



**FCT – FACULDADE UNIME DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**UNIDADE LAURO DE FREITAS**

**CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**PROGRAMAÇÃO PARA WEB I**  
**SISTEMA WEB PARA LOCAÇÃO DE AUDIOBOOKS**

ALBERT LEMOS  
CLAUDIO MORAIS  
MARCIO AMORIM  
MOISÉS FARIAS  
PAULO ROBERTO

LAURO DE FREITAS - BA

2017

**FCT – FACULDADE UNIME DE CIÊNCIAS EXATAS E TECNOLÓGICAS**

**UNIDADE LAURO DE FREITAS**

**CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**PROGRAMAÇÃO PARA WEB I**

**SISTEMA WEB PARA LOCAÇÃO DE AUDIOBOOKS**

Esta atividade foi desenvolvida a pedido do Professor Wellington Lacerda, da disciplina de Programação para Web I, com o objetivo de explorar e fixar os conceitos e habilidades teóricas e técnicas a respeito do conteúdo abordado em sala de aula.

LAURO DE FREITAS - BA

2017

## Sumário

1 - Descrição da Revisão do Documento.	p.4
2 - Lista de Figuras.	p.5
3 - Lista de Tabelas.	p.6
4 - Lista de Siglas.	p.7
5 – Introdução.	p.8
6 – Tema do Projeto.	p.9
6.1 – Delimitação do problema.	p.9
6.2 – Objetivo do Projeto.	p.9
6.3 – Justificativa da Escolha do Tema.	p.9
6.4 – Método de Trabalho.	p.10
7 – Descrição Geral do Sistema.	p.11
8 – Requisitos do Sistema.	p.12
8.1 - Prioridade dos Requisitos.	p.12
8.2 – Requisitos Funcionais.	p.13
8.3 – Requisitos Não Funcionais.	p.15
9 – UML- Linguagem de Modelagem Unificada.	p.16
9.1 – Diagrama de Caso de Uso do Sistema.	p.16
9.2 – Diagrama de Classe e Pacote.	p.21
9.3 – Diagrama de Atividade.	p.23
10 – Framework Laravel: <i>Views, Controllers, Models</i> e Rotas do Sistema	p.52
11 – Banco de Dados do Sistema	p.58
12 – Estimativa de Custo do Projeto	p.59
13 – Considerações.	p.60
14 – Referências.	p.61

## 1 - Descrição da Revisão do Documento

Data da Revisão	Responsável	Número da Revisão	Descrição
28/08/2017	Albert Lemos, Moises Farias	1	Pesquisa de Tecnologias
04/09/2017	Paulo, Marcio, Claudio	2	Análise de Requisitos
11/09/2017	Albert Lemos, Moises Farias	3	Diagrama de Caso de Uso
18/09/2017	Paulo, Marcio, Claudio	4	Diagrama de Classe e Pacote
25/09/2017	Albert Lemos, Moises Farias	5	Diagrama de Atividade
02/10/2017	Claudio, Paulo, Marcio	6	Interface Gráfica
09/10/2017	Albert Lemos, Moises Farias	7	Laravel: View, Control, Model e Rotas
16/10/2017	Albert Lemos, Moises, Paulo, Marcio, Claudio	8	Entrega da documentação
23/10/2017	Albert Lemos, Moises, Paulo, Marcio, Claudio	9	Apresentação do Projeto

## 2 - Lista de Figuras

Figura 1: Exemplo de um sistema de locação de AudioBooks.	p.11
Figura 2 – Diagrama de Caso de Uso do Sistema de locação de AudioBooks.	p.17
Figura 3 – Diagrama de Pacotes e de classes do Sistema de locação de AudioBooks.	p.22
Figura 4 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Usuário.	p.24
Figura 5 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Usuário.	p.26
Figura 6 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Usuário.	p.28
Figura 7 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Título.	p.30
Figura 8 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Título.	p.32
Figura 9 – Diagrama de Atividade do Sistema de locação de AudioBooks: Excluir Título.	p.34
Figura 10 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Título.	p.36
Figura 11 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Categoria.	p.38
Figura 12 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Categoria.	p.40
Figura 13 – Diagrama de Atividade do Sistema de locação de AudioBooks: Excluir Categoria.	p.42
Figura 14 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Categoria	p.44
Figura 15 – Diagrama de Atividade do Sistema de locação de AudioBooks: Locar Título.	p.46
Figura 16 – Diagrama de Atividade do Sistema de locação de AudioBooks: Reservar Título.	p.48
Figura 17 – Diagrama de Atividade do Sistema de locação de AudioBooks: Solicitar Relatório.	p.50
Figura 18 – Diagrama MVC Laravel.	p.52
Figura 19 – Mecanismo de Rotas Laravel.	p.52
Figura 20 – Exemplo de <i>Controllers</i> Laravel.	p.53
Figura 21 – Exemplo de <i>Model</i> Laravel.	p.53
Figura 22 – Exemplo de <i>View</i> Laravel.	p.54
Figura 23 – Banco de dados do Sistema.	p. 58

### **3 - Lista de Tabelas**

Tabela 1 – Estimativa do Custo do Projeto do sistema de locação de AudioBooks.

p. 59

## 4 - Lista de Siglas

OO – Orientada a Objeto.

AOO – Análise Orientada a Objeto.

UML – Linguagem de Modelagem Unificada.

RF – Requisitos Funcionais.

RNF – Requisitos Não Funcionais.

Rota – Caminho/Endereço.

Control – Controlador.

View – Interface Gráfica.

Model – Modelos.

## 5 - Introdução

Este trabalho foi elaborado com o objetivo de desenvolver a documentação da implementação de uma aplicação para web com a funcionalidade de realizar locação de AudioBooks, colocando em prática conceitos e técnicas de estudo aprendidas em sala de aula, como ferramentas que são utilizadas para o desenvolvimento web (PHP, JavaScript, HTML, CSS, dentre outras), a análise de requisitos onde iremos realizar o levantamento e especificação dos requisitos funcionais e dos requisitos não funcionais, os diagramas de caso de uso, quando estamos analisando a criação de um software por meio de orientação a objeto, diagrama de classe e os diagramas de atividade, ou seja, utilizar a UML que é uma linguagem de modelagem unificada do padrão de orientação a objeto. O software irá funcionar num servidor web que irá fornecer o serviço aos seus clientes. Este tipo de programa é encontrado na web com o objetivo de disponibilizar para os usuários os áudios de livros de acordo com a necessidade dos mesmos.



## **6 – Tema do Projeto**

O projeto de software em questão tem como tema principal analisar e especificar os requisitos essenciais para a criação da documentação a respeito do desenvolvimento de uma aplicação web com o objetivo de disponibilizar audiobooks para seus usuários conforme a necessidade dos mesmos de forma eficiente e eficaz.

### **6.1 – Delimitação do problema**

Como disponibilizar áudios de livros na WEB de forma eficiente e eficaz para os clientes que possuem tal necessidade, levando em consideração a comodidade e a qualidade na prestação do serviço?

### **6.2 – Objetivo do Projeto**

#### **Objetivo Principal**

O objetivo principal do projeto é disponibilizar para o cliente um sistema onde o mesmo possa fazer a locação do áudio de livros conforme a sua necessidade de forma eficiente e eficaz, atendendo as expectativas do cliente.

#### **Objetivo específico**

Criação de uma aplicação WEB, que possa disponibilizar o link de audiobooks conforme a necessidade de locação de cada usuário, de forma eficiente e eficaz para que os mesmos possam ter mais comodidade e facilidade para acessar esses recursos.

### **6.3 – Justificativa da Escolha do Tema**

Muitas pessoas procuram por mais comodidade no seu dia a dia, e um sistema de audiobooks traz exatamente isso, onde poderá ser solicitado a locação de áudio de livros que geralmente possuem um valor bem alto considerando a compra de um exemplar novo. Através desse sistema web o áudio do livro poderá sair praticamente gratuito considerando o valor real do exemplar físico.

## **6.4 – Método de Trabalho**

Para elaborar este documento de software foi utilizado como base pesquisas feitas a respeito de outros sistemas web que oferecem o mesmo serviço. A modelagem utilizada para implementar o sistema segue o paradigma de orientação a objeto, ou seja, onde tudo é considerado um objeto do mundo real. Este tipo de modelagem tem como principais características a criação de classes modelos, métodos funcionais e atributos. Geralmente os objetos trabalham interconectados realizando requisições uns aos outros para executar uma determinada funcionalidade do sistema.

Como metodologia de desenvolvimento foi adotada a metodologia ágil de desenvolvimento de software Scrum com a utilização do software de gerenciamento de projetos chamado Trello.

## 7 – Descrição Geral do Sistema

O Sistema de *AudioBooks* permite aos seus usuários adquirir títulos (áudio-livros) por um determinado período de tempo com qualidade e facilidade disponibilizando-os a qualquer momento que o usuário desejar com uma interface amigável e responsiva.

A aplicação permite cadastro de cliente, aluguel de títulos e reserva de títulos no módulo usuário além de notificá-lo quando títulos reservados estiverem disponíveis, e cadastro de títulos, títulos que estão por vir, além de disponibilizar relatórios gerenciais no módulo administrador.

Para o desenvolvimento da aplicação está sendo utilizado um *framework*, pois o mesmo captura as funcionalidades comuns à várias aplicações e por consequência acelera o processo de implementação. O *Laravel 5.2* é a *framework* utilizada para o desenvolvimento da aplicação, pois conta com um ferramental com evolução constante e seus releases são feitos de 6 em 6 meses.

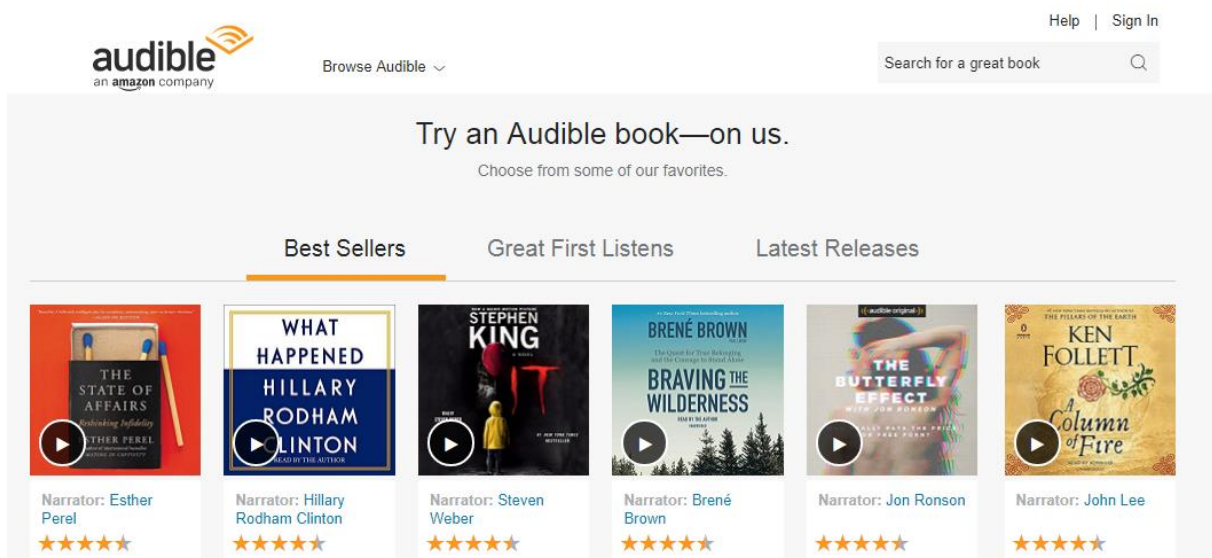


Figura 1: Exemplo de um sistema de locação de *AudioBooks* (Audible Inc.).

Disponível em: < <https://www.audible.com/>>. Acessado em: 12 de Outubro de 2017.

## 8 – Requisitos do Sistema

Para especificar quais requisitos deverão ser implementados no sistema é necessário que a equipe de desenvolvimento juntamente com o cliente entre em comum acordo sobre as reais necessidades dos usuários. Segundo Bezerra (2015, p.32) “um requisito é uma condição ou capacidade que deve ser alcançada ou possuída por um sistema ou um de seus componentes. ” Os requisitos podem ser denominados funcionais e não funcionais. Os requisitos funcionais definem as funcionalidades do sistema, já os requisitos não funcionais declaram as características de qualidade que o sistema deve possuir (BEZERRA, 2015).

### 8.1 - Prioridade dos Requisitos

Na área de engenharia de software e de análise de sistemas, além de especificar quais são os requisitos, também é necessário ressaltar níveis de prioridade entre eles, ou seja, é necessário informar o quão determinado requisito é importante para o funcionamento do sistema em si. Neste estudo de caso iremos utilizar denominações como “essencial”, “importante” e “desejável”.

**Essencial:** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são vitais para funcionamento do software em questão e devem ser implementados sob a perspectiva de ser prioridade zero durante o desenvolvimento.

**Importante:** é o requisito sem o qual o sistema pode entrar em funcionamento, mas não de maneira satisfatória, ou seja, não irá atender todas as expectativas do cliente. É recomendado que os mesmos sejam implementados para evitar problemas futuros, mas senão forem o software poderá ser desenvolvido e utilizado mesmo assim.

**Desejável:** é o requisito que não compromete as funcionalidades básicas do software, isto é, consegue funcionar de maneira satisfatória sem eles e atende as necessidades especificadas pelos usuários. Geralmente este tipo de prioridade é direcionada para requisitos que podem ser implementados em versões posteriores do próprio software.

## 8.2 – Requisitos Funcionais

### **RF01 – Cadastrar Usuário.**

Este requisito permitirá que o Usuário se cadastre no sistema.

Prioridade: Essencial.

### **RF02 – Alterar Cadastro de Usuário**

Este requisito permitirá que o Usuário altere o seu cadastro (senha de acesso).

Prioridade: Importante

### **RF 03 – Pesquisar Título.**

Este requisito permitirá que seja realizada a consulta de títulos no sistema tanto pelo Administrador ou pelo usuário cliente.

Prioridade: Essencial.

### **RF04 – Excluir Título**

Este requisito permitirá que o Administrador possa excluir o cadastro de um título no sistema.

Prioridade: Importante.

### **RF05 – Locar Título**

Este requisito permitirá que o Usuário possa locar títulos, associando-os a seu cadastro.

Prioridade: Essencial.

### **RF06 – Reservar Título**

Este requisito permitirá que o Usuário possa reservar títulos, associando-os a seu cadastro, sendo lembrado quando o título estiver disponível para locação.

Prioridade: Essencial.

**RF07 – Cadastro de Título**

Este requisito permitirá que o Administrador possa cadastrar um título no sistema.

Prioridade: Essencial.

**RF08 – Alterar Título**

Este requisito permitirá que o Administrador possa alterar um título no sistema.

Prioridade: Importante.

**RF09 – Pesquisar Usuário**

Este requisito permitirá que o Administrador possa pesquisar um usuario no sistema.

Prioridade: Importante.

**RF10 – Solicitar Relatório**

Este requisito permitirá que o Administrador possa solicitar ao sistema um relatório no sistema. (Usuários cadastrados dentro de um período, Títulos cadastrados dentro de um período)

Prioridade: Importante.

**RF11 – Cadastrar Categoria**

Este requisito permitirá que o Administrador possa cadastrar uma categoria de título no sistema.

Prioridade: Essencial.

**RF12 – Alterar Categoria**

Este requisito permitirá que o Administrador possa alterar uma categoria de título no sistema.

Prioridade: Importante.

**RF13 – Pesquisar Categoria**

Este requisito permitirá que o Administrador possa pesquisar uma categoria de título no sistema.

Prioridade: Essencial.

**RF14 – Excluir Categoria**

Este requisito permitirá que o Administrador possa excluir uma categoria de título no sistema.

Prioridade: Importante.

**8.3 – Requisitos Não Funcionais****RNF01 – Conexão com banco de dados**

O sistema possuirá uma conexão com banco de dados para que os dados possam ser salvos e consultados sempre que necessário.

Prioridade: Essencial.

**RNF02 – Framework Laravel**

O sistema será desenvolvido utilizando a framework de desenvolvimento web Laravel 5.4.

Prioridade: Essencial.

**RNF03 – Plataforma Web**

O sistema será voltado para plataforma Web seguindo o paradigma de orientação a objeto, ou seja, o mesmo funcionará num servidor, onde os clientes poderão acessar os dados e fazer locação e reserva dos títulos cadastrados através de um navegador web.

Prioridade: Essencial.

## **9 – UML- Linguagem de Modelagem Unificada**

A UML nos dias atuais é considerada uma linguagem universal de modelagem de sistemas, utilizada para descrever e projetar sistemas por meio de diagramas que irão fazer parte da documentação do software (FOWLER, 2014). Geralmente este tipo de linguagem é utilizada quando estamos desenvolvendo um software utilizando o padrão de orientação a objetos. Segundo Bezerra (2015, p.20) o paradigma de orientação a objetos tem como principal objetivo diminuir a diferença semântica entre a realidade sendo modelada e os modelos que estão sendo construídos. Para este padrão tudo é considerado um objeto que está conectado a outros objetos para realizar uma determinada funcionalidade específica.

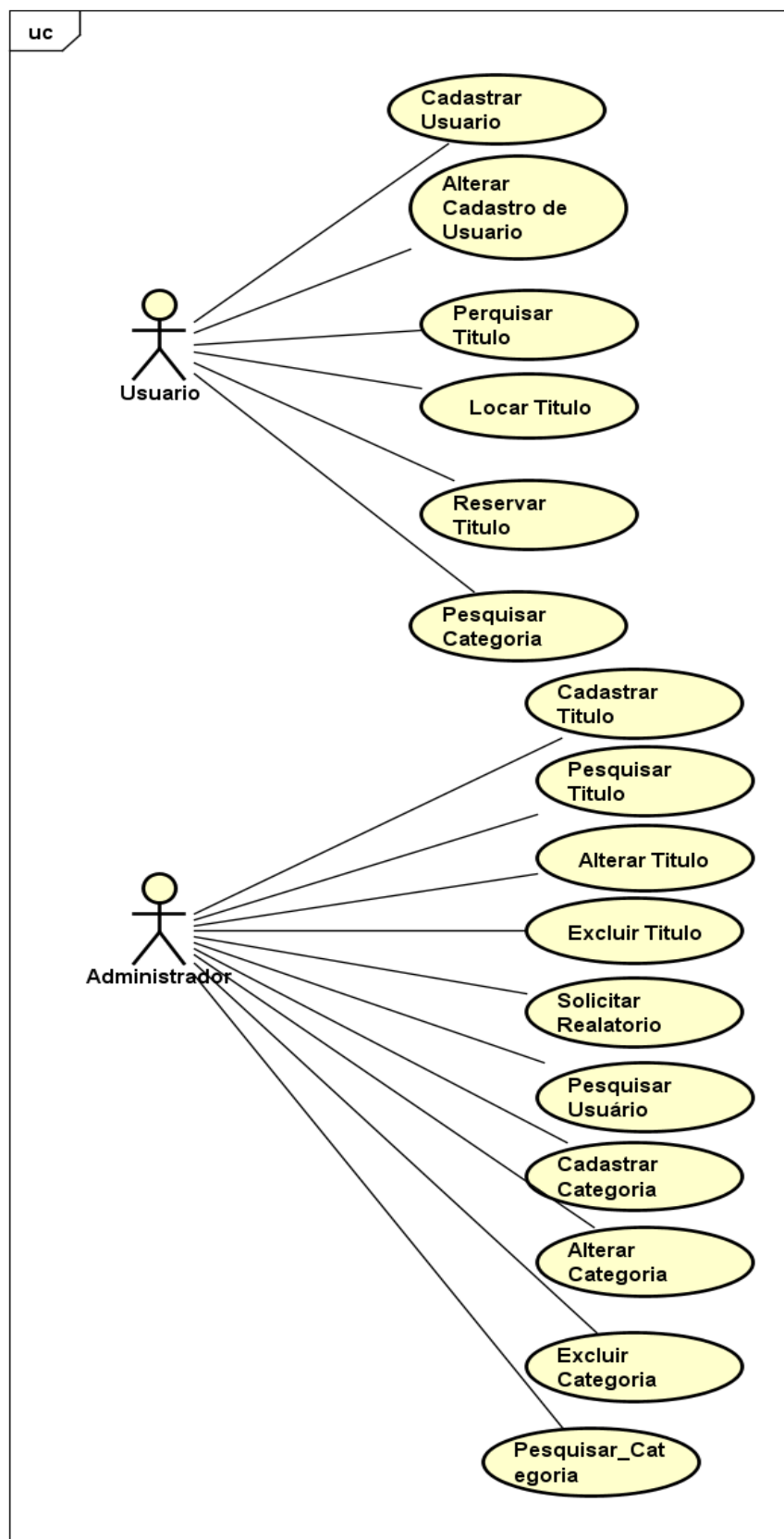
Esta linguagem de modelagem contempla alguns diagramas como de caso de uso, de pacote, de classe, de atividade, de sequencias, dentre outros. Abaixo segue os diagramas que terão a função de descrever comportamentos e funcionalidades do sistema de terminal de consulta de preço.

### **9.1 – Diagrama de Caso de Uso do Sistema**

O diagrama de caso de uso tem como principal objetivo especificar os atores e suas funcionalidades que estão relacionadas ao sistema (FOWLER, 2014).

Segue abaixo a figura que mostra o diagrama de caso de do sistema de locação de AudioBooks e suas especificações.





powered by Astah

Figura 2 – Diagrama de Caso de Uso do Sistema de locação de AudioBooks.  
(Fonte: Elaborada pelo Autor)

## **Especificação dos Casos de Uso do Sistema**

**Caso de Uso:** Manter Título.

**Ator:** Administrador.

**Finalidade:** Manutenção dos Títulos cadastrados no sistema.

**Visão Geral:** Este caso de uso é do tipo CRUD, ou seja, o administrador terá a funcionalidade de cadastrar, pesquisar, alterar e excluir Títulos do sistema. Este caso de uso tem a função de manter informações de títulos cadastrados para serem utilizadas em outras funcionalidades do próprio sistema.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Manter Categoria.

**Ator:** Administrador.

**Finalidade:** Manutenção das categorias de títulos cadastradas no sistema.

**Visão Geral:** Este caso de uso é do tipo CRUD, ou seja, o administrador terá a funcionalidade de cadastrar, pesquisar, alterar e excluir categorias de títulos do sistema. Este caso de uso tem a função de manter informações das categorias cadastradas para serem utilizadas em outras funcionalidades do próprio sistema.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Pesquisar Usuário.

**Ator:** Administrador.

**Finalidade:** Pesquisar informações de usuário do sistema.

**Visão Geral:** Este caso de uso terá a funcionalidade de buscar informações de usuários que usam o sistema para locar audiobooks.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Solicitar Relatório.

**Ator:** Administrador.

**Finalidade:** Gerar Relatórios do sistema.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o administrador possa solicitar relatórios no sistema a respeito de algumas transações como, por exemplo, uma lista dos audiobooks que estão locados.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Cadastrar Usuário.

**Ator:** Usuário.

**Finalidade:** Criar um novo usuário para o sistema.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o próprio usuário possa se cadastrar no sistema de audiobooks para ter acesso ao mesmo.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Alterar Cadastro de Usuário.

**Ator:** Usuário.

**Finalidade:** Alterar os dados do cadastro de um usuário já existente.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o próprio usuário possa alterar seus dados cadastrais com, por exemplo, a senha de acesso ao sistema.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Pesquisar Título.

**Ator:** Usuário

**Finalidade:** Consultar Títulos cadastrados.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o usuário possa consultar o registro de um título cadastrado através de uma tela do sistema.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Pesquisar Categoria.

**Ator:** Usuário

**Finalidade:** Consultar categorias cadastradas.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o usuário possa consultar o registro de uma categoria de título cadastrada através de uma tela do sistema.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Locar Título.

**Ator:** Usuário

**Finalidade:** Alugar um AudioBook.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o usuário possa alugar um audiobook que esteja cadastrado no sistema por um tempo determinado.

**Referência Cruzada:** Acessar Sistema.

**Caso de Uso:** Reservar Título.

**Ator:** Usuário

**Finalidade:** Reservar um AudioBook.

**Visão Geral:** Este caso de uso terá a funcionalidade de permitir que o usuário possa reservar um audiobook que esteja cadastrado no sistema como lançamento, ou seja, que ainda não está disponível para alugar, mas que estará disponível em breve. Com este caso de uso é criada uma fila de espera.

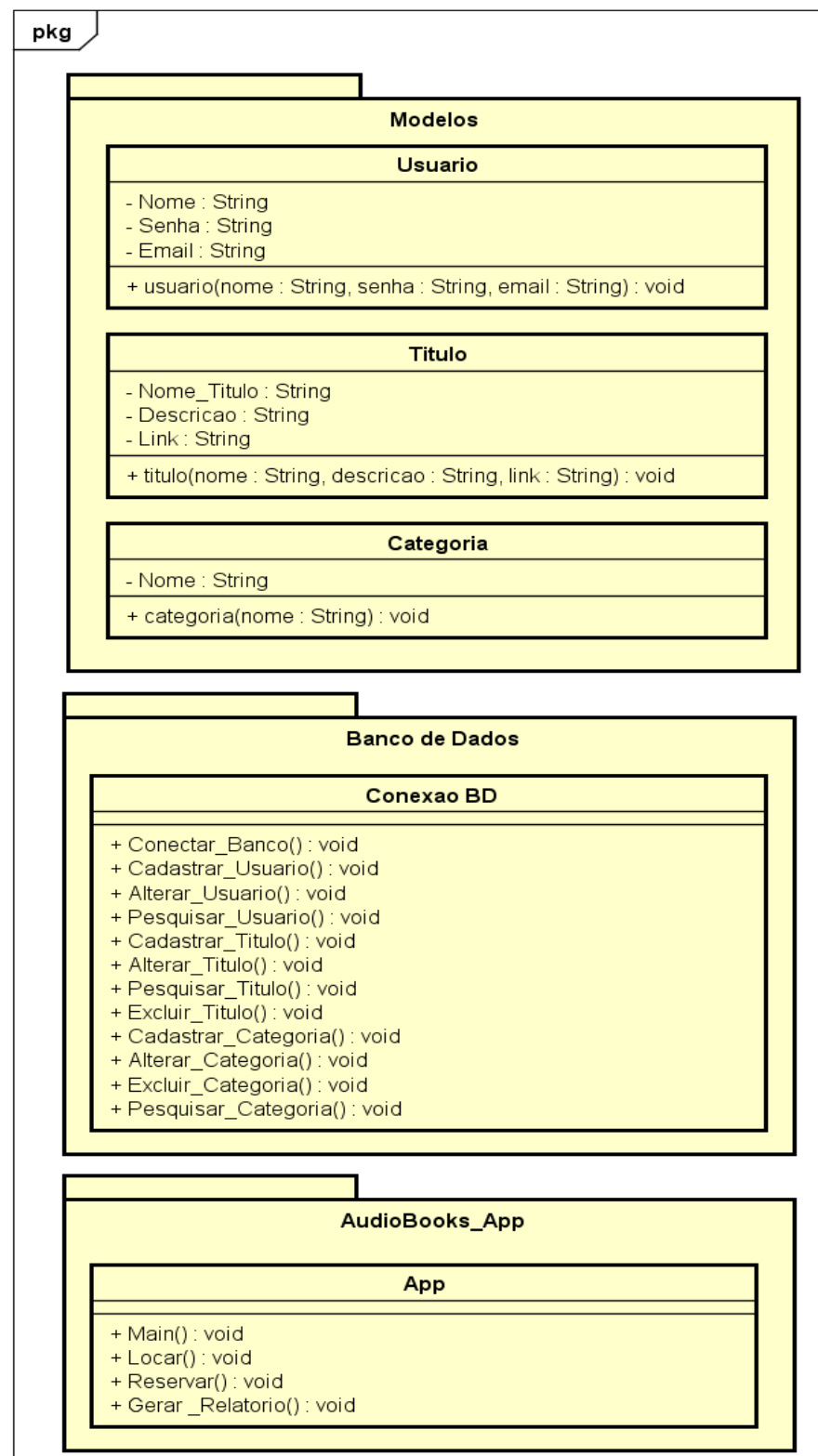
**Referência Cruzada:** Acessar Sistema.

## 9.2 – Diagrama de Classe e Pacote

O diagrama de pacote e o de classe também fazem parte da linguagem de modelagem de sistema que é utilizada em projetos e na documentação durante a criação de um software.

O diagrama de pacotes possui a função de organizar o projeto, ou seja, separar conteúdos que pertencem ao um mesmo projeto. Esse tipo de diagrama costuma organizar o projeto em camadas. Já o diagrama de classe tem a funcionalidade de servir como um modelo para a criação de um determinado objeto que será utilizado dentro do sistema.

O diagrama de pacote e de classes do sistema será mostrado na figura abaixo:



powered by Astah

Figura 3 – Diagrama de Pacotes e de classes do Sistema de locação de AudioBooks.

(Fonte: Elaborada pelo Autor)

## **Descrição geral do diagrama de pacote e de classe do sistema**

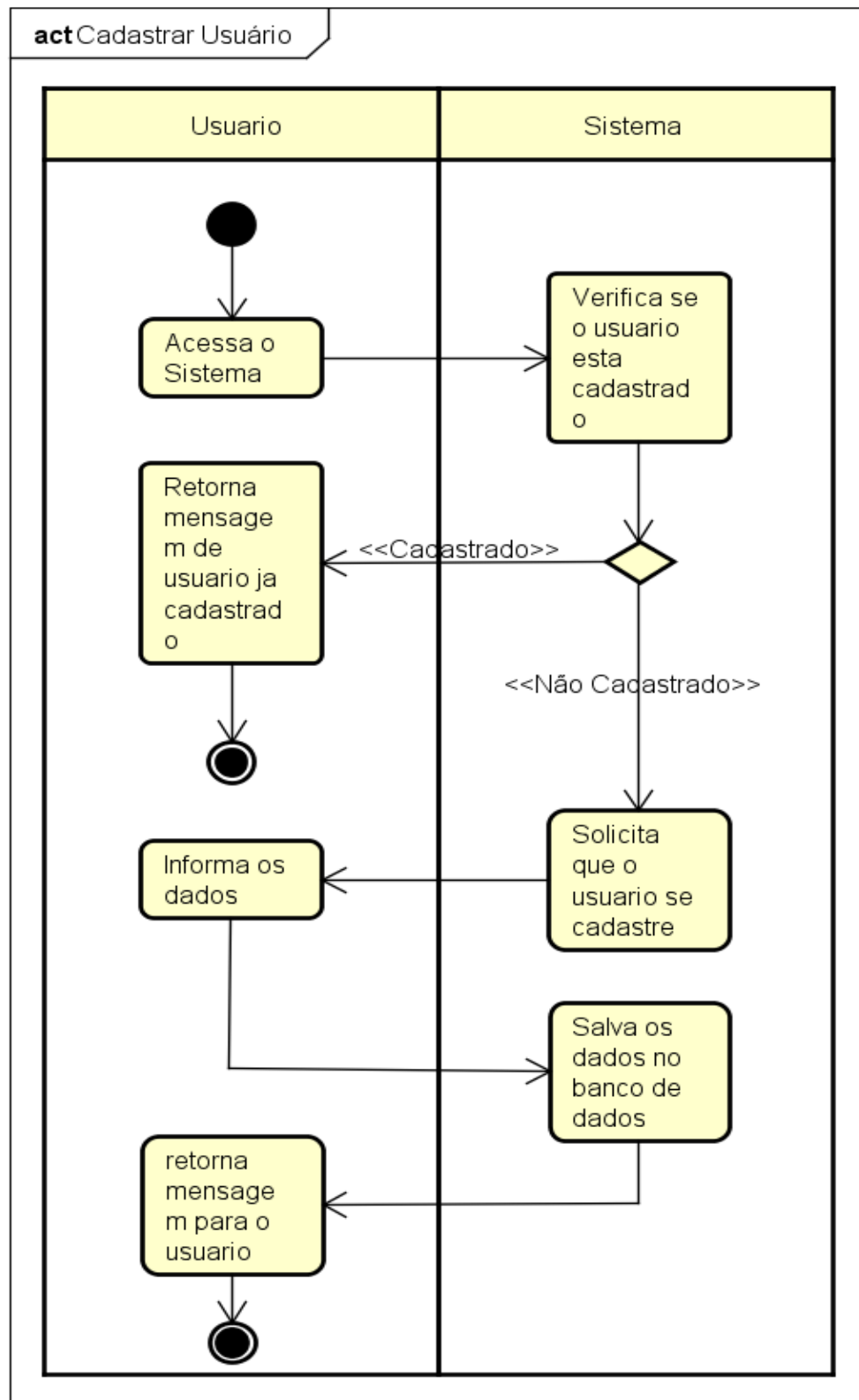
Como é mostrado na figura acima existem três pacotes: No primeiro pacote temos as classes modelos para criação dos seguintes objetos: usuário, título e categoria. No segundo pacote temos uma classe que irá ter alguns métodos para se conectar com o banco e executar algumas operações como cadastrar, alterar, pesquisar e excluir informações da base de dados. No terceiro pacote temos a classe App que será executada por meio do método principal do projeto e mais alguns métodos como locar, reservar e solicitar relatório que fazem parte da lógica do sistema. Neste terceiro pacote também encontra-se a interface gráfica do sistema.

### **9.3 – Diagrama de Atividade**

O diagrama de atividade tem como principal objetivo mostrar o comportamento do sistema durante a execução de um determinado processo. Segundo Guedes (2014, p.89) “Esse diagrama é utilizado, como o próprio nome diz, para modelar atividades que podem ser um método ou um algoritmo, ou mesmo um processo completo. ” .

Segue abaixo os diagramas de atividade do sistema de locação de AudioBooks:

### Diagrama de atividade para cadastrar Usuário



powered by Astah

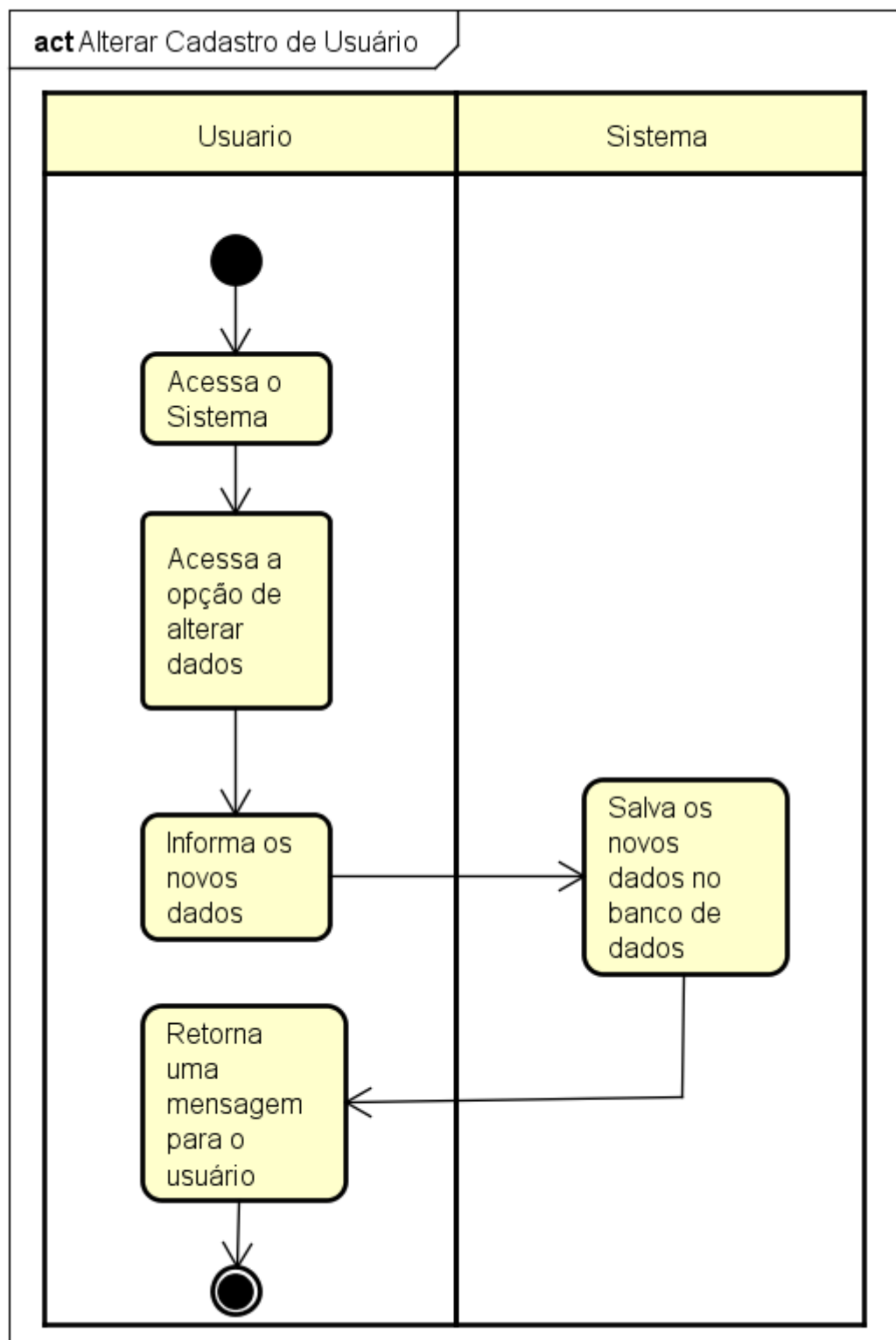
Figura 4 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Usuário.

(Fonte: Elaborada pelo Autor)



Este diagrama especifica como o sistema irá se comportar durante o cadastro de um novo usuário, onde é necessário realizar o acesso, informar os dados do novo usuário e se o mesmo ainda não tiver sido cadastrado os dados são salvos na base de dados.

#### **Diagrama de atividade para Alterar Cadastro de Usuário**



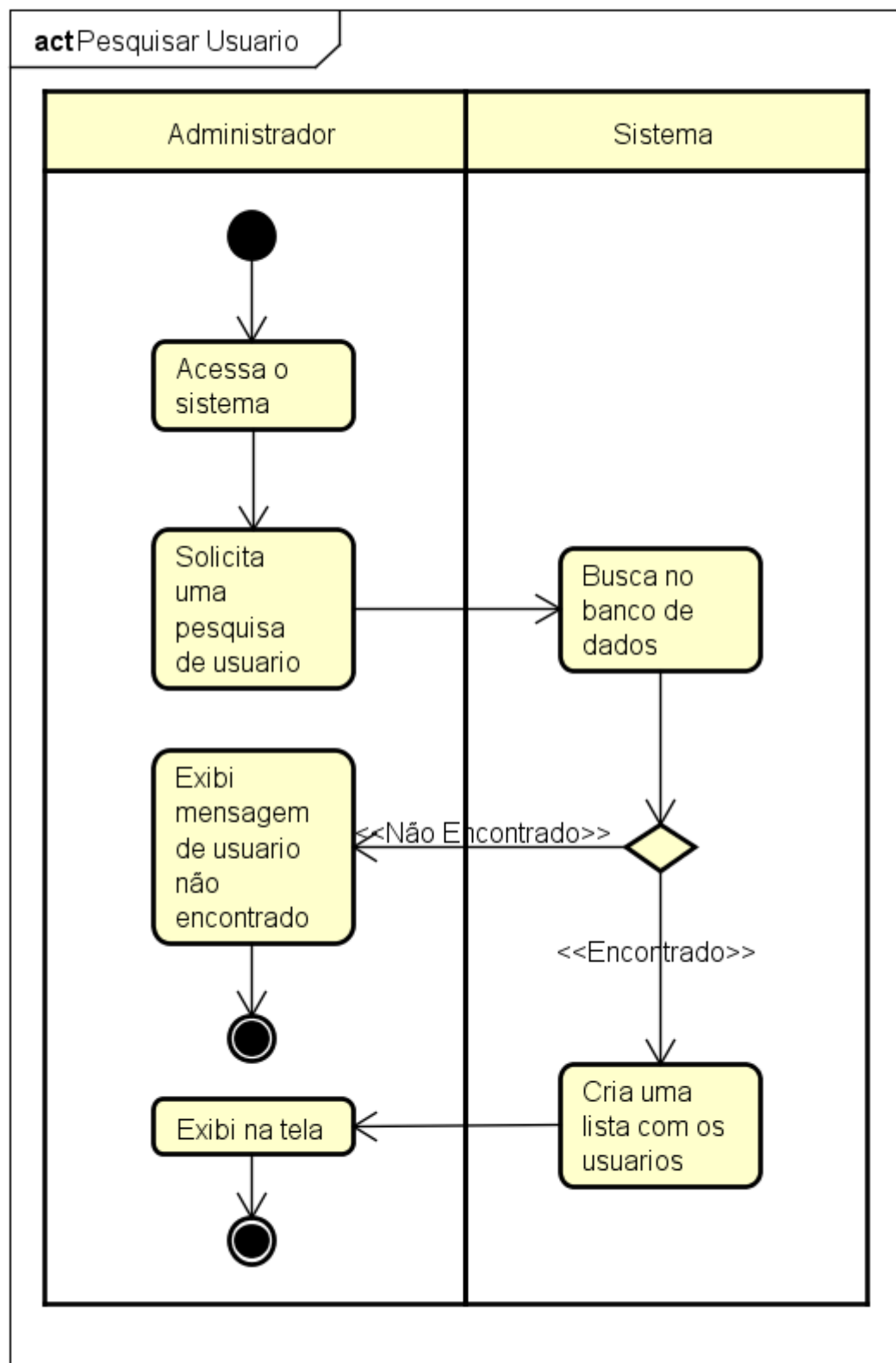
powered by Astah

Figura 5 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Usuário.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a alteração do cadastro de um usuário, onde é necessário realizar o acesso, selecionar a opção de alterar dados cadastrais, informar os novos dados e o sistema irá salvar os dados no banco de dados.

#### **Diagrama de atividade para Pesquisar Usuário**



powered by Astah

Figura 6 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Usuário.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a pesquisa de usuários do sistema, onde o administrador do sistema irá informar o nome do usuário e se o mesmo existir na base de dados o sistema retornará os dados na tela.

#### **Diagrama de atividade para Cadastrar Título**

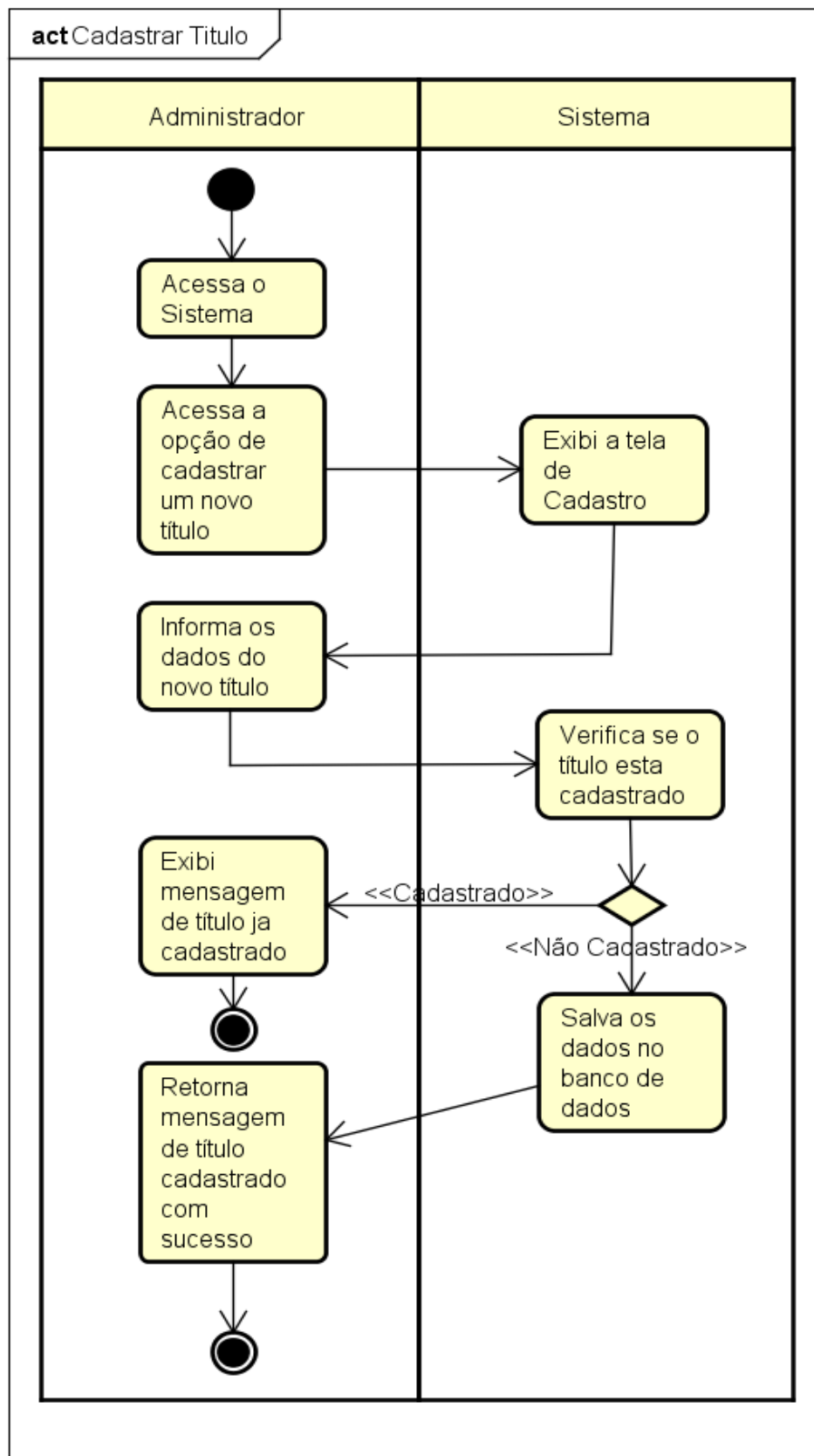
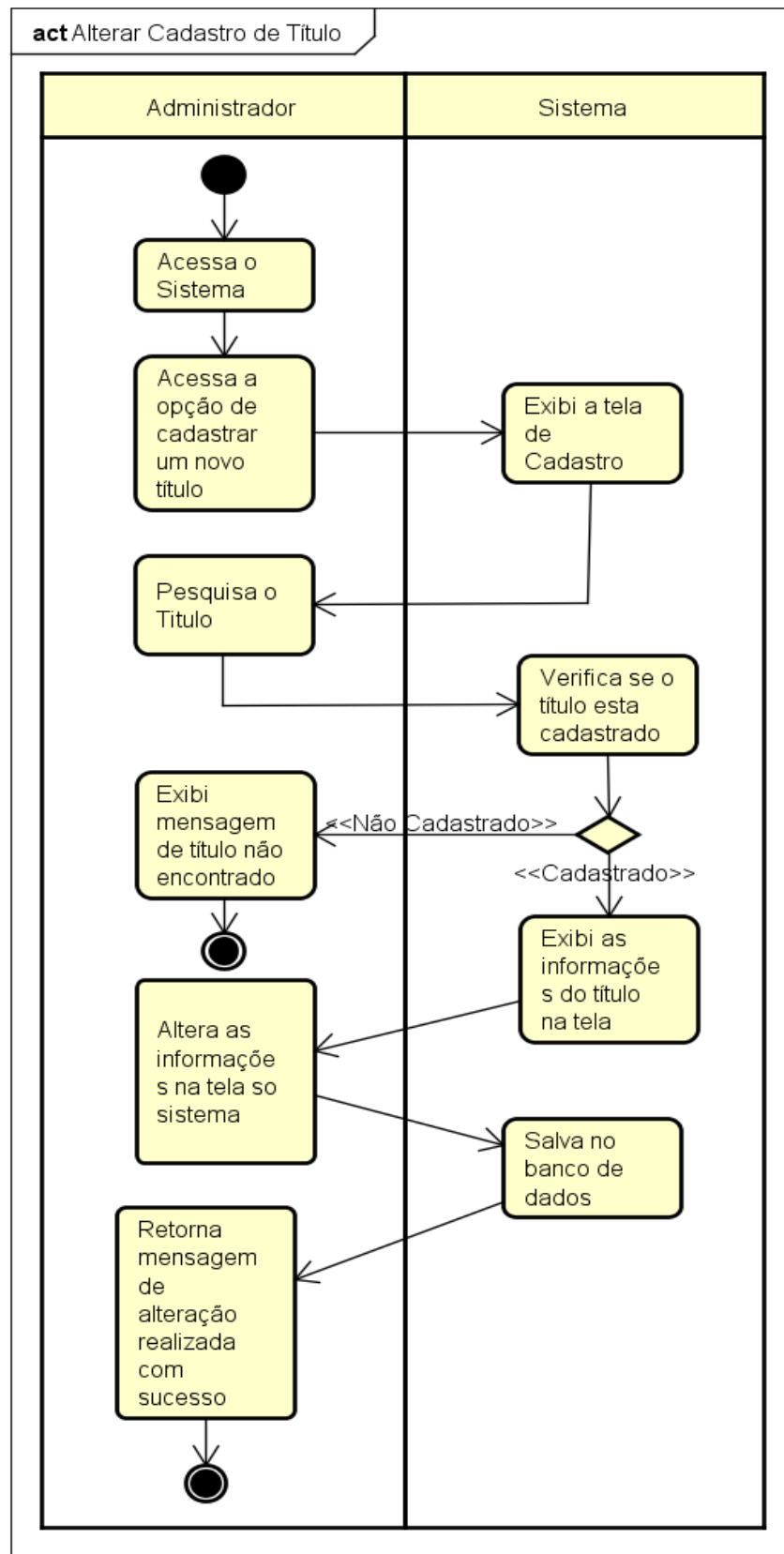


Figura 7 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Título.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante o cadastro de um novo título, onde é necessário realizar o acesso, e o administrador irá informar os dados do novo título e se o mesmo ainda não tiver sido cadastrado, os dados serão salvos na base de dados.

#### **Diagrama de atividade para Alterar Título**



powered by Astah

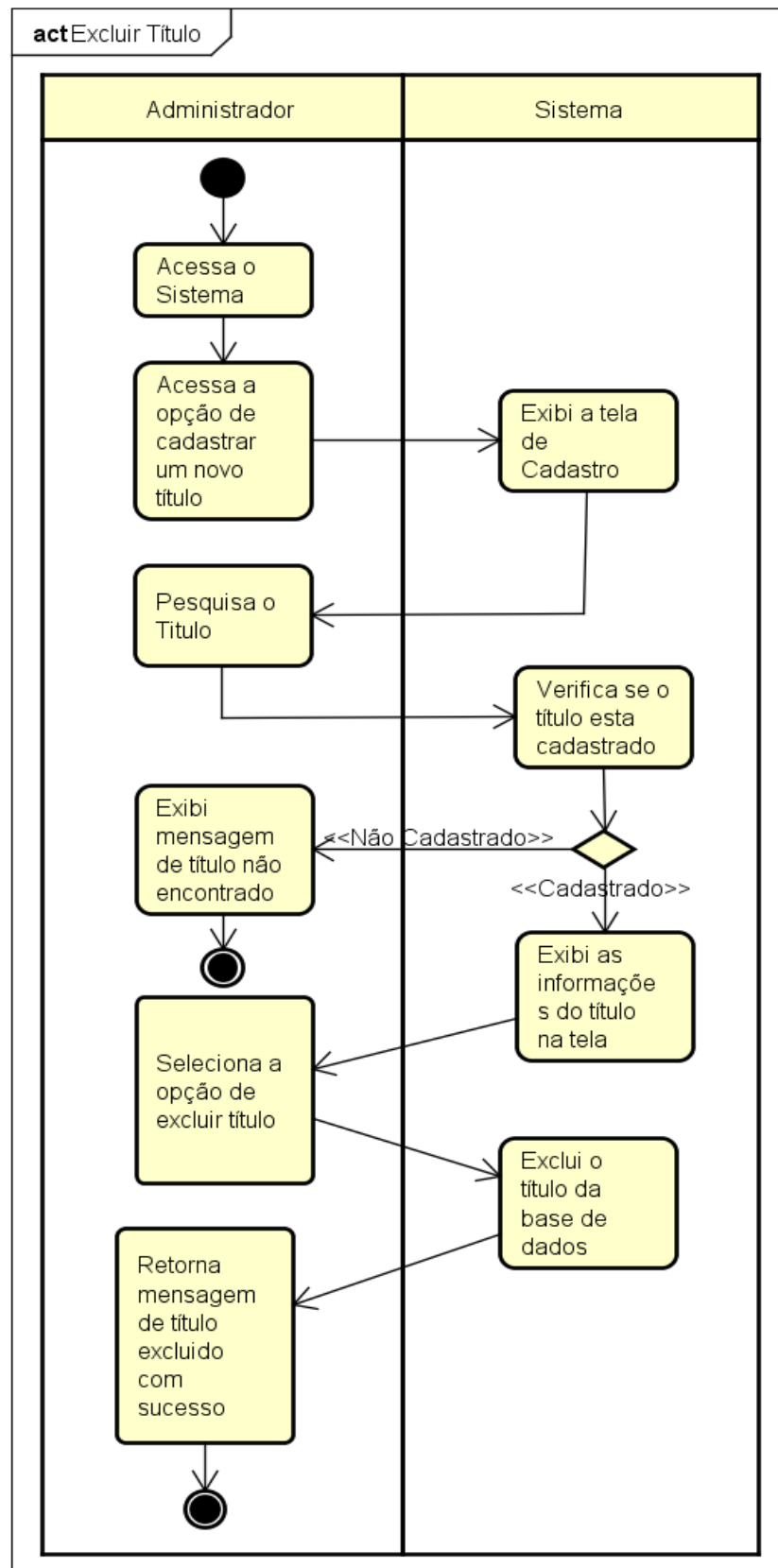
Figura 8 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Título.

(Fonte: Elaborada pelo Autor)



Este diagrama especifica como o sistema irá se comportar durante a alteração do cadastro de um título, onde é necessário realizar o acesso, pesquisar o título, informar os novos dados e o sistema irá salvar os dados no banco de dados.

#### **Diagrama de atividade para Excluir Título**



powered by Astah

Figura 9 – Diagrama de Atividade do Sistema de locação de AudioBooks: Excluir Título.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a exclusão do cadastro de um título, onde é necessário realizar o acesso, pesquisar o título, selecionar a opção excluir e o sistema irá apagar os dados do banco de dados.

#### **Diagrama de atividade para Pesquisar Título**

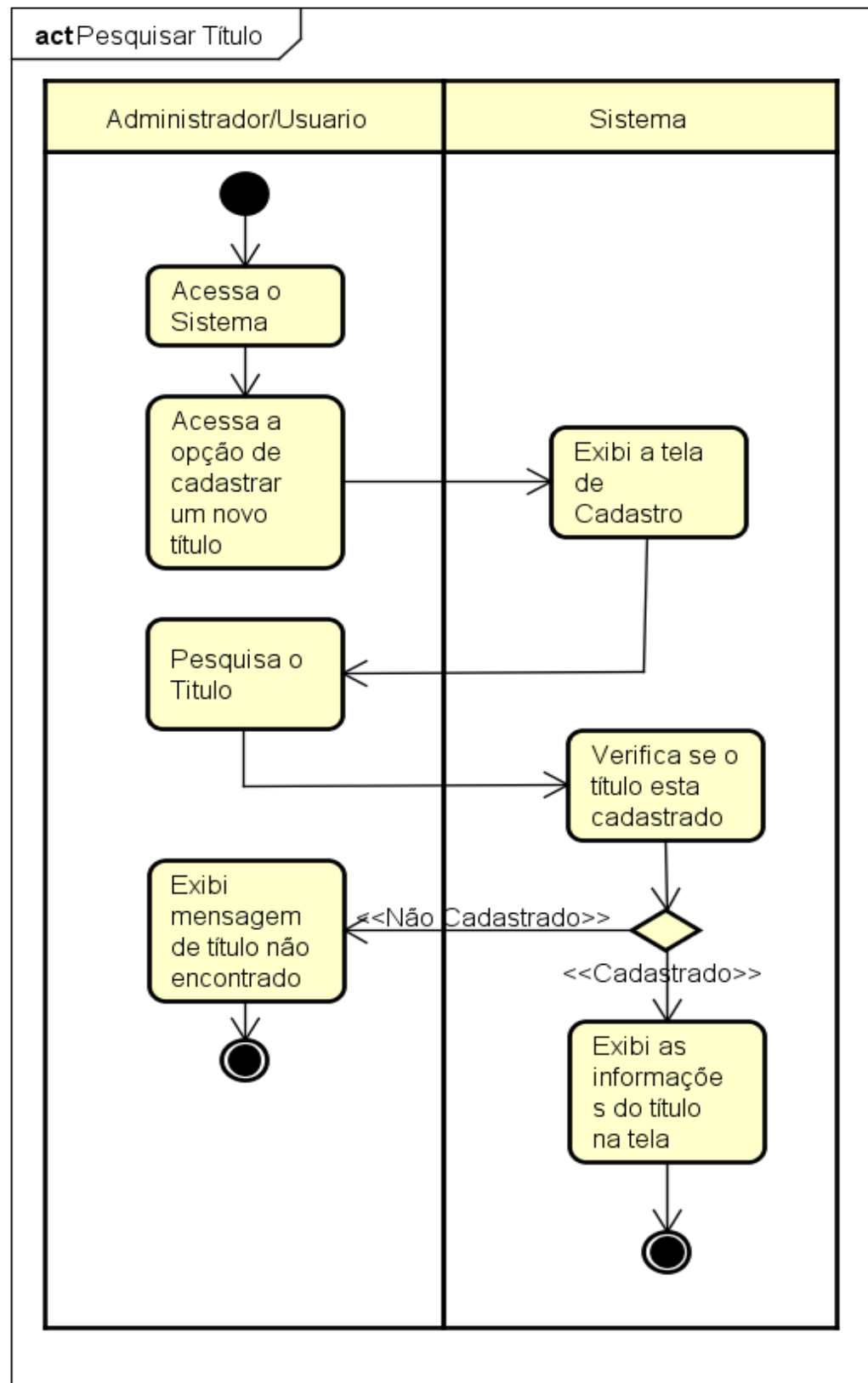
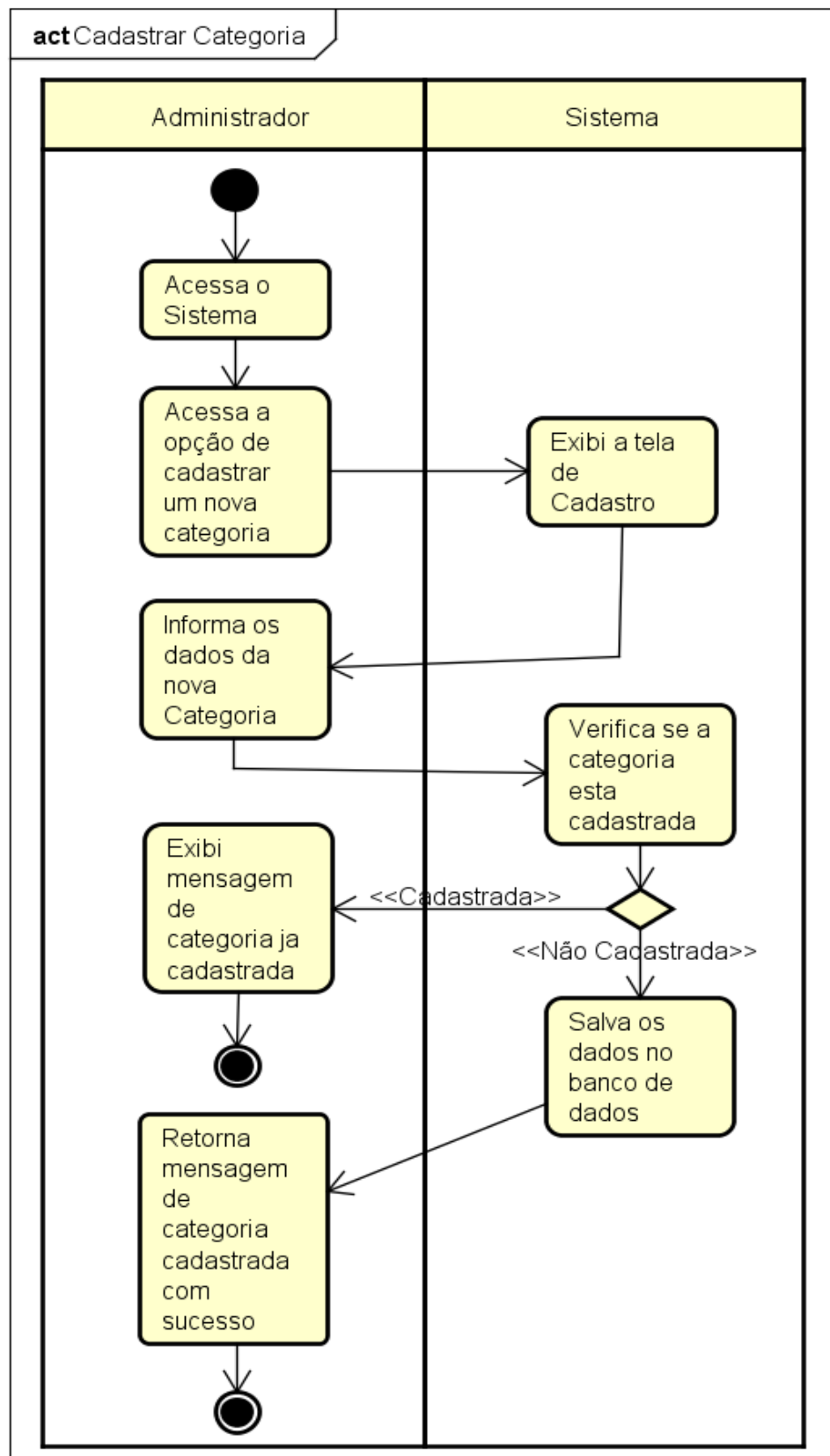


Figura 10 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Título.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a pesquisa de um título, onde é necessário realizar o acesso, pesquisar o título, e se o mesmo existir será exibido os dados na tela para o usuário.

#### **Diagrama de atividade para Cadastrar Categoria**



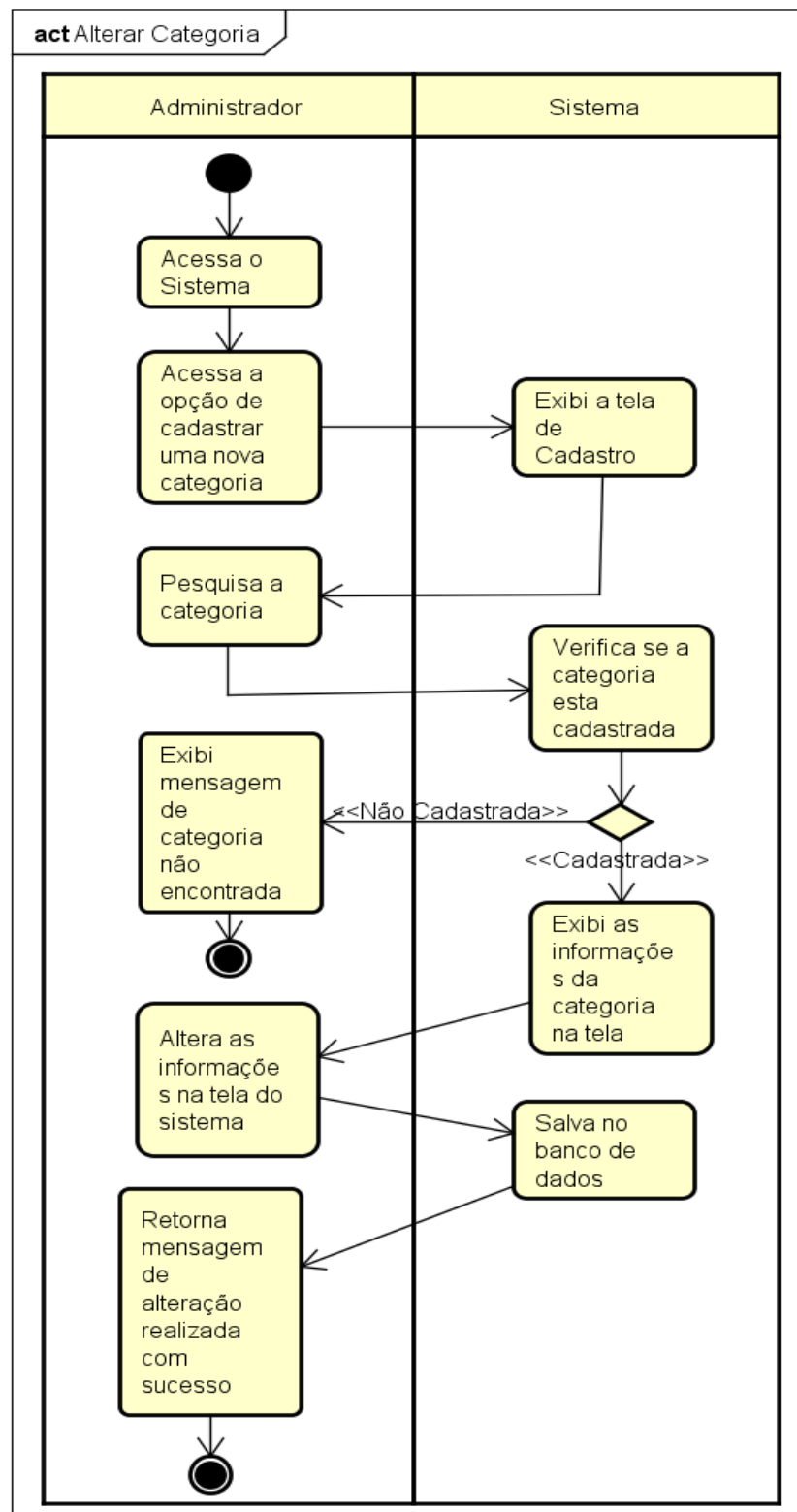
powered by Astah

Figura 11 – Diagrama de Atividade do Sistema de locação de AudioBooks: Cadastrar Categoria.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante o cadastro de uma nova categoria, onde é necessário realizar o acesso, e o administrador irá informar os dados da nova categoria e se a mesma ainda não tiver sido cadastrada, os dados serão salvos na base de dados.

#### **Diagrama de atividade para Alterar o Cadastro de uma Categoria**



powered by Astah

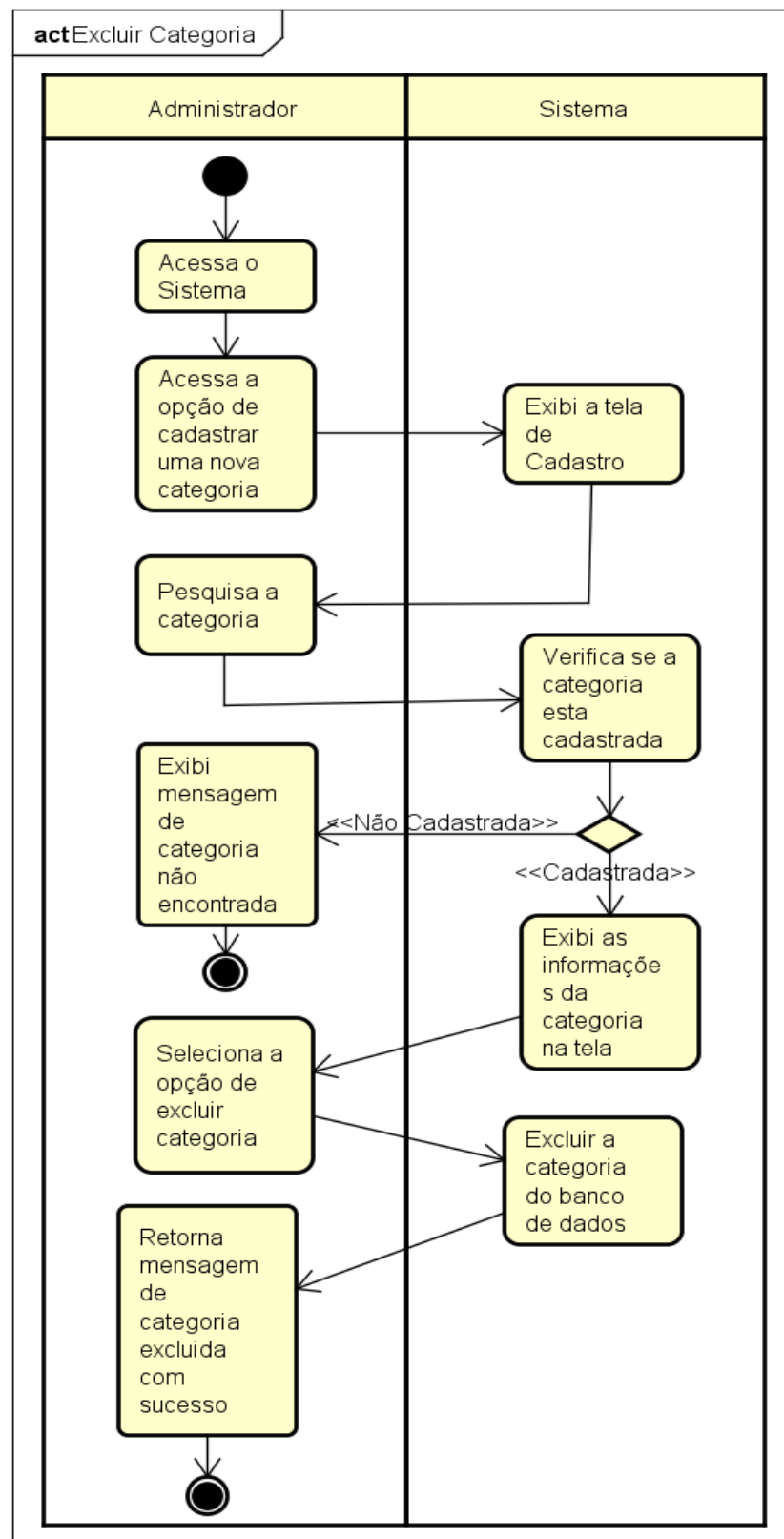
Figura 12 – Diagrama de Atividade do Sistema de locação de AudioBooks: Alterar Categoria.

(Fonte: Elaborada pelo Autor)



Este diagrama especifica como o sistema irá se comportar durante a alteração do cadastro de uma categoria, onde é necessário realizar o acesso, pesquisar a categoria desejada, informar os novos dados e o sistema irá salvar os dados no banco de dados.

#### **Diagrama de atividade para Excluir Categoria**



powered by Astah

Figura 13 – Diagrama de Atividade do Sistema de locação de AudioBooks: Excluir Categoria.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a exclusão do cadastro de uma categoria, onde é necessário realizar o acesso, pesquisar a categoria, selecionar a opção excluir e o sistema irá apagar os dados do banco de dados.

#### **Diagrama de atividade para Pesquisar Categoria**

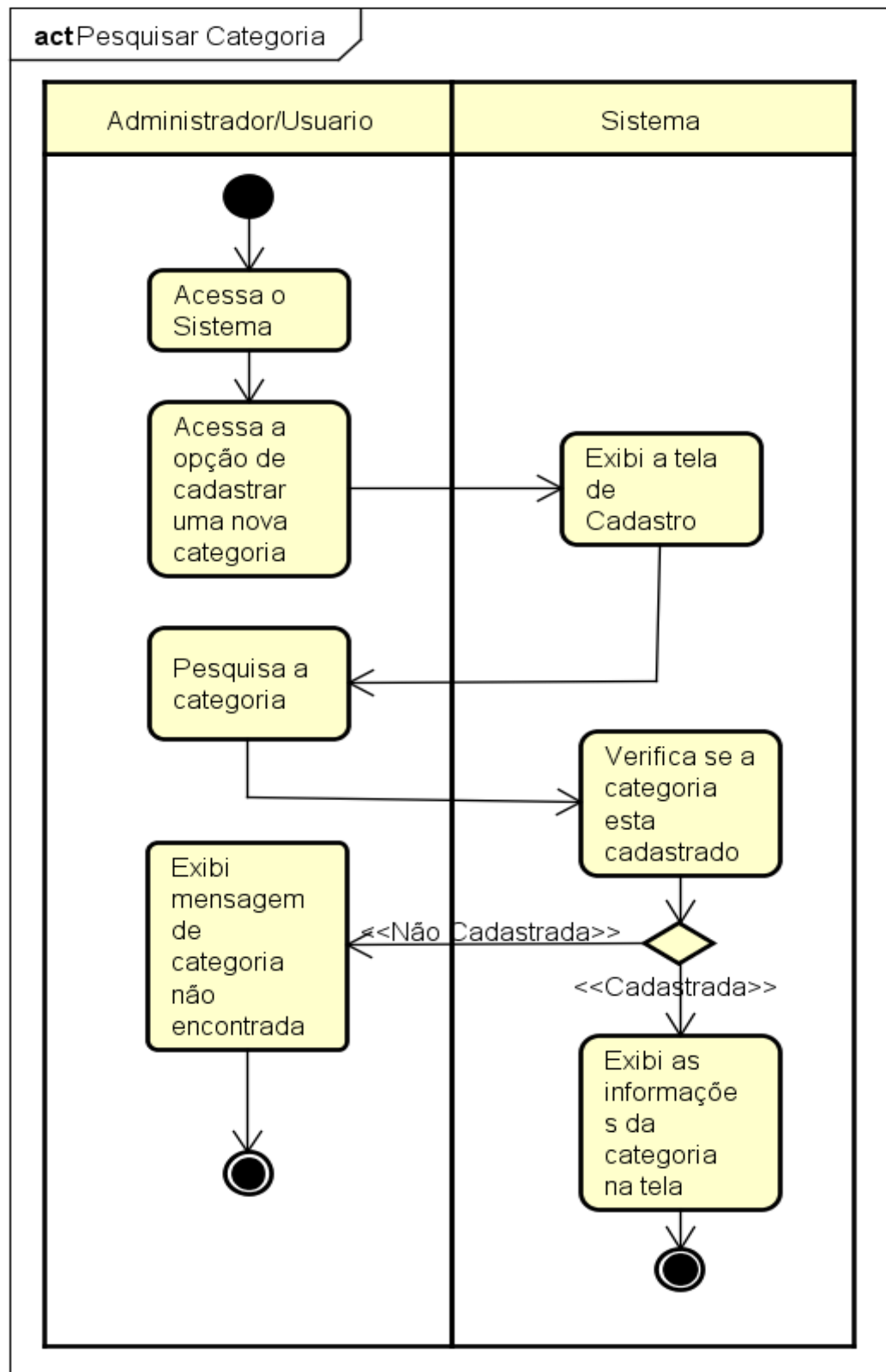
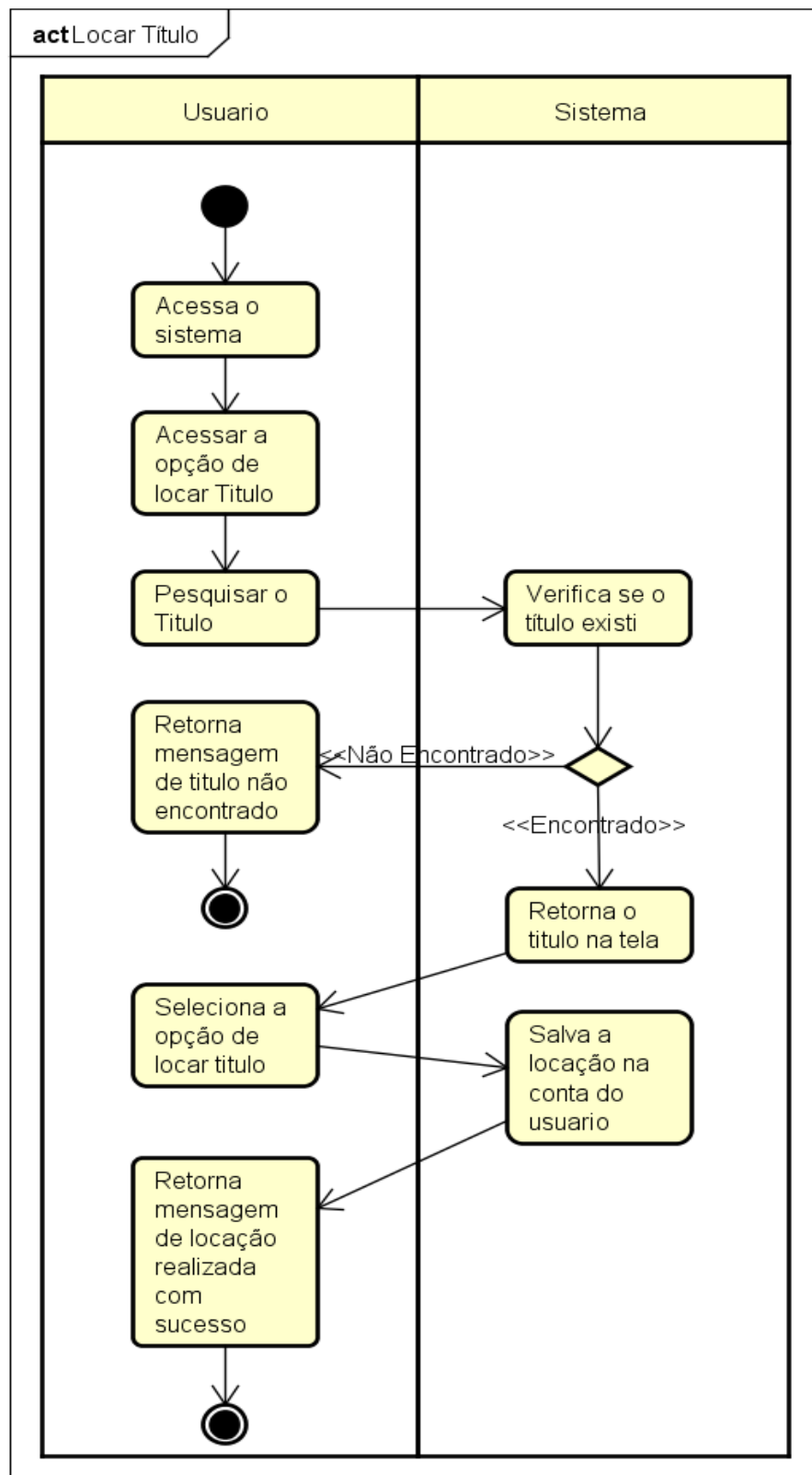


Figura 14 – Diagrama de Atividade do Sistema de locação de AudioBooks: Pesquisar Categoria.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a pesquisa de uma categoria, onde é necessário realizar o acesso, pesquisar a categoria, e se a mesma existir será exibido na tela os dados para o usuário.

#### **Diagrama de atividade para Locar Título**



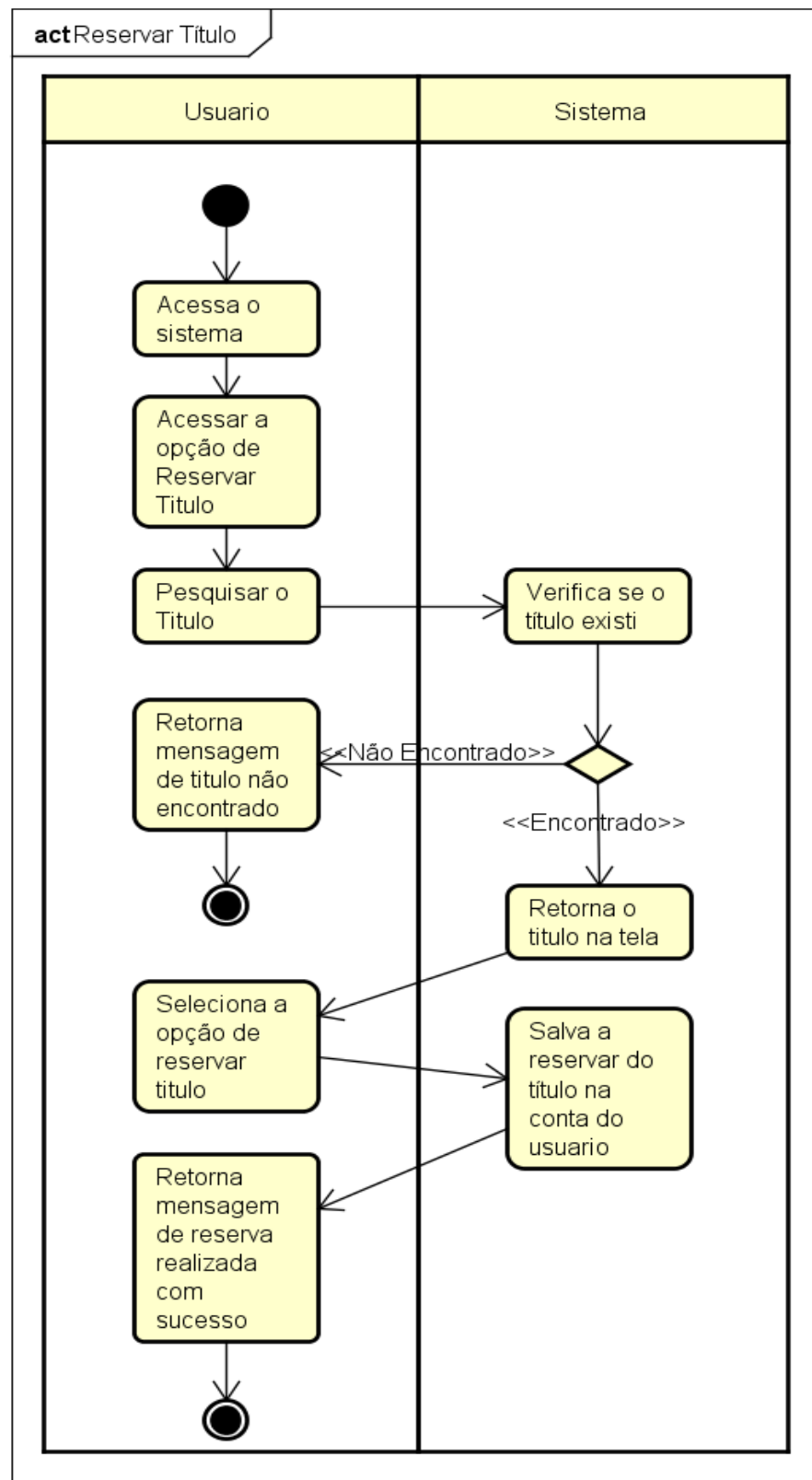
powered by Astah

Figura 15 – Diagrama de Atividade do Sistema de locação de AudioBooks: Locar Título.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a locação de um título, onde é necessário acessar o sistema, pesquisar o título e selecionar a opção locar, que o sistema irá salvar a locação na conta do usuário.

#### **Diagrama de atividade para Reservar Título**



powered by Astah

Figura 16 – Diagrama de Atividade do Sistema de locação de AudioBooks: Reservar Título.

(Fonte: Elaborada pelo Autor)



Este diagrama especifica como o sistema irá se comportar durante a reserva de um título, onde é necessário acessar o sistema, pesquisar o título e selecionar a opção reservar título, que o sistema irá salvar a reserva na conta do usuário.

#### **Diagrama de atividade para Solicitar Relatório**

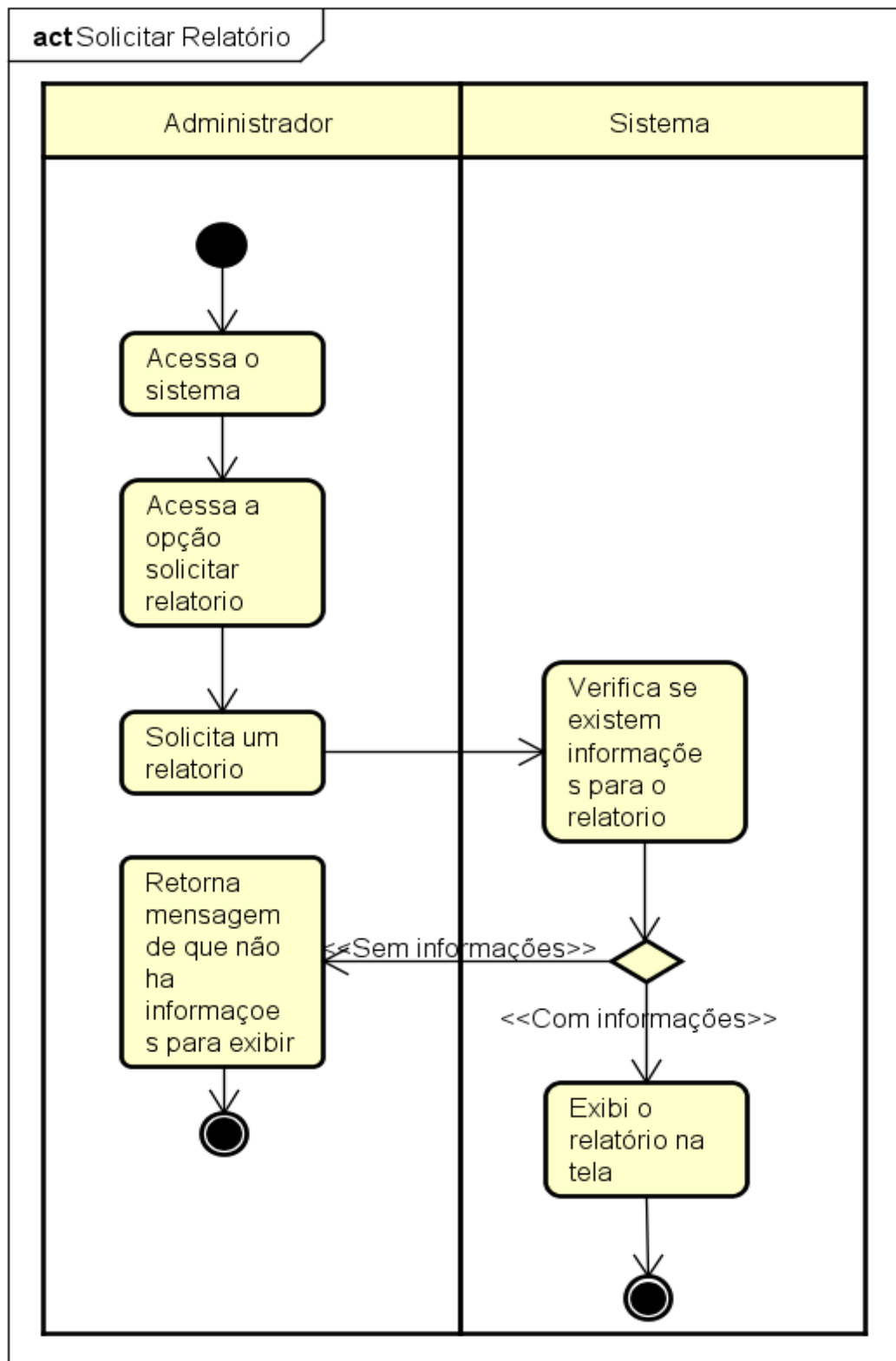


Figura 17 – Diagrama de Atividade do Sistema de locação de AudioBooks: Solicitar Relatório.

(Fonte: Elaborada pelo Autor)

Este diagrama especifica como o sistema irá se comportar durante a solicitação de um relatório feita pelo administrador do sistema, onde é necessário acessar o sistema, selecionar a opção de relatório desejada e se existir alguma informação para ser exibida será mostrada na tela para o usuário.

## 10 – Framework Laravel: Views, Controllers, Models e Rotas do Sistema

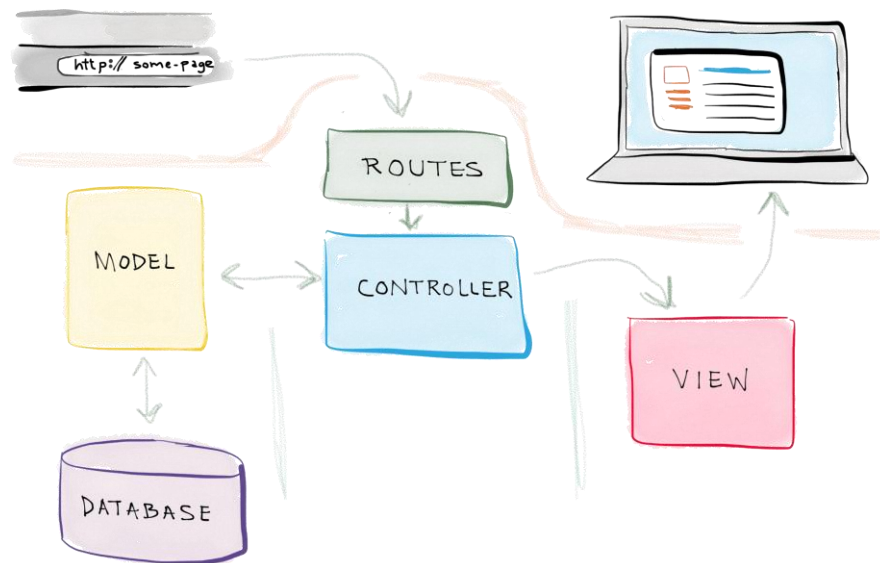


Figura 18: Diagrama MVC Laravel.

Fonte: < <https://auth0.com/>>. Acessado em: 12 de Outubro de 2017.

### Rotas/Routes:

Rotas são os caminhos da sua aplicação web que direcionam as requisições do usuário para o Controlador Apropriado:

Mecanismo de Rotas:

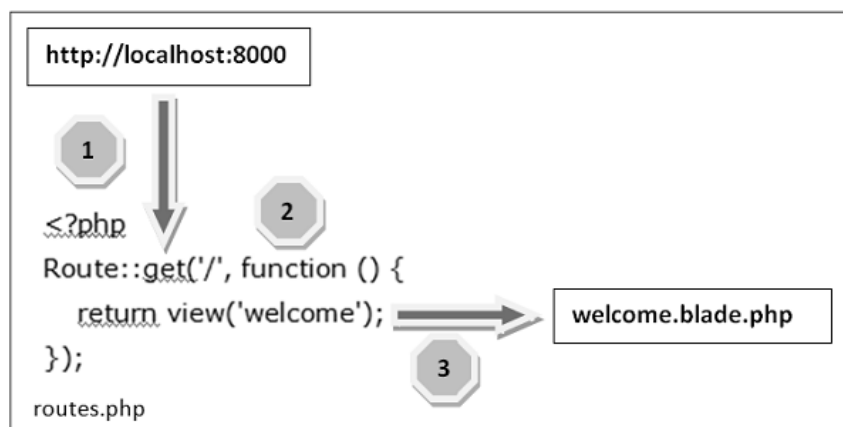


Figura 19: Mecanismo de Rotas Laravel.

Fonte: < [https://www.tutorialspoint.com/laravel/laravel\\_routing.htm/](https://www.tutorialspoint.com/laravel/laravel_routing.htm/)>.

Ex: Rota [HTTP://localhost:8000/](http://localhost:8000/), do tipo **get** chama uma função que retorna para o usuário uma *view*: `Welcome.blade.php`

### **Controladores/Controllers:**

Controladores agem direcionando o trafego entre as *Views* e os *Models*:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class inicialController extends Controller
{
    public function show(){
        return view('welcome');
    }
}
```

Figura 20 – Exemplo de *Controllers* Laravel.

(Fonte: Elaborada pelo Autor)

Ex: Controller `inicialController` chama função `show()` que retorna para usuário uma *view*.

### **Modelo/Model:**

Cada Tabela do banco de dados tem um *Model* correspondente, que é usada para interagir com a mesma.

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class titulos extends Model
{
    //tabela associada a esse model
    protected $table = 'titulo';
    //chave primaria da tabela
    protected $primaryKey = 'id_titulo';
}
```

Figura 21 – Exemplo de *Model* Laravel.

(Fonte: Elaborada pelo Autor)

Ex: Model `titulos` representa a tabela `titulo` no banco de dados, e permite operações de Insert, Select, Delete, Update etc.

**Tela/View:**

Separa a logica da aplicação e a logica de apresentação da pagina.

```
<html>
  <body>
    <h1>Hello, World</h1>
  </body>
</html>
```

Figura 22 – Exemplo de *View* Laravel.

(Fonte: Elaborada pelo Autor)

Ex: No Laravel *views* são arquivos .php com tags html que formalizam a visualização das informações.

**Títulos:****Cadastro**

- Coleta de dados →

Para que títulos sejam cadastrados o ADM deve acessar a rota /cadastrodetitulos que retornará uma *view* contendo os formulários necessários para a coleta dos dados do título.

- Tratamento dos dados pelo controlador →

Após a coleta dos dados um Botão chama o controlador `titulo_Controller` que através da função `cadastrar` repassa as informações para a *model* `títulos`.

- Inserção dos dados →

Model `títulos` Insere as novas informações no banco de dados.

## Exclusão

- Coleta de dados →

Para que títulos sejam excluídos o ADM deve acessar a rota `/exclusaodetitulos` que retornará uma *view* contendo uma lista com todos os títulos atualmente cadastrados no sistema.

- Tratamento dos dados pelo controlador →

Para cada título cadastrado existirá um botão associado a ele o qual chama o controlador `título_Controller` que através da função `excluir` repassa a id do título para a *model* `títulos`.

- Exclusão dos dados→

*Model* `títulos` exclui o título associado ao id informado no banco de dados.

## Locadores:

### Cadastro:

- Coleta de dados →

Para que um locador seja cadastrado o USUARIO deve acessar a rota `/register` que retornará uma *view* contendo os formulários necessários para a coleta dos dados do usuário.

- Tratamento dos dados pelo controlador →

Após a coleta dos dados um Botão chama o controlador `RegisterController` que através da função `validador` valida as informações, e através da função `create` repassa a informações para a *model* `User`

- Inserção dos dados→

*Model* `User` Insere as novas informações no banco de dados.

### Alterar Cadastro:

- Coleta de dados →

Para que um locador possa alterar seu cadastro o USUARIO deve acessar a rota `/minhaconta` que retornará uma *view* contendo os formulários necessários para a coleta dos dados do usuário.

- Tratamento dos dados pelo controlador →

Após a coleta dos dados um Botão chama o controlador `RegisterController` que através da função `atualisar` repassa a informações para a *model* `Usuario`

- Inserção dos dados→

*Model* `Usuario` atualiza as novas informações no banco de dados.

## Locação:

### Cadastro:

- Coleta de dados →

Para que uma locação seja cadastrada o USUARIO deve pesquisar um titulo por nome ou categoria através dos mecanismos de pesquisa contidos na *view*: `home.php`

A busca é feita com o controlador `titulo_Controller` através da função `consulta` que retorna todos os títulos correspondentes à pesquisa.

- Tratamento dos dados pelo controlador →

Após a coleta dos dados cada titulo conterà um botão associado que chama o controlador `locação_Controller` que através da função `locar` repassa o id do titulo e id do usuário para a *model* `locação`.

- Inserção dos dados→

*Model* `locação` Insere as novas informações no banco de dados.



## Reserva:

### Cadastro:

- Coleta de dados →

Para que uma reserva seja cadastrada o USUARIO deve pesquisar um titulo através da categoria Lançamentos contido na view: home.php

A busca é feita com o controlador `titulo_Controller` através da função `consulta` que retorna todos os títulos correspondentes à pesquisa.

- Tratamento dos dados pelo controlador →

Após a coleta dos dados cada titulo conterà um botão associado que chama o controlador `locação_Controller` que através da função `reservar` repassa o id do titulo e id do usuário para a model `reserva`.

- Inserção dos dados→

Model `reserva` Insere as novas informações no banco de dados.

## 11 – Banco de Dados do Sistema

No desenvolvimento do sistema WEB para locação de AudioBooks será utilizada uma conexão com banco de dados MySQL para armazenar e consultar informações. Na figura a seguir é mostrado a estrutura e relacionamentos do banco de dados usado pela aplicação.

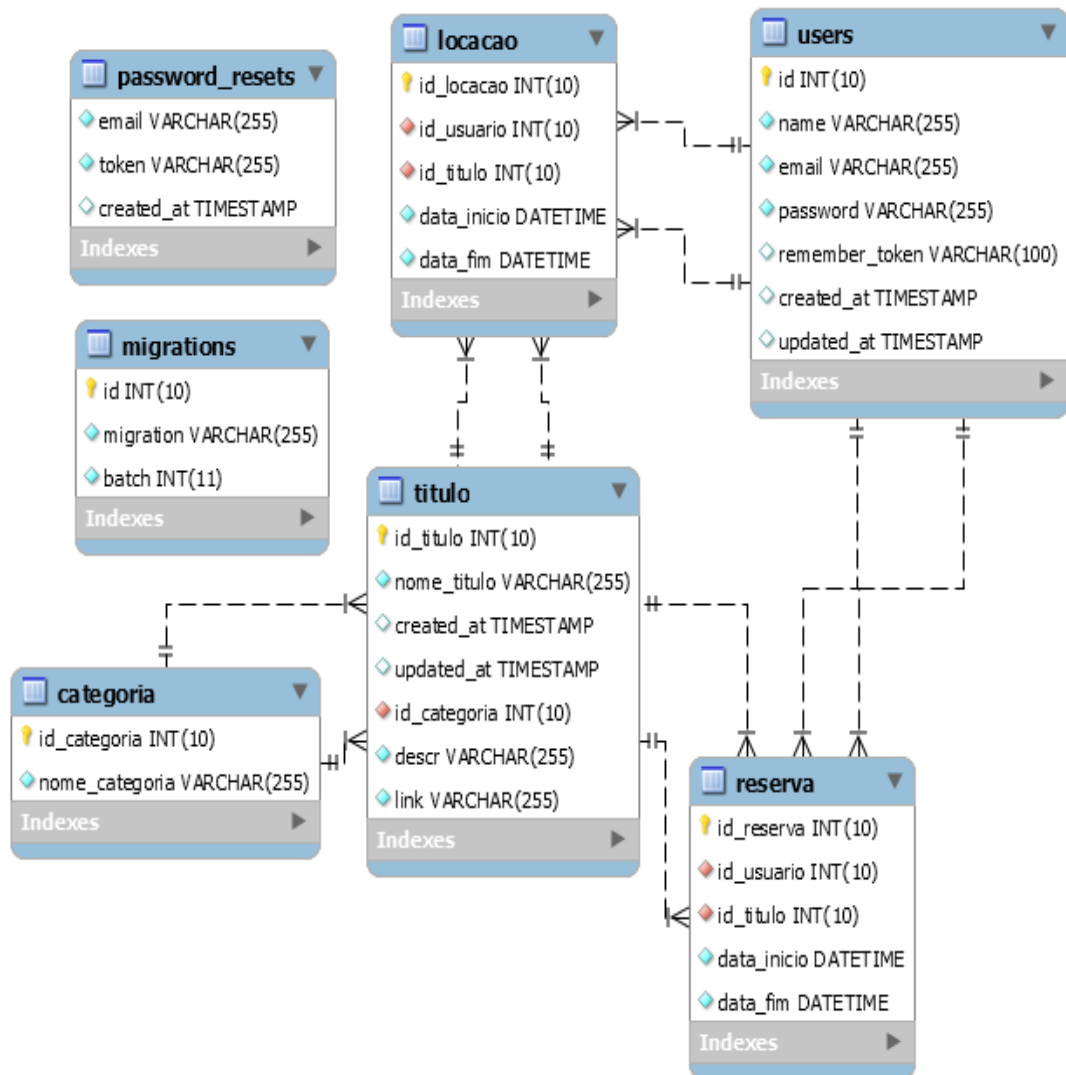


Figura 23 – Banco de dados do Sistema.

(Fonte: Elaborada pelo Autor).

## 12 – Estimativa de Custo do Projeto

Portanto analisando o custo para desenvolver este projeto de software, ficou especificado na tabela a seguir o valor total para implementar o mesmo tomando como base o valor da hora de trabalho do profissional qualificado.

<b>ESTIMATIVA DE CUSTO DO PROJETO DO SISTEMA DE LOCAÇÃO DE AUDIOBOOKS</b>				
<b>VALOR/HORA</b>	<b>R\$ 50,00</b>			
<b>SERVIÇO</b>	<b>DIA S</b>	<b>HORAS/DIA</b>	<b>TOTAL DE HORAS</b>	<b>TOTAL</b>
<i><b>Pesquisas de ferramentas</b></i>	15	2	30	R\$ 1.500,00
<i><b>Aprendizagem das ferramentas escolhidas</b></i>	20	3	60	R\$ 3.000,00
<i><b>Desenvolvimento da documentação</b></i>	15	4	60	R\$ 3.000,00
<i><b>Desenvolvimento de diagramas(parte técnica)</b></i>	10	3	30	R\$ 1.500,00
<i><b>Material necessário(computador)</b></i>				R\$ 4.000,00
<i><b>Desenvolvimento</b></i>	60	6	360	R\$ 18.000,00
				R\$ -
<b>CUSTO TOTAL ESTIMADO</b>				R\$ 31.000,00

Tabela 1 – Estimativa do Custo do Projeto do sistema de locação de AudioBooks.

(Fonte: Elaborada pelo Autor)

### **13 - Considerações**

Este documento contém a análise de requisitos, e análise orientada a Objeto sobre o projeto de um software para um sistema WEB de locação de AudioBooks. Ele permite uma melhor comunicação entre o grupo de desenvolvimento e o cliente, principalmente quando se fala sobre a especificação dos requisitos, que são as funcionalidades do software.

Neste documento estão descritos os eventos, ações e os casos de uso, que são mecanismos que irão demonstrar de forma mais detalhada não só para o cliente, mas também para a equipe de desenvolvimento, como o software irá se comportar de acordo com aquilo que foi solicitado pelo próprio cliente. Isto acaba sendo fundamental, pois desta forma ambas as partes estarão mais integradas dentro do processo.

Tendo esse ponto de vista, este documento se torna fundamental não somente durante o processo de planejamento do software, mas também durante o desenvolvimento da aplicação.

## 14 – Referências

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Disponível em: <<https://books.google.com.br/books?hl=ptR&lr=&id=elvjBwAAQBAJ&oi=fnd&pg=PT4&dq=analise+de+requisitos&ots=xTXhPTsOAA&sig=cAlrM1pycHUIOVdfIKlg18eAwBw#v=onepage&q&f=false>>. Acessado em: 15 de Outubro de 2017.

FOWLER, Martin. **UML essencial: uma breve guia para linguagem padrão**. Disponível em: <[https://books.google.com.br/books?hl=ptR&lr=&id=xxoXcuh0oS0C&oi=fnd&pg=PR8&dq=uml+unified+modeling+language&ots=us7H9lql6q&sig=DdDmmswR\\_\\_mOFyILwTsvwiPVGhU#v=onepage&q=uml%20unified%20modeling%20language&f=false](https://books.google.com.br/books?hl=ptR&lr=&id=xxoXcuh0oS0C&oi=fnd&pg=PR8&dq=uml+unified+modeling+language&ots=us7H9lql6q&sig=DdDmmswR__mOFyILwTsvwiPVGhU#v=onepage&q=uml%20unified%20modeling%20language&f=false)>. Acessado em: 15 de Outubro de 2017.

GUEDES, Gilleanes T. A.. **UML 2 – Guia Prático – Segunda Edição**. Disponível em: <<https://books.google.com.br/books?hl=ptR&lr=&id=YEaGDQAAQBAJ&oi=fnd&pg=PA15&dq=diagrama+de+atividade&ots=ragnDFCqMf&sig=epplxQOauFgPFsdE0jAljTpUs#v=onepage&q=diagrama%20de%20atividade&f=false>>. Acessado em: 15 de Outubro de 2017.

GUEDES, Gilleanes T. A. . **UML Uma Abordagem Prática**. 2. Ed. São Paulo: Novatec Editora LTDA. 2006.

RIBEIRO, Antônio Mendes. **Glossário UML**. UFMG. Disponível em: <[http://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/interacao/diagrama\\_de\\_interacao.htm](http://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/interacao/diagrama_de_interacao.htm)>. Acesso em: 15 de Outubro de 2017.

DOOLEY, John. **Introdução ao Padrão MVC**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Acesso em: 15 de Outubro de 2017.

Laravel. **Laravel Documentação**. Disponível em: <<https://laravel.com/docs/5.5>>. Acesso em: 15 de Outubro de 2017.

Tutorialspoint. **Laravel - Routing**. Disponível em: <[https://www.tutorialspoint.com/laravel/laravel\\_routing.htm](https://www.tutorialspoint.com/laravel/laravel_routing.htm)>. Acesso em: 15 de Outubro de 2017.

auth0.com. **Creating your first Laravel app and adding authentication.** Disponível em: <  
<https://auth0.com/blog/creating-your-first-laravel-app-and-adding-authentication/>>. Acesso em: 10 de  
Outubro de 2017.