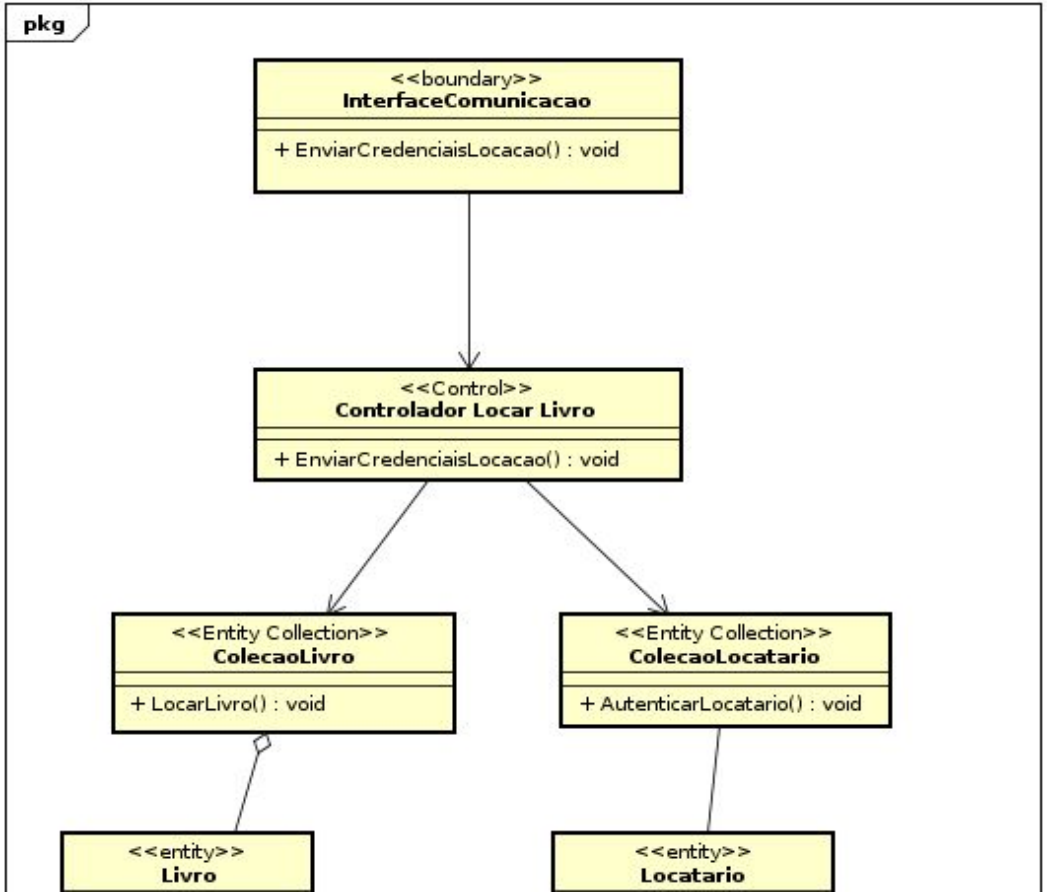


Diagrama de Classe (Locar Livro)

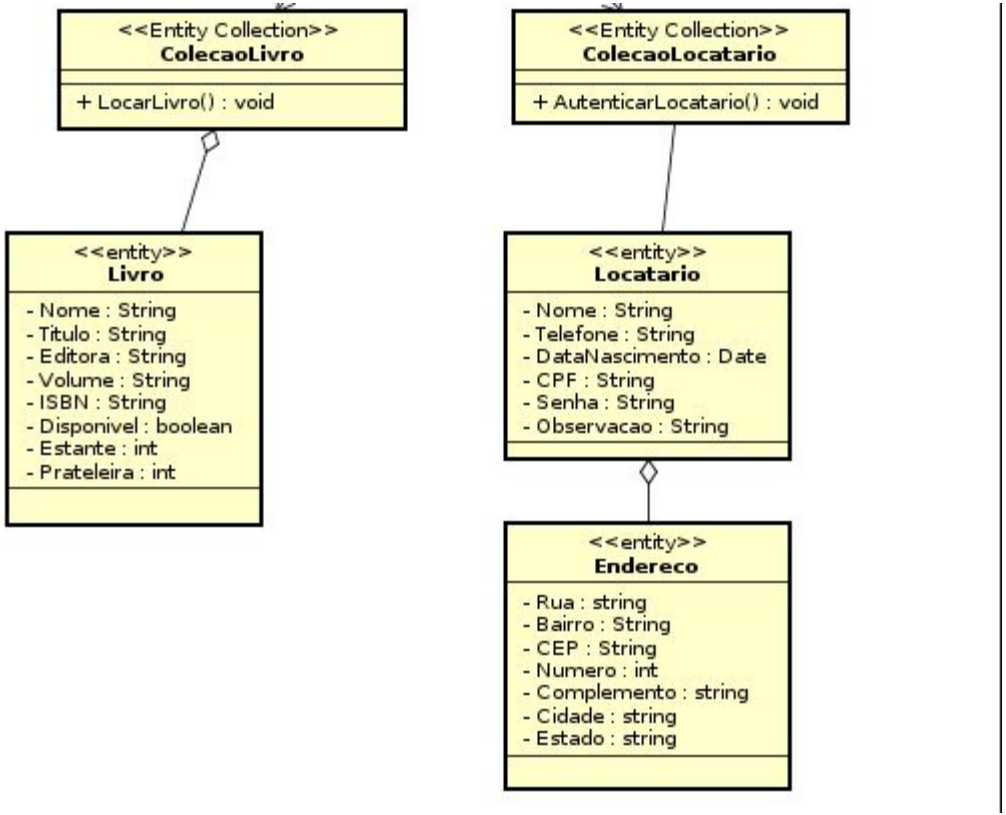


Diagrama de Sequência (Locar Livro)

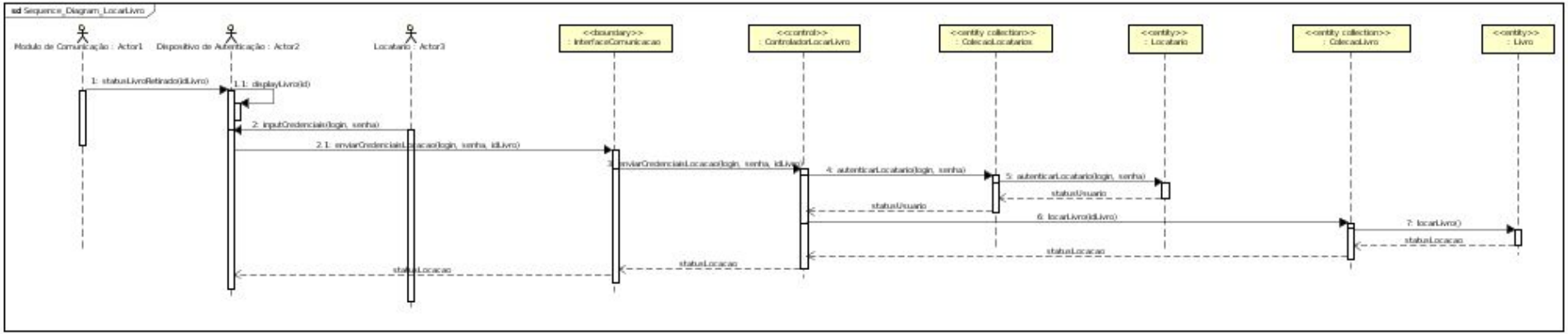


Diagrama de Sequência (Locar Livro)

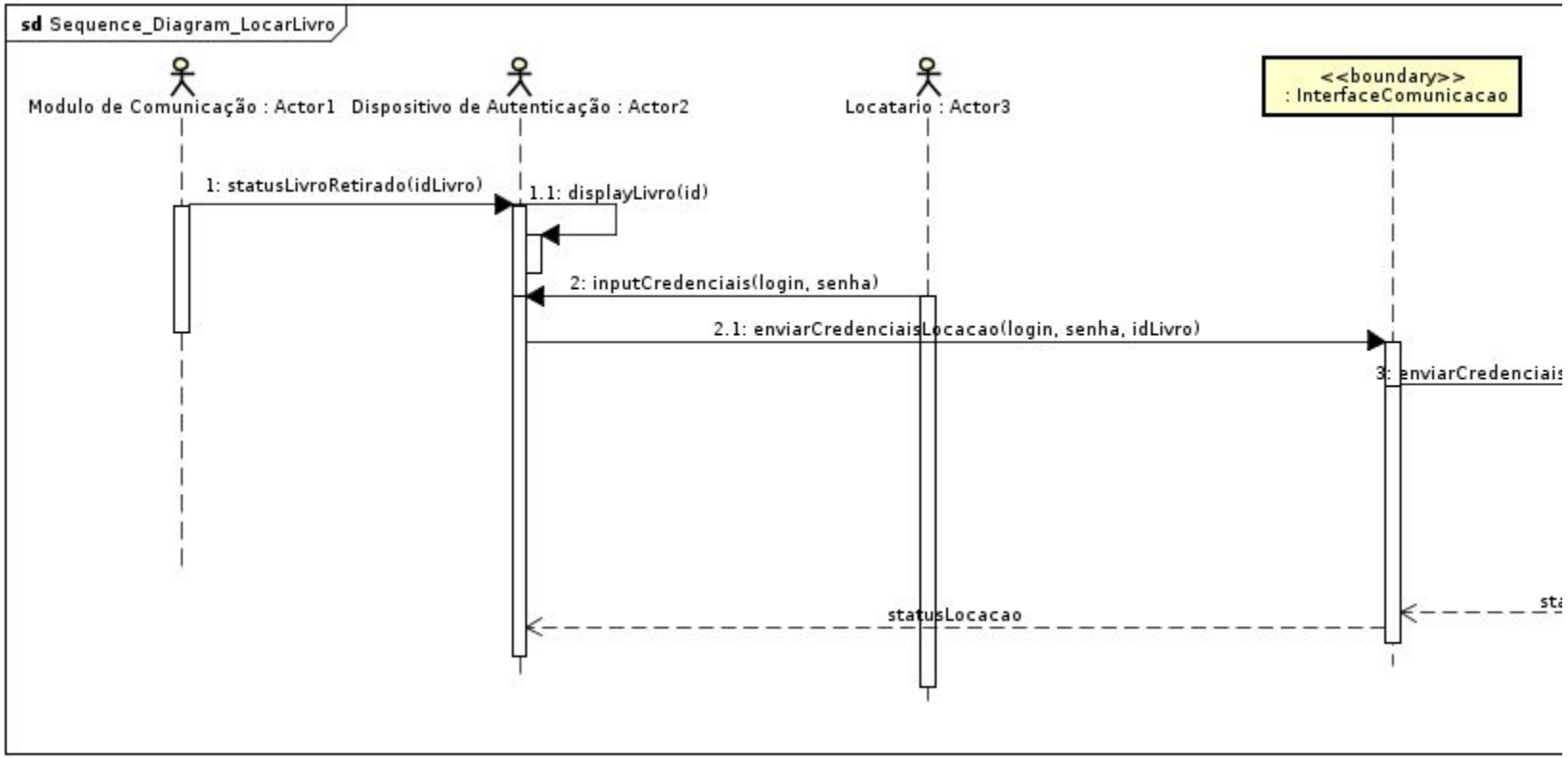


Diagrama de Sequência (Locar Livro)

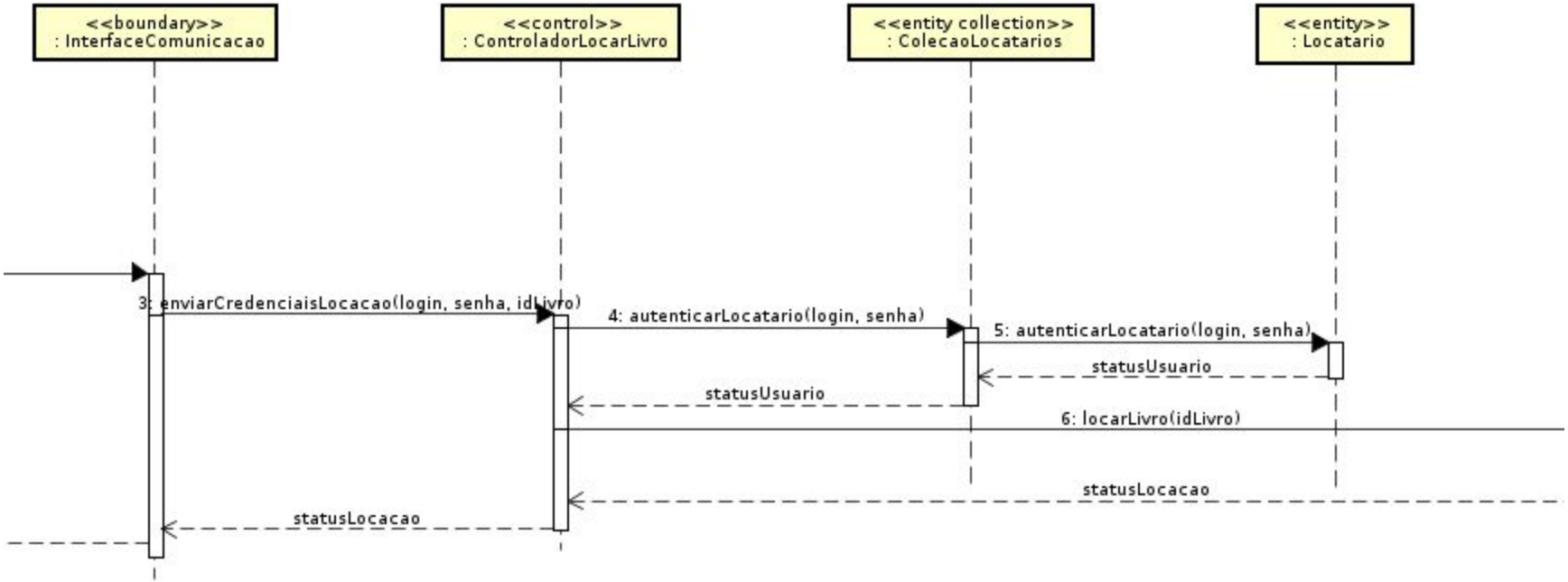
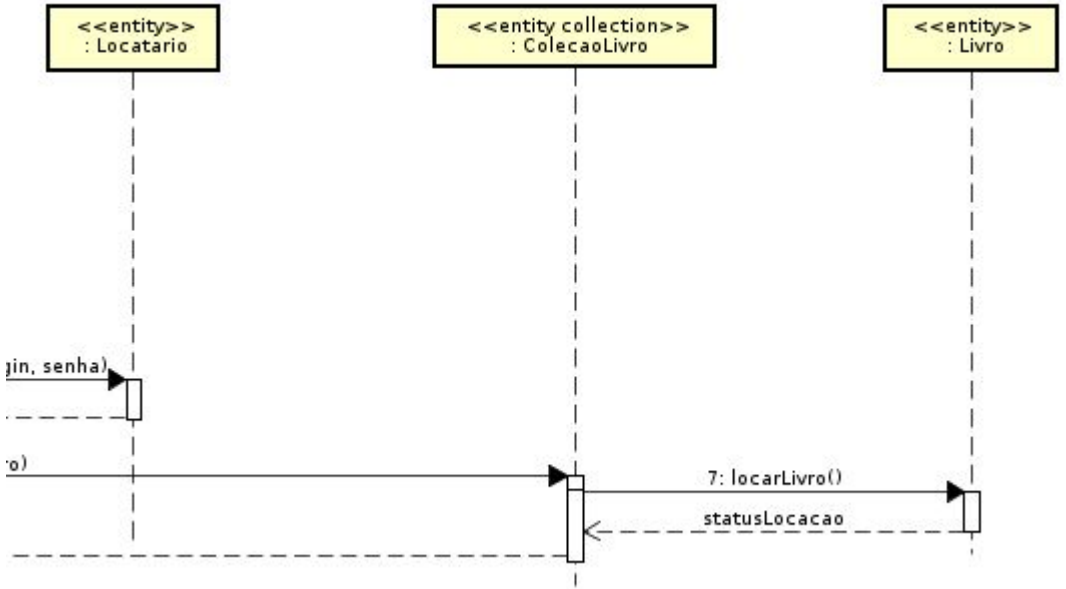


Diagrama de Sequência (Locar Livro)



UC04

[UC 04]: Editar Livro

Identificador: [UC 04]
Descrição: este caso de uso é responsável por editar as informações de um livro específico no sistema da biblioteca.
Atores: bibliotecário
Prioridade: importante
Pré-condições: o livro deve existir no sistema.
Pós-condições: o livro terá suas informações atualizadas no banco de dados.
Fluxo de Eventos Principal
1. O biliotecário busca um livro no sistema. 2. O bibliotecário seleciona a opção de editar livro; 3. O bibliotecário digita os dados do livro que deseja modificar na interface de edição; 4. O bibliotecário pode salvar ou cancelar as mudanças; 5. O sistema salva a alteração no banco de dados.

Diagrama de Classe (Editar Livro)

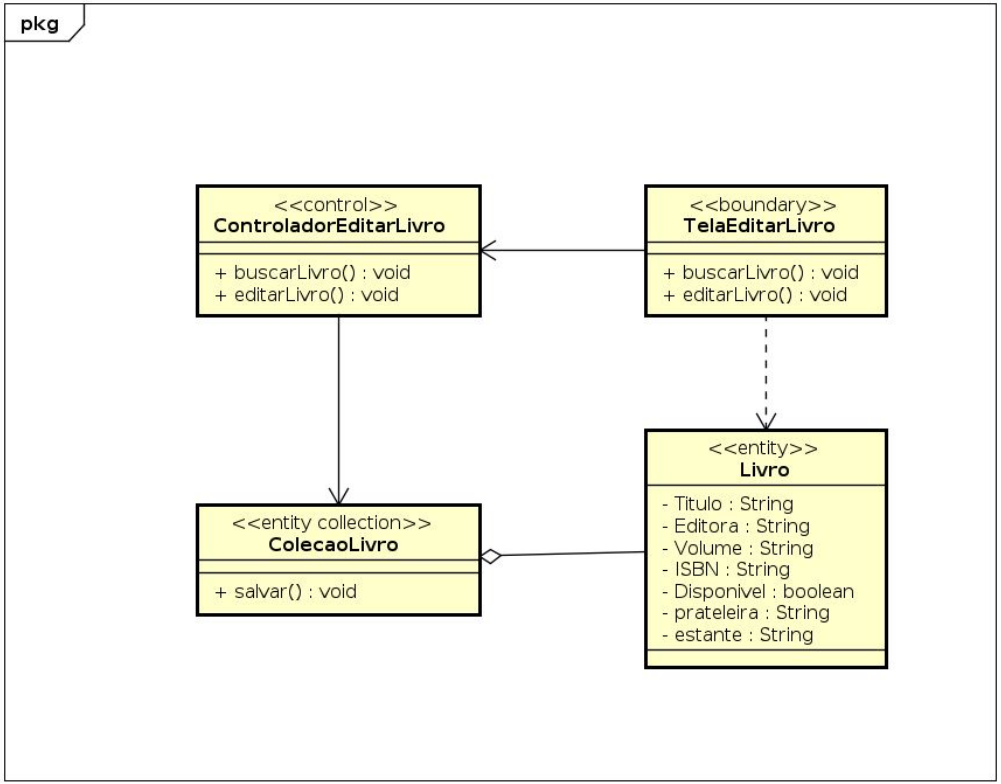
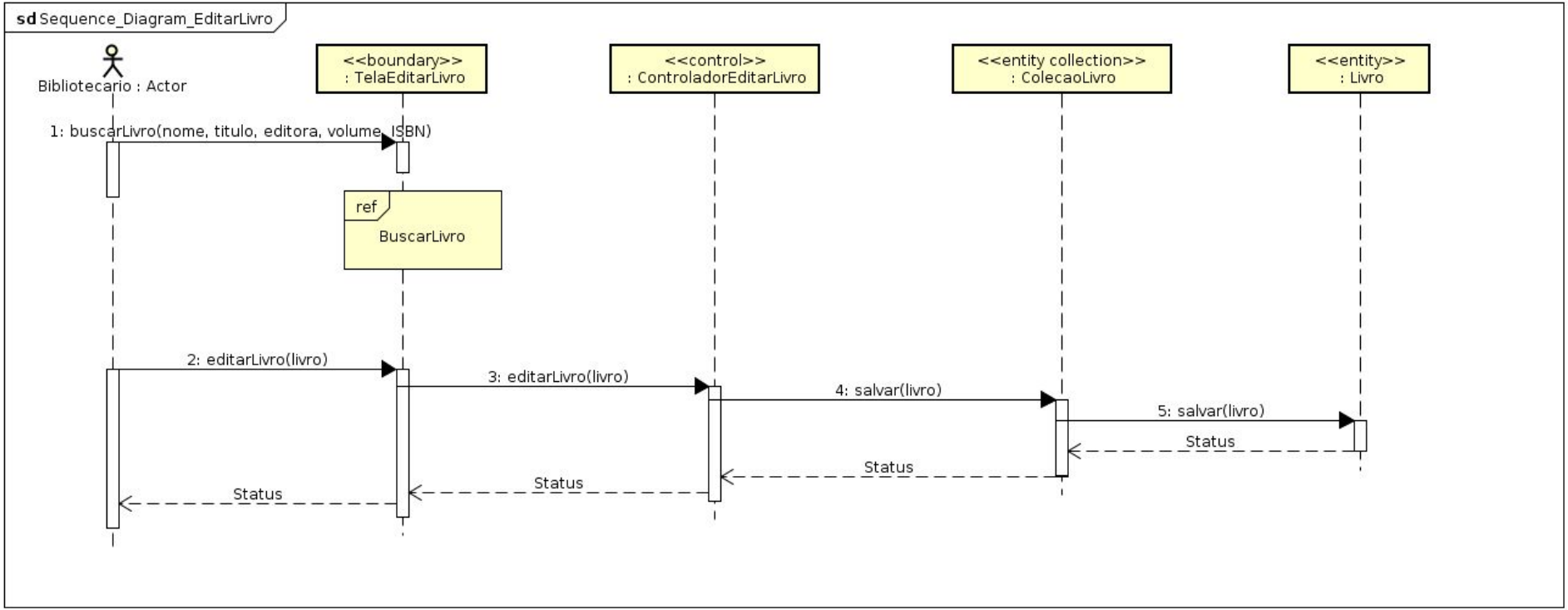


Diagrama de Sequência (Editar Livro)



UC05

[UC05]: Remoção de Livro do Sistema

Identificador: [UC 05]
Descrição: este caso de uso é responsável por remover o livro do sistema da biblioteca.
Atores: Bibliotecário
Prioridade: importante
Pré-condições: o livro deve existir no sistema.
Pós-condições: o livro terá suas informações removidas do banco de dados.
Fluxo de Eventos Principal
1. O bibliotecário seleciona a opção de remover livro; 2. O sistema abre a opção de buscar por um livro específico no sistema; 3. O bibliotecário digita os dados do livro; 4. O sistema recupera o livro; 6. O bibliotecário seleciona para remover o livro do sistema; 7. O sistema remove o livro do banco de dados.

Diagrama de Classe (Remover Livro)

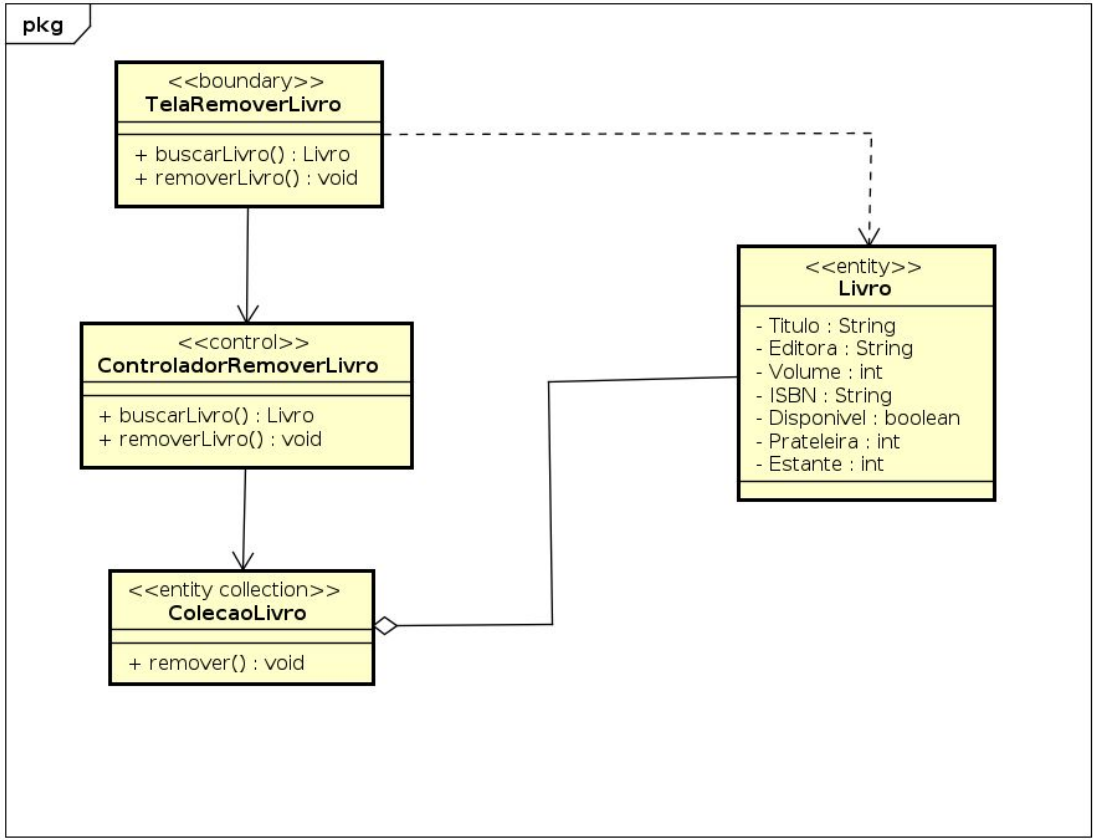
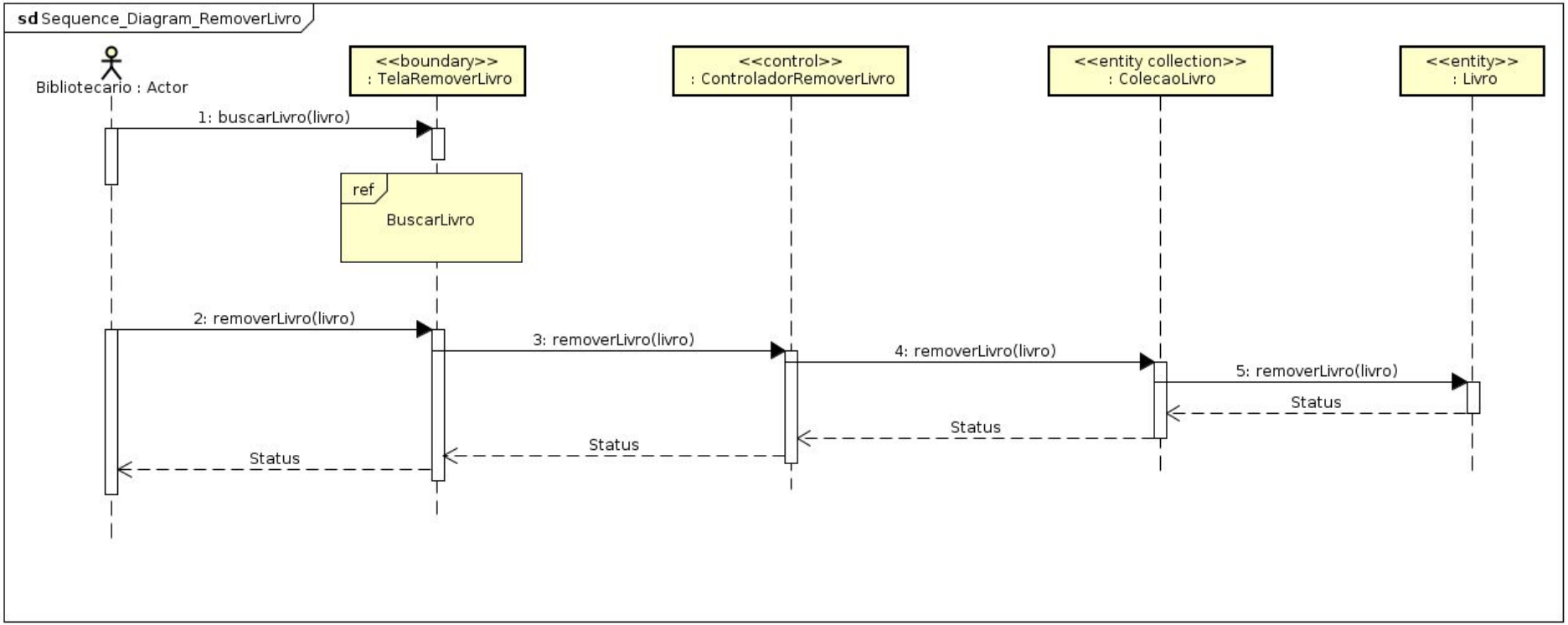
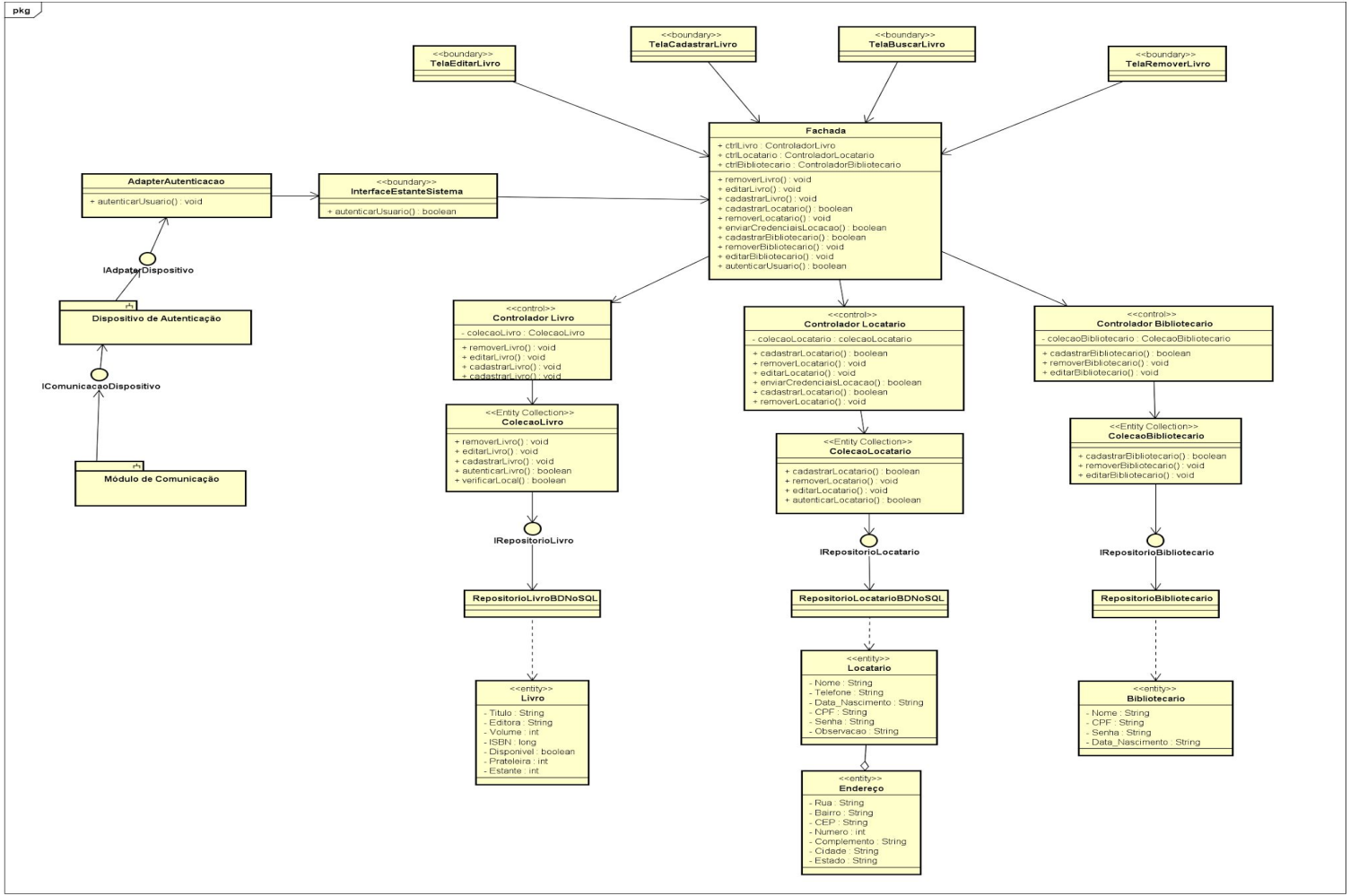
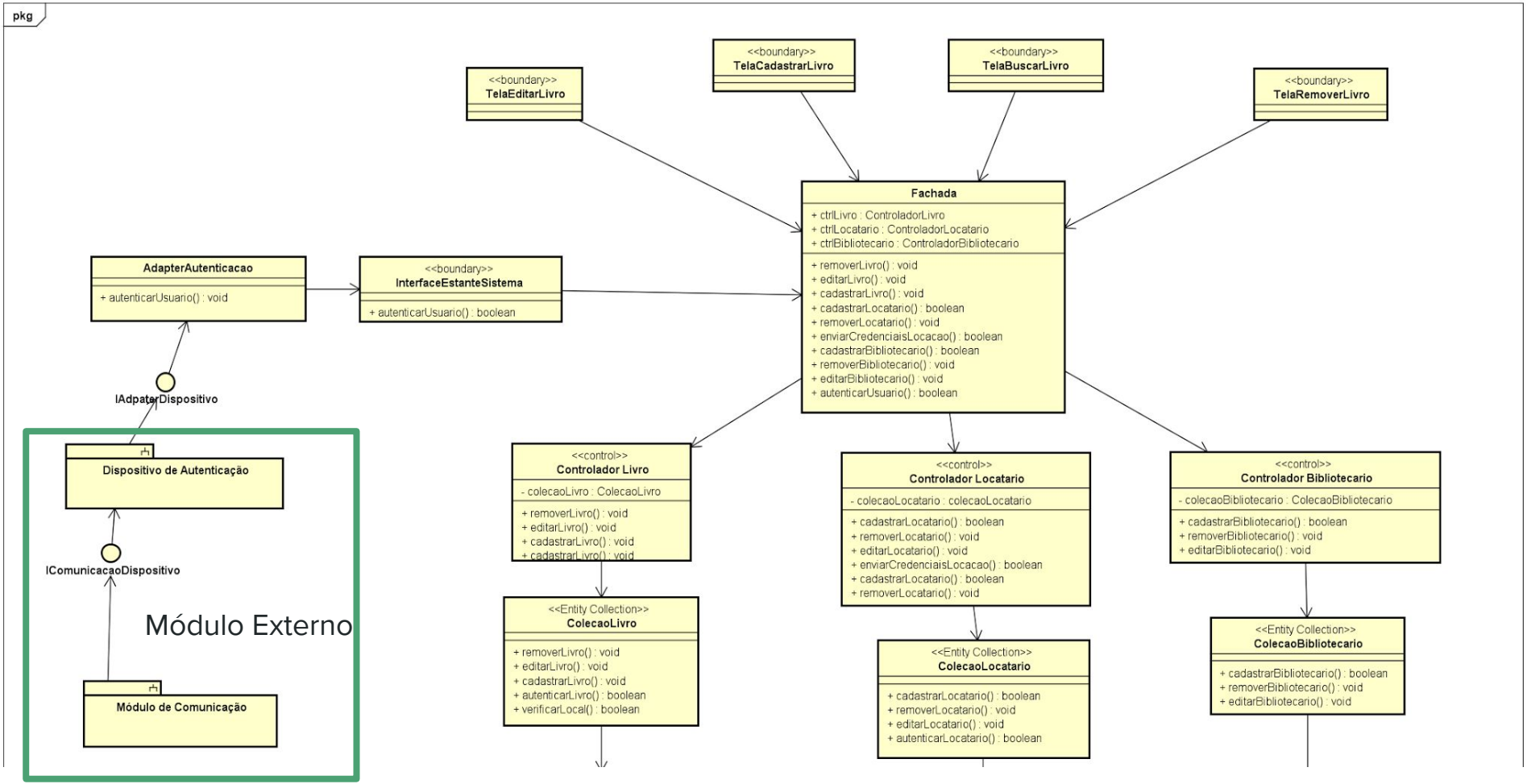


Diagrama de Sequência (Remover Livro)

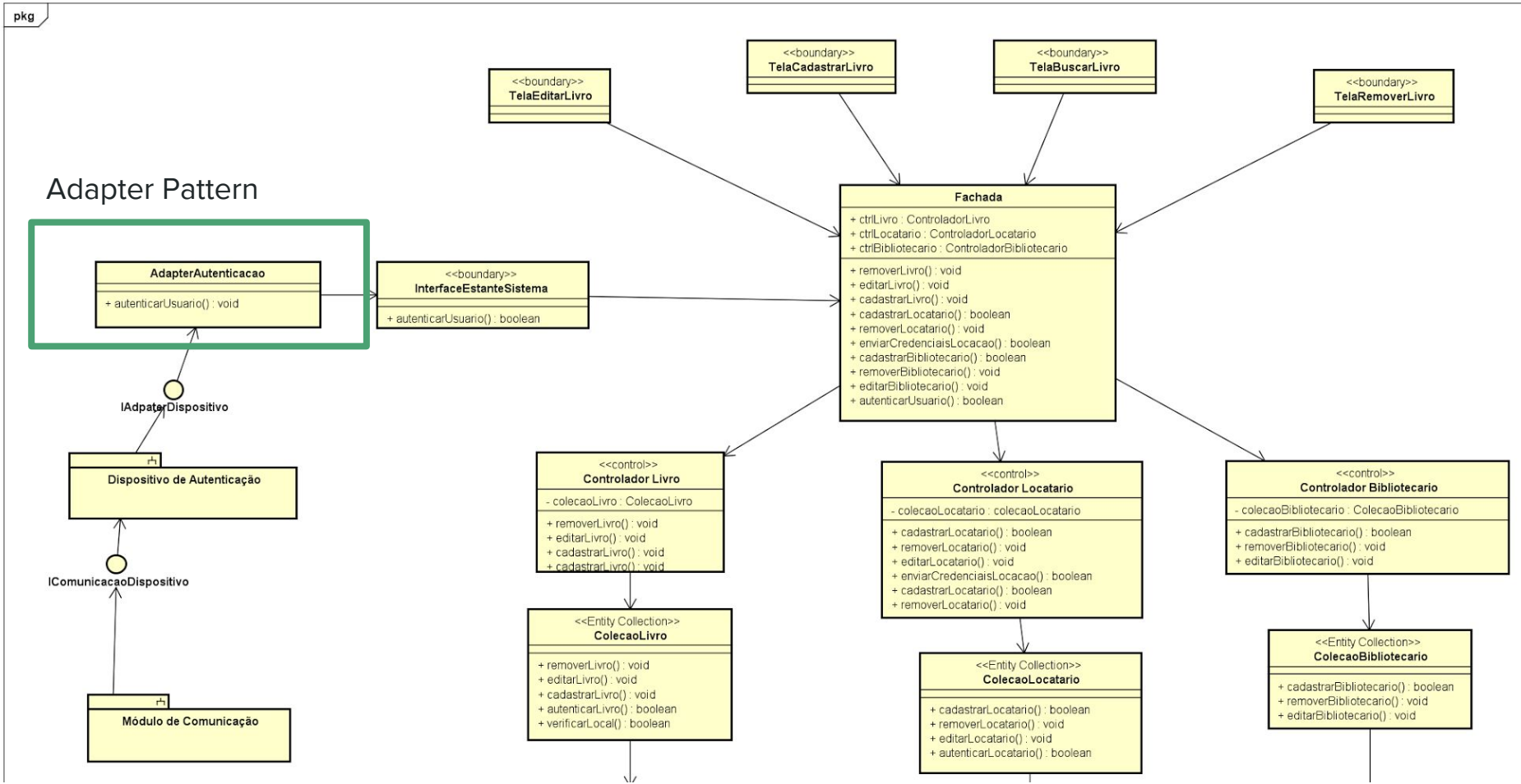


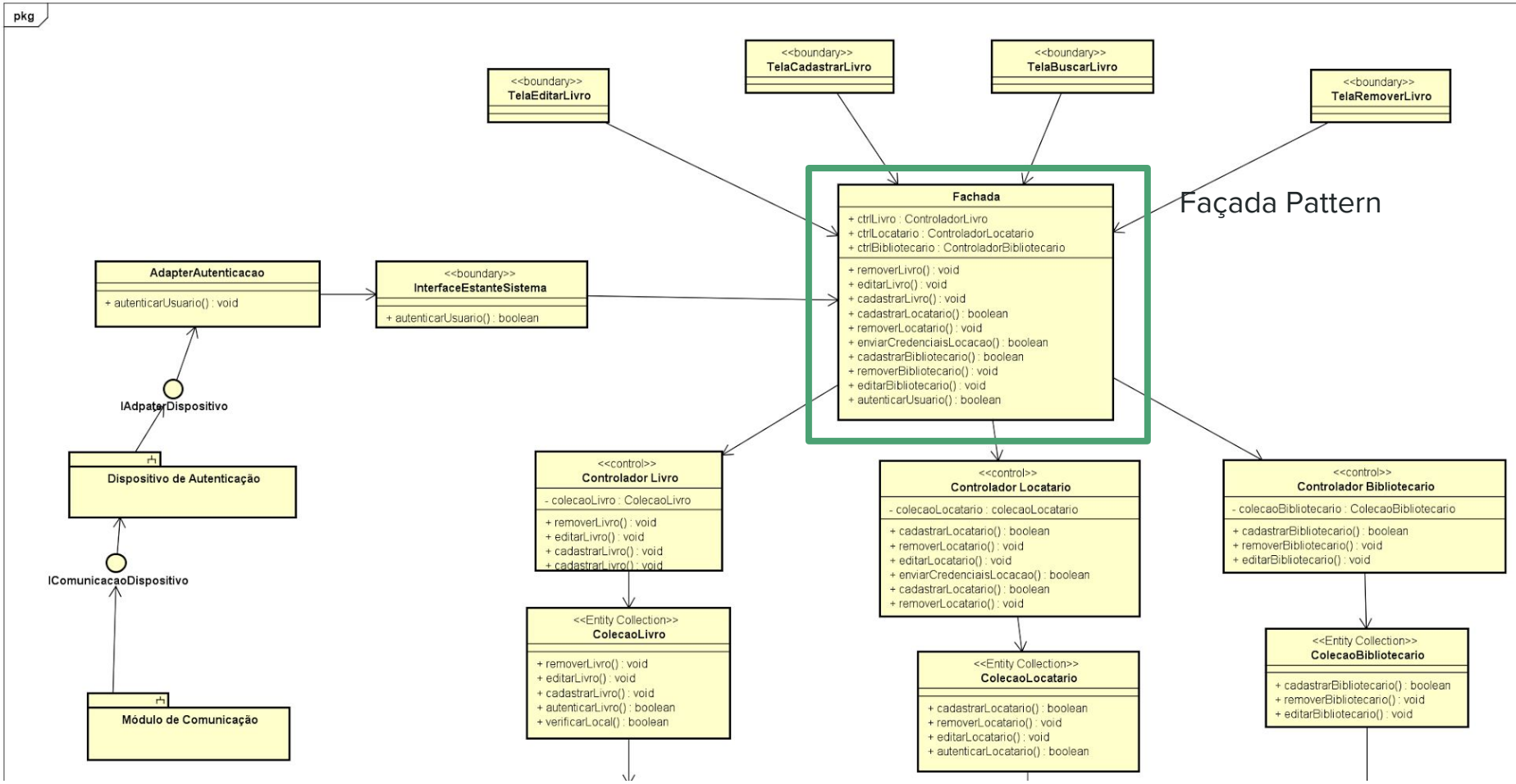
Arquitetura do Sistema

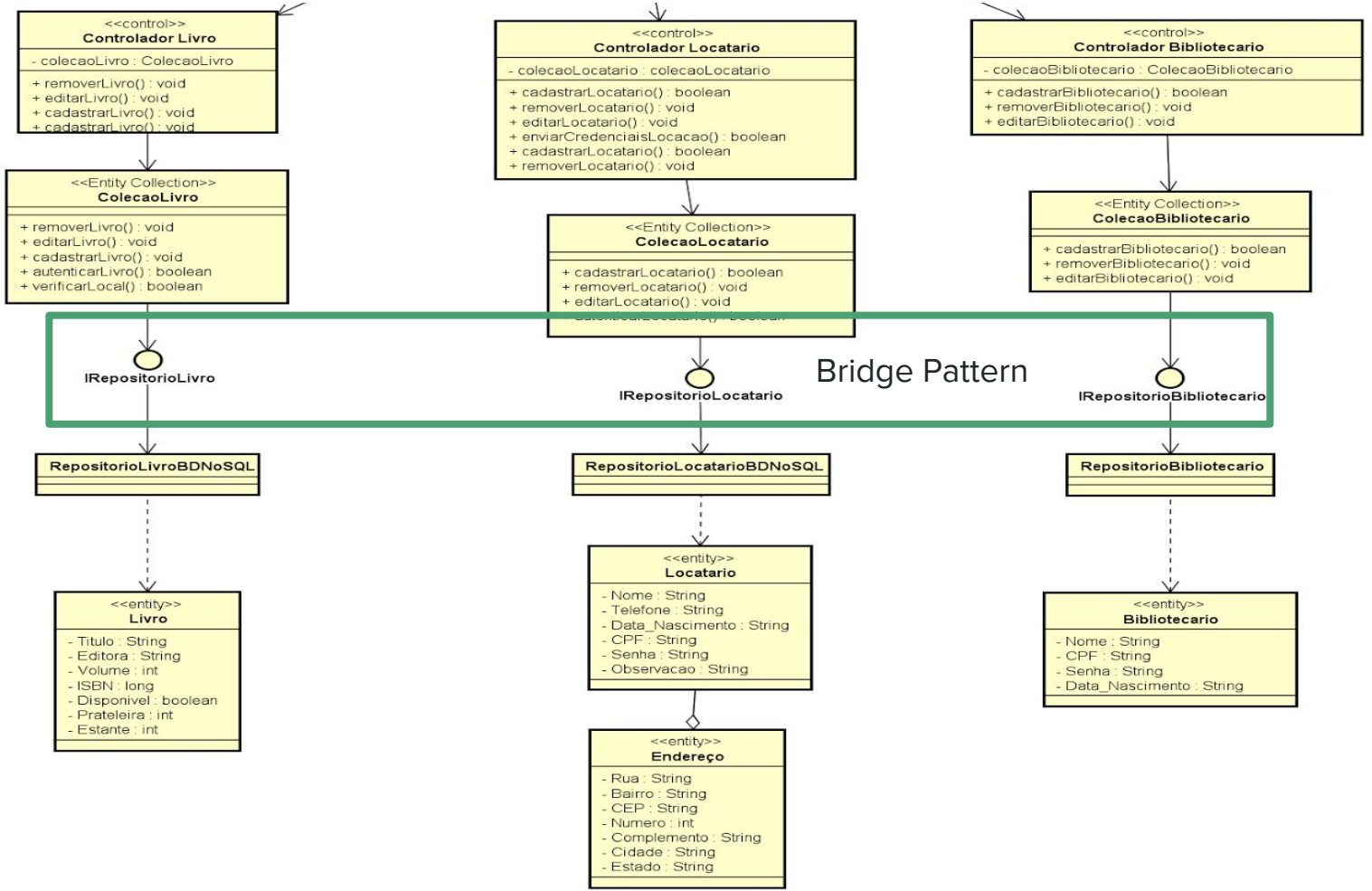

pkg
<<boundary>>
TelaEditarLivro
<<boundary>>
TelaCadastrarLivro
<<boundary>>
TelaBuscarLivro
<<boundary>>
TelaRemoverLivro
Fachada
+ ctrlLivro : ControladorLivro
+ ctrlLocatario : ControladorLocatario
+ ctrlBibliotecario : ControladorBibliotecario
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
+ autenticarUsuario() : boolean
AdapterAutenticacao
+ autenticarUsuario() : void
<<boundary>>
InterfaceEstanteSistema
+ autenticarUsuario() : boolean
IAdpaterDispositivo
Dispositivo de Autenticação
IComunicacaoDispositivo
Módulo de Comunicação
<<control>>
Controlador Livro
- colecaoLivro : ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLivro() : void
<<Entity Collection>>
ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ autenticarLivro() : boolean
+ verificarLocal() : boolean
IRepositorioLivro
RepositorioLivroBDNoSQL
<<entity>>
Livro
- Titulo : String
- Editora : String
- Volume : int
- ISBN : long
- Disponivel : boolean
- Prateleira : int
- Estante : int
<<control>>
Controlador Locatario
- colecaoLocatario : colecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
<<Entity Collection>>
ColecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ autenticarLocatario() : boolean
IRepositorioLocatario
RepositorioLocatarioBDNoSQL
<<entity>>
Locatario
- Nome : String
- Telefone : String
- Data_Nascimento : String
- CPF : String
- Senha : String
- Observacao : String
<<entity>>
Endereço
- Rua : String
- Bairro : String
- CEP : String
- Numero : int
- Complemento : String
- Cidade : String
- Estado : String
<<control>>
Controlador Bibliotecario
- colecaoBibliotecario : ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
<<Entity Collection>>
ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
IRepositorioBibliotecario
RepositorioBibliotecario
<<entity>>
Bibliotecario
- Nome : String
- CPF : String
- Senha : String
- Data_Nascimento : String


pkg
<<boundary>>
TelaEditarLivro
<<boundary>>
TelaCadastrarLivro
<<boundary>>
TelaBuscarLivro
<<boundary>>
TelaRemoverLivro
Fachada
+ ctrlLivro : ControladorLivro
+ ctrlLocatario : ControladorLocatario
+ ctrlBibliotecario : ControladorBibliotecario
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
+ autenticarUsuario() : boolean
AdapterAutenticacao
+ autenticarUsuario() : void
<<boundary>>
InterfaceEstanteSistema
+ autenticarUsuario() : boolean
IAdpaterDispositivo
Dispositivo de Autenticação
IComunicacaoDispositivo
Módulo de Comunicação
Módulo Externo
<<control>>
Controlador Livro
- colecaoLivro : ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLivro() : void
<<Entity Collection>>
ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ autenticarLivro() : boolean
+ verificarLocal() : boolean
<<control>>
Controlador Locatario
- colecaoLocatario : colecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
<<Entity Collection>>
ColecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ autenticarLocatario() : boolean
<<control>>
Controlador Bibliotecario
- colecaoBibliotecario : ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
<<Entity Collection>>
ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void

Adapter Pattern



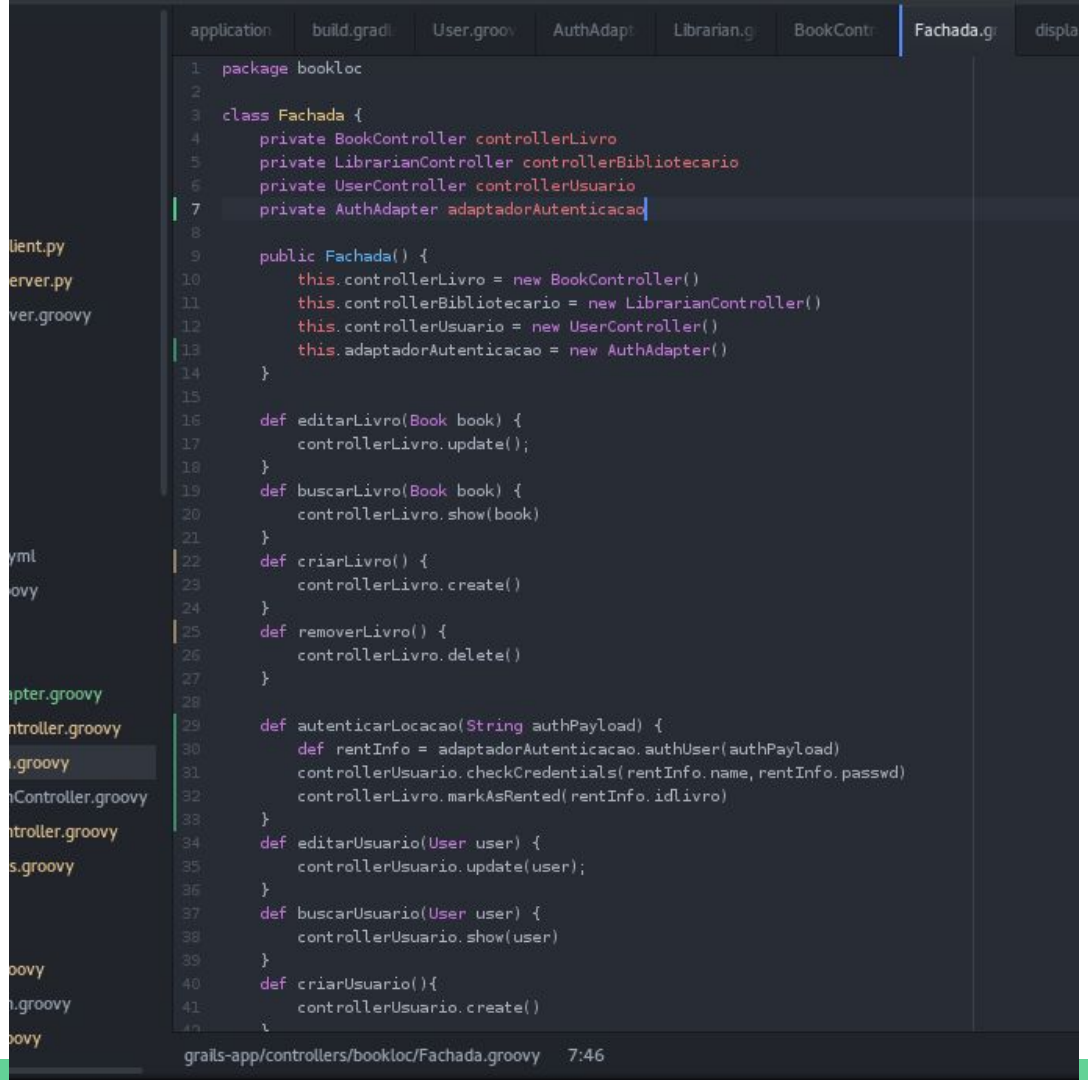

pkg
<<boundary>>
TelaEditarLivro
<<boundary>>
TelaCadastrarLivro
<<boundary>>
TelaBuscarLivro
<<boundary>>
TelaRemoverLivro
Fachada
+ ctrlLivro : ControladorLivro
+ ctrlLocatario : ControladorLocatario
+ ctrlBibliotecario : ControladorBibliotecario
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
+ autenticarUsuario() : boolean
Façada Pattern
AdapterAutenticacao
+ autenticarUsuario() : void
<<boundary>>
InterfaceEstanteSistema
+ autenticarUsuario() : boolean
IAdpaterDispositivo
Dispositivo de Autenticação
IComunicacaoDispositivo
Módulo de Comunicação
<<control>>
Controlador Livro
- colecaoLivro : ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLivro() : void
<<Entity Collection>>
ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ autenticarLivro() : boolean
+ verificarLocal() : boolean
<<control>>
Controlador Locatario
- colecaoLocatario : colecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
<<Entity Collection>>
ColecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ autenticarLocatario() : boolean
<<control>>
Controlador Bibliotecario
- colecaoBibliotecario : ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
<<Entity Collection>>
ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void


<<control>>
Controlador Livro
- colecaoLivro : ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ cadastrarLivro() : void
<<Entity Collection>>
ColecaoLivro
+ removerLivro() : void
+ editarLivro() : void
+ cadastrarLivro() : void
+ autenticarLivro() : boolean
+ verificarLocal() : boolean
IRepositorioLivro
RepositorioLivroBDNoSQL
<<entity>>
Livro
- Titulo : String
- Editora : String
- Volume : int
- ISBN : long
- Disponivel : boolean
- Prateleira : int
- Estante : int
<<control>>
Controlador Locatario
- colecaoLocatario : colecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
+ enviarCredenciaisLocacao() : boolean
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
<<Entity Collection>>
ColecaoLocatario
+ cadastrarLocatario() : boolean
+ removerLocatario() : void
+ editarLocatario() : void
IRepositorioLocatario
RepositorioLocatarioBDNoSQL
<<entity>>
Locatario
- Nome : String
- Telefone : String
- Data_Nascimento : String
- CPF : String
- Senha : String
- Observacao : String
<<entity>>
Endereço
- Rua : String
- Bairro : String
- CEP : String
- Numero : int
- Complemento : String
- Cidade : String
- Estado : String
<<control>>
Controlador Bibliotecario
- colecaoBibliotecario : ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
<<Entity Collection>>
ColecaoBibliotecario
+ cadastrarBibliotecario() : boolean
+ removerBibliotecario() : void
+ editarBibliotecario() : void
IRepositorioBibliotecario
RepositorioBibliotecario
<<entity>>
Bibliotecario
- Nome : String
- CPF : String
- Senha : String
- Data_Nascimento : String
Bridge Pattern

Implementação

```python
# -*- coding: utf-8 -*-
import socket
import sys
import json

HOST = 'localhost'    # The remote host
PORT = 50008          # The same port as used by the server
s = None
for res in socket.getaddrinfo(HOST, PORT, socket.AF_UNSPEC, socket.SOCK_STREAM):
    af, socktype, proto, canonname, sa = res
    try:
        s = socket.socket(af, socktype, proto)
    except OSError as msg:
        s = None
        continue
    try:
        s.connect(sa)
    except OSError as msg:
        s.close()
        s = None
        continue
    break

if s is None:
    print('could not open socket')
    sys.exit(1)

auth_info = {'name':'','passwd':'','bookid':0}
auth_info['name'] = input('Name: ')
auth_info['passwd'] = input('Passwd: ')
auth_info['bookid'] = input('BookId: ')
json_auth_info = json.dumps(auth_info)

s.sendall(bytes(json_auth_info,'utf-8'))
data = s.recv(1024)
s.close()

print('Received', repr(data))

```

displayAuth/display_auth_client.py 22:10


application
build.gradl
User.groov
AuthAdapt
Librarian.g
BookContr
Fachada.g
displa
1 package bookloc
2
3 class Fachada {
4 private BookController controllerLivro
5 private LibrarianController controllerBibliotecario
6 private UserController controllerUsuario
7 private AuthAdapter adaptadorAutenticacao
8
9 public Fachada() {
10 this.controllerLivro = new BookController()
11 this.controllerBibliotecario = new LibrarianController()
12 this.controllerUsuario = new UserController()
13 this.adaptadorAutenticacao = new AuthAdapter()
14 }
15
16 def editarLivro(Book book) {
17 controllerLivro.update();
18 }
19 def buscarLivro(Book book) {
20 controllerLivro.show(book)
21 }
22 def criarLivro() {
23 controllerLivro.create()
24 }
25 def removerLivro() {
26 controllerLivro.delete()
27 }
28
29 def autenticarLocacao(String authPayload) {
30 def rentInfo = adaptadorAutenticacao.authUser(authPayload)
31 controllerUsuario.checkCredentials(rentInfo.name, rentInfo.passwd)
32 controllerLivro.markAsRented(rentInfo.idlivro)
33 }
34 def editarUsuario(User user) {
35 controllerUsuario.update(user);
36 }
37 def buscarUsuario(User user) {
38 controllerUsuario.show(user)
39 }
40 def criarUsuario(){
41 controllerUsuario.create()
grails-app/controllers/bookloc/Fachada.groovy 7:46


_client.py
_server.py
erver.groovy
import groovy.ui.GroovySocketServer
import groovy.json.JsonSlurper
import java.net.ServerSocket
class AuthAdapter {
def authUser(String authPayload){
def jsonMap = slurper.parseText(authPayload)
jsonMap
}
}