



EASY LIBRARY CENTER: SISTEMA DE GESTÃO PARA BIBLIOTECAS

Mateus Antônio Huebra Henrique
Ezequias Ferreira de Souza

Curso: Tecnologia em Análise e Desenvolvimento de Sistemas Período: 6º
Área de Pesquisa: Ciências Exatas e da Terra

Resumo: O gerenciamento de arquivos e informações de forma física em papel é um método ultrapassado. A aplicação desenvolvida tem como objetivo automatizar e facilitar o gerenciamento de todos os dados mantidos por uma biblioteca, além torná-la mais atrativa em meio a modernidade, disponibilizando uma página pública para checar a existência e disponibilidade dos livros e exemplares. A aplicação foi desenvolvida utilizando um *framework* simples e autoral em linguagem PHP (Pré-processador de Hipertexto), no servidor, e o *framework* Quasar, que utiliza *Javascript*, no cliente. Além do uso do MySQL como sistema gerenciador de banco de dados. Também foram utilizados diagramas da UML (Linguagem de Modelagem Unificada) para o projeto. O Sistema inclui tabelas para o cadastramento e controle de livros, exemplares, empréstimos, categorias, autores, leitores e editoras, além de um módulo para a visualização de algumas estatísticas.

Palavras-chave: Bibliotecas. Gerenciamento. Sistema.

1. INTRODUÇÃO

Guardar todos os registros e documentos de uma organização em arquivos físicos, nos dias de hoje, é uma escolha um tanto quanto ilógica. Escolas e faculdades, há poucos anos atrás, ocupavam salas inteiras com pastas e fichas com o histórico dos alunos. Bastava um incêndio, e toda a informação seria perdida para sempre. Qualquer necessidade de consulta tomava um tempo e esforço inconveniente para ser realizada. Para construir relatórios estatísticos com base em todos os dados armazenados então, era praticamente inviável.

Esse é um dos problemas que afetam, também, as bibliotecas do tempo contemporâneo. Outro problema vem do fato de que ir aos centros bibliotecários têm se tornado cada vez mais uma escolha desinteressante e fora de moda para as pessoas atualmente. Se locomover de sua casa até a biblioteca sem saber se eles possuem o livro que você necessita, ou sequer saber se há um exemplar disponível, é um risco de perda de tempo que, para a maioria das pessoas, não vale a pena correr.

Com base nisso, chegamos à questão: como automatizar o gerenciamento das informações mantidas por uma biblioteca, além de torná-la mais atrativa em meio a modernidade?

Nos dias de hoje, com a ajuda dos sistemas de tecnologia da informação (TI), buscas em bancos de dados imensos levam questão de segundos, ou até menos. Diferentes tipos de relatórios podem ser gerados automaticamente, sem esforço algum do usuário. E o acesso à Internet já está amplamente mais democratizado do que há algumas décadas atrás, com a maior parte da população brasileira atual já tendo acesso a ela.

O Sistema será a ferramenta utilizada pelos funcionários para automatizar o controle de todos os livros, exemplares, leitores e empréstimos nos arquivos da

biblioteca. Isso facilitará tanto as buscas pela disponibilidade de exemplares quanto as consultas no histórico de empréstimos de alguém. Além de tornar extremamente fácil a coleta de dados para relatórios, para saber, por exemplo, qual a categoria de livros mais locados em determinado mês. O Sistema também contará com uma parte aberta ao público, onde será possível pesquisar, pela Internet, a existência de um livro ou a disponibilidade de um exemplar do mesmo, buscando resolver todos os problemas citados.

2. DESENVOLVIMENTO

2.1. Referencial Teórico

Vivemos em uma época em que o uso da tecnologia e dos sistemas de TI, nas organizações e na vida das pessoas, se torna cada vez mais presente e em constante evolução. E é necessário se adaptar a isso.

Afinal, é inegável a facilidade que os sistemas de informação trazem para os processos, tanto organizacionais quanto cotidianos. Com poucos cliques você solicita a um dispositivo computacional a recuperação e organização de grandes quantidades de informação e a resolução de cálculos importantes, com uma taxa de erro mínima (BATISTA, 2017). Isso traz agilidade e precisão que nenhum ser humano seria capaz de alcançar manualmente. Com as bibliotecas, que são o alvo deste Sistema, não é diferente.

Nas bibliotecas, a automação surge para facilitar, uniformizar e reduzir o tempo de trabalho, atender melhor as necessidades de seus usuários, gerando um grande avanço neste campo. A utilização de softwares especializados para gestão de centros de informação foi o que possibilitou esse avanço (RODRIGUES; PRUDÊNCIO, 2019, p.1).

A automação traz uma enorme facilidade para os processos de qualquer organização e não é de hoje que se vê a necessidade de uma integração entre bibliotecas tradicionais e digitais. Há mais de 20 anos Cunha (1999, p.267) já havia apontado que tal conciliação seria o futuro:

Biblioteca digital tem sido um tema cada vez mais discutido nas literaturas de ciência da informação e de informática [...]. Esses projetos de bibliotecas digitais entrarão na fase operacional nos próximos anos, portanto, dentro em breve, entraremos em uma fase híbrida, convivendo com a biblioteca tradicional que conhecemos e com a futurística biblioteca digital.

Essas automações facilitam o uso até mesmo para os frequentadores da biblioteca, que podem realizar consultas sem sair de casa, como Morigi e Souto (2006, p.3) já afirmavam:

Com sistemas informatizados implantados nas unidades de informação as interações entre bibliotecários e usuários também passam por mudanças significativas. As consultas a um bibliotecário ou a um catálogo automatizado realizadas através da Internet subsistem com as consultas presenciais.

2.2. Desenvolvimento do Sistema

O Sistema foi desenvolvido em dois módulos separados: servidor e cliente. Deste modo, é possível criar diferentes aplicações clientes para um mesmo servidor, que funciona como uma API (*Application Programming Interface*). Segundo Carneiro dos Santos (2013, p.99), “API é o conjunto de rotinas, padrões e instruções de programação que permite que os desenvolvedores criem aplicações que possam acessar determinado serviço na internet”. Isso é útil, por exemplo, no caso do proprietário da organização desejar, mais futuramente, desenvolver um cliente *mobile*, isto é, um aplicativo para os dispositivos móveis.

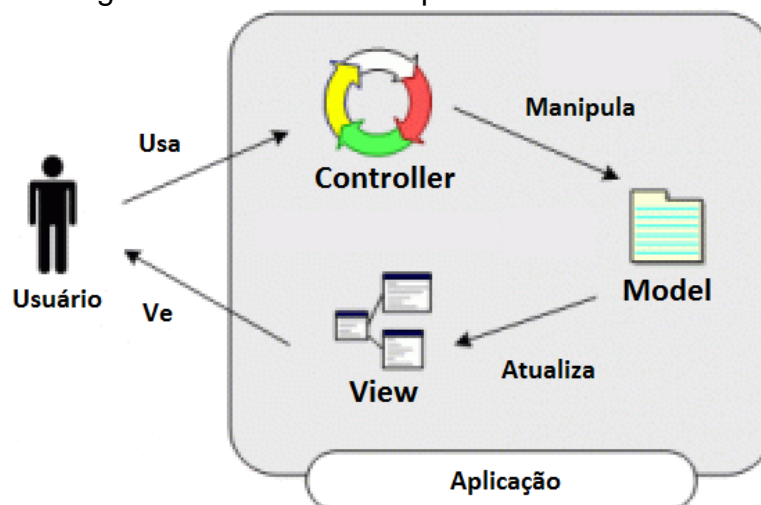
2.2.1. Servidor (PHP)

O *backend* utiliza um *framework* simples que foi programado especificamente para a construção deste Sistema, porém podendo ser reutilizado para projetos futuros. Nas palavras de Gamma (2009, p. 332) *Framework* é “um conjunto de classes que cooperam entre si provendo assim um projeto reutilizável para um domínio específico de classes de sistema”. Sendo assim, eles podem ser usados como uma ferramenta, para facilitar a programação de qualquer sistema na linguagem desenvolvida.

O *framework* em questão utiliza PHP, em sua versão 7, uma das mais conhecidas linguagens para desenvolver aplicações *Web*. E foi projetado utilizando o padrão MVC (*model, view, controller*) que, segundo Orlando (2009, p. 48), “é um padrão arquitetural cujo objetivo é isolar lógica de negócio, entrada e apresentação, permitindo assim independência no desenvolvimento”. Ou seja, o padrão MVC divide a aplicação em três módulos: *model, view* e *controller*.

O módulo *model* contém as classes que devem ser usadas para manipular o banco de dados. O módulo *view* gera, de forma visual, uma interface para a exibição do estado do aplicativo para o usuário. E, por fim, o módulo *controller* processa as requisições do usuário, manipulando as classes de *model* e, então, retornando os dados para a camada *view*.

Figura 1 – Modelo da arquitetura MVC.



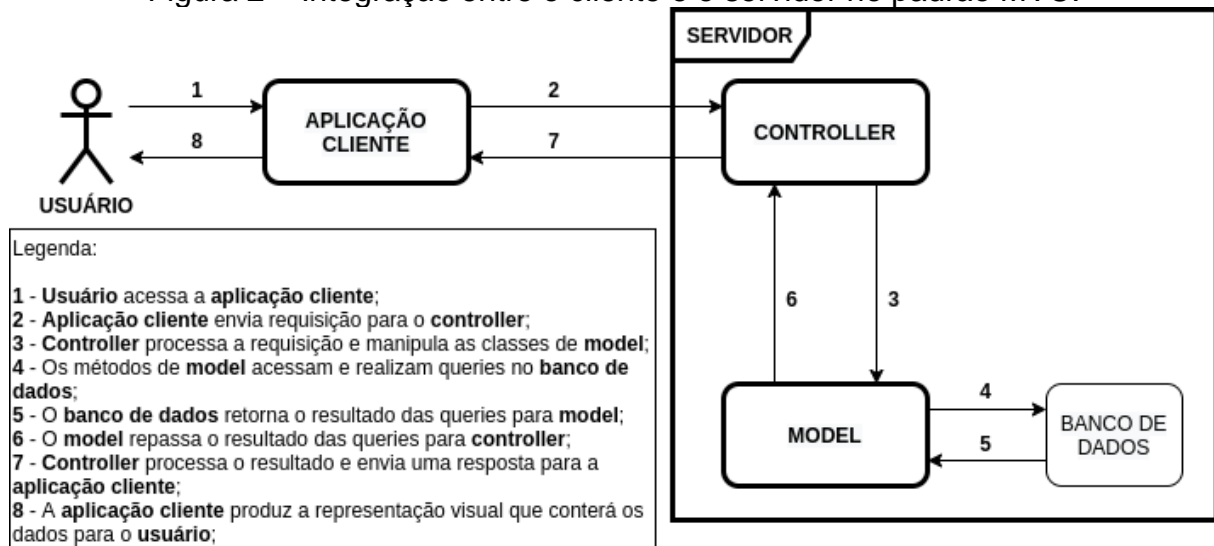
Fonte: (ORLANDO, 2009).

2.2.2. Cliente (Quasar)

O *frontend* foi desenvolvido com o uso do *Quasar Framework*. Quasar é um *framework* de desenvolvimento *Web* baseado em *Vue.js* que torna possível, com uma codificação única, compilar todo o código para diferentes plataformas (QUASAR, 2021). Compilar se refere ao ato de conversão do código para uma linguagem executável pelo sistema operacional ou navegador. É possível compilar facilmente para um aplicativo de página única (SPA - *Single Page Application*), um aplicativo da *Web* progressivo (PWA - *Progressive Web Application*), para aplicativos *desktop* (Mac, Windows e Linux) e até mesmo para aplicativos móveis (Android e iOS).

O Sistema em questão foi compilado como um SPA. Ou seja, um tipo de página *Web* que é inteiramente navegável em um único *index*, sem a necessidade de realizar atualizações da página no navegador, onde todas as requisições ao servidor são feitas via *Javascript*. Sendo assim, ele substitui o uso do módulo *view* que poderia ser usado no servidor, funcionando da forma vista na Figura 2.

Figura 2 – Integração entre o cliente e o servidor no padrão MVC.



Fonte: Acervo pessoal.

O Quasar, apesar de ser um *framework*, funciona inteiramente com base em outro *framework*, o *Vue.js*.

2.2.2.1. Vue.js

Vue (pronuncia-se *view*) é um *framework* de *Javascript* progressivo, utilizado para o desenvolvimento de uma interface visual com a ajuda de componentes reativos. Nas palavras de Incau (2017, p.12):

Componentes reativos nada mais são do que fragmentos de código que possuem sua marcação (HTML), seu estilo (CSS) e seu próprio comportamento (Javascript). [...]

Os componentes são fragmentos menores que as páginas. Na verdade, eles servem para compor as páginas. Desse modo, temos um modo fácil e elegante de se reaproveitar código. Isto é o que temos de mais moderno para o desenvolvimento front-end hoje em dia.

[...] reatividade é tão simples quanto a definição de componentes. A ideia é que, quando a "informação" de um componente mudar (através do Javascript), sua marcação (HTML) saiba reagir e se adaptar a isto.

2.2.3. Comunicação Cliente-Servidor (JSON)

As principais requisições e respostas trocadas entre o cliente e o servidor estarão na linguagem JSON (*Javascript Object Notation*). Uma forma muito leve e simples de realizar a troca de dados entre sistemas. Para Puluceno (2012, p.44):

O JSON é um mecanismo de codificação/decodificação de valores para intercâmbio de dados. Possui uma sintaxe de alto nível, fácil de ser entendida, facilitando o trabalho dos programadores e dos computadores. Ele é nativo da linguagem Javascript. O JSON é também um formato de texto que é completamente independente da linguagem de programação, mas usa convenções que são familiares para os programadores como Java, PHP, C/C++ e muitas outras. Estas propriedades fazem do JSON um formato ideal para transmissão de dados orientado a objetos através da rede.

2.2.3.1. Autenticação (JWT)

A autenticação dos usuários logados no Sistema ocorrerá com o uso do padrão JWT (*JSON Web Token*). Nas palavras de Montanheiro *et al.* (2017):

O *JSON Web Token* (JWT) é um padrão aberto (RFC 7519), que tem por objetivo definir um modo compacto e independente, que pode ser enviado dentro de um cabeçalho HTTP e conter as informações do usuário, para a transmissão segura de informações entre cliente e servidor, através de um objeto JSON. O token gerado pelo JWT é salvo no dispositivo do usuário e suas informações podem ser verificadas a cada solicitação, pois são criptografadas utilizando um segredo, através do algoritmo HMAC (*Hashbased Message Authentication Code*) ou de um par de chaves públicas e privadas, garantindo assim a sua confiabilidade.

Resumindo, JWT é um padrão utilizado para criar um *token*, uma espécie de “passe”, no formato JSON. Ele é usado para transportar dados de forma segura entre diferentes sistemas (neste caso, entre cliente e servidor). Desta forma, as credenciais dos usuários se manterão seguras, mesmo que alguém intercepte o código, pois apenas o servidor terá a chave para codificar e decodificar o *token*.

Quando o usuário efetua o *login*, o *token* é gerado no servidor, carregando dentro de si os dados necessários para identificar aquele usuário e suas permissões dentro do Sistema. O *token* é então enviado para o cliente que o armazenará no *Storage* local do navegador. Foi optado pelo *Storage* local ao invés do *Storage* de sessão, pois o segundo limpa os dados quando a aba ou a janela do navegador é fechada, enquanto o primeiro não.

Toda requisição HTTP (*Hypertext Transfer Protocol*) enviada de um usuário conectado a sua conta pegará o *token* que está no *Storage* local e o colocará em um cabeçalho chamado de *Authorization*. Ao receber a requisição, o servidor irá descriptografar o *token* presente em tal cabeçalho e checar se os dados do usuário

batem com algum dos usuários autorizados presentes no banco de dados, autenticando o mesmo. O Sistema irá buscar também pelas permissões do usuário, antes de o permitir realizar a ação requisitada.

2.2.4. Banco de Dados (MySQL)

O banco de dados foi criado utilizando o MySQL, um dos sistemas de gerenciamento de banco de dados mais famosos do mundo. O serviço, que atualmente pertence à Oracle, utiliza a linguagem SQL (*Structure Query Language*) para criar, modelar e gerenciar os bancos de dados do servidor. SQL é a linguagem mais popular para esse tipo de operação.

Suehring (2002) cita os principais motivos para escolher o MySQL: é ideal para aplicações *Web*, oferece todos os recursos que uma aplicação empresarial precisa, é código aberto, é estável tendo um baixo *overhead* (processamento excessivo), além de as tabelas poderem alcançar grandes tamanhos.

Em tempo de execução, o módulo *model* do código PHP é o responsável por executar as consultas, inserções, edições e exclusões em SQL, realizando o famoso CRUD (*Create Read Update Delete*) no banco de dados.

2.3. Metodologia

A metodologia escolhida para ser utilizada neste artigo foi a pesquisa exploratória com base em levantamento bibliográfico, como artigos e livros ao redor dos temas de biblioteca e das tecnologias empregadas no desenvolvimento do Sistema.

A caracterização do estudo como pesquisa exploratória normalmente ocorre quando há pouco conhecimento sobre a temática a ser abordada. Por meio do estudo exploratório, busca-se conhecer com maior profundidade o assunto de modo a torná-lo mais claro ou construir questões importantes para a condução da pesquisa (RAUPP e BEUREN, 2006, p.80).

2.4. RESULTADOS

2.4.1. Visão Geral do Sistema

O Sistema *Easy Library Center* é dividido em três níveis: público, funcionários e administrador. Quanto mais alto o nível, mais módulos o usuário terá acesso, liberando funções exclusivas para seu respectivo cargo.

Para usuários de nível público, o Sistema apresentará inicialmente uma tela inicial com a opção de pesquisa de livros. O usuário poderá usar alguns filtros para essa pesquisa como categoria, ano, entre outros. Os resultados, além de mostrar uma descrição dos livros, dirá se há ou não exemplares disponíveis para aquele livro. Para os usuários dos níveis a seguir, será necessário efetuar *login* no Sistema.

Os usuários funcionários terão acesso aos módulos de cadastro de livros, exemplares, empréstimos, categorias, autores, leitores e editoras, além de um módulo para a visualização de algumas estatísticas.

E, por fim, o usuário administrador terá acesso total ao Sistema, incluindo um módulo para gerenciar as contas dos outros funcionários e suas respectivas permissões dentro do Sistema.

2.4.2. Diagramas UML

A *Unified Modeling Language* (UML) é a linguagem padrão adotada para a modelagem de diagramas que representam alguns pontos da estrutura de *softwares* orientados a objetos. Nas palavras de Booch *et al.* (2006):

A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis.

Todos os diagramas descritos abaixo podem ser conferidos no Apêndice A no final do artigo.

2.4.2.1. Diagrama de Caso de Uso

O diagrama de caso de uso propõe exemplos de atividades que um usuário poderia fazer dentro do sistema, apresentando um cenário principal e cenários alternativos, em caso de erro de entrada ou outras adversidades que possam vir a ocorrer. Esse diagrama ajuda a definir os requisitos funcionais do sistema.

2.4.2.2. Diagrama de Atividades

O diagrama de atividades tem como finalidade representar como é o fluxo de uma atividade para outra dentro do sistema. Envolve mostrar algumas etapas sequenciais do sistema e quem as executa.

2.4.2.3. Diagrama de Componentes

O diagrama de componentes tem como objetivo demonstrar como estão organizados os componentes que compõem o sistema e com quais outros componentes eles interagem.

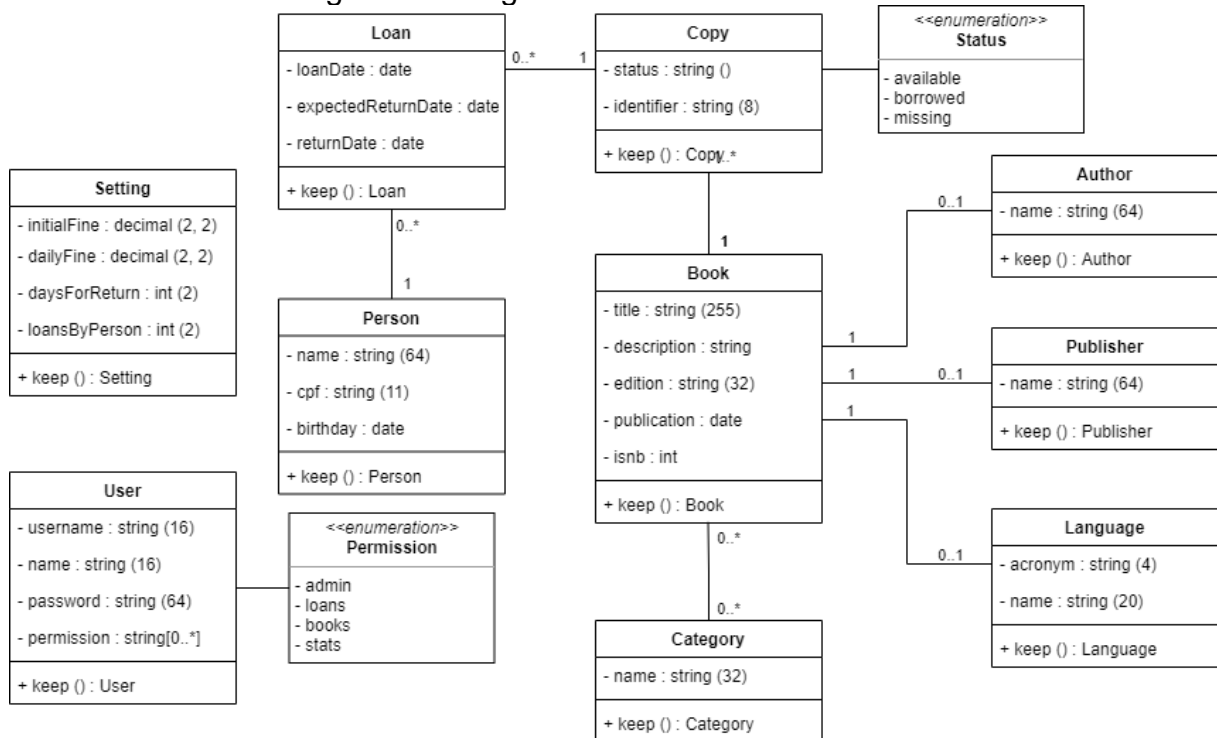
2.4.2.4. Diagrama de Sequência

O diagrama de sequência apresenta um tipo de linha do tempo, mostrando de forma sequencial as tarefas executadas, destacando quem as realiza em diferentes colunas de usuário e módulos do sistema, como servidor e banco de dados.

2.4.2.5. Diagrama de Classes

O diagrama de classes é um dos mais importantes quando o assunto é programação orientada a objetos. Ele representa visualmente como são constituídas as classes do sistema, seus atributos e seus métodos, além das relações delas umas com as outras. É possível também entender a multiplicidade de cada classe, como visto por exemplo na Figura 3, onde cada *Copy* (exemplar) pode ter nenhum ou vários *Loan* (empréstimo), enquanto cada *Loan* terá sempre apenas um exemplar.

Figura 3 – Diagrama de classes do Sistema.



Fonte: Acervo pessoal.

2.4.2.6. Diagrama de Objetos

O diagrama de objetos tem como objetivo simular valores que objetos instanciados de algumas das classes do diagrama anterior podem vir a ter durante a execução do sistema.

2.4.2.7. Diagrama de Máquina de Estado

O diagrama de máquina de estado mostra os possíveis estados que um objeto e os processos responsáveis por essas mudanças de estado.

2.5. SISTEMA

2.5.1. Requisitos do Sistema

O Sistema *Easy Library Center* precisa de dois servidores para o seu funcionamento. Um servidor para o *backend* que rodará PHP em sua versão 7 além de suportar o banco de dados MySQL. E um servidor para *frontend* para guardar os arquivos HTML, CSS, *Javascript* e algumas imagens do cliente.

E para acessar o Sistema, será necessário o uso de um navegador instalado em um computador com acesso à Internet. É recomendável utilizar um navegador atualizado.

2.5.2. Funcionamento do Sistema

Como dito anteriormente, o Sistema é dividido em três níveis de permissão. As telas apresentadas e explicadas do sistemas serão agrupadas por cada nível.

2.5.2.1. Funcionamento do Sistema: Público

As telas a seguir são as que o usuário comum, provável frequentador da biblioteca, terá acesso. Não é necessário nenhum tipo de autenticação para acessar tais páginas.

A Figura 4 exibe a tela inicial do Sistema. Ela apresenta um campo de busca no centro da página destinado à busca de livros. Quando a página é carregada ou o usuário passa o *mouse* em cima do campo, uma dica flutuante explica mais detalhadamente a finalidade da busca.

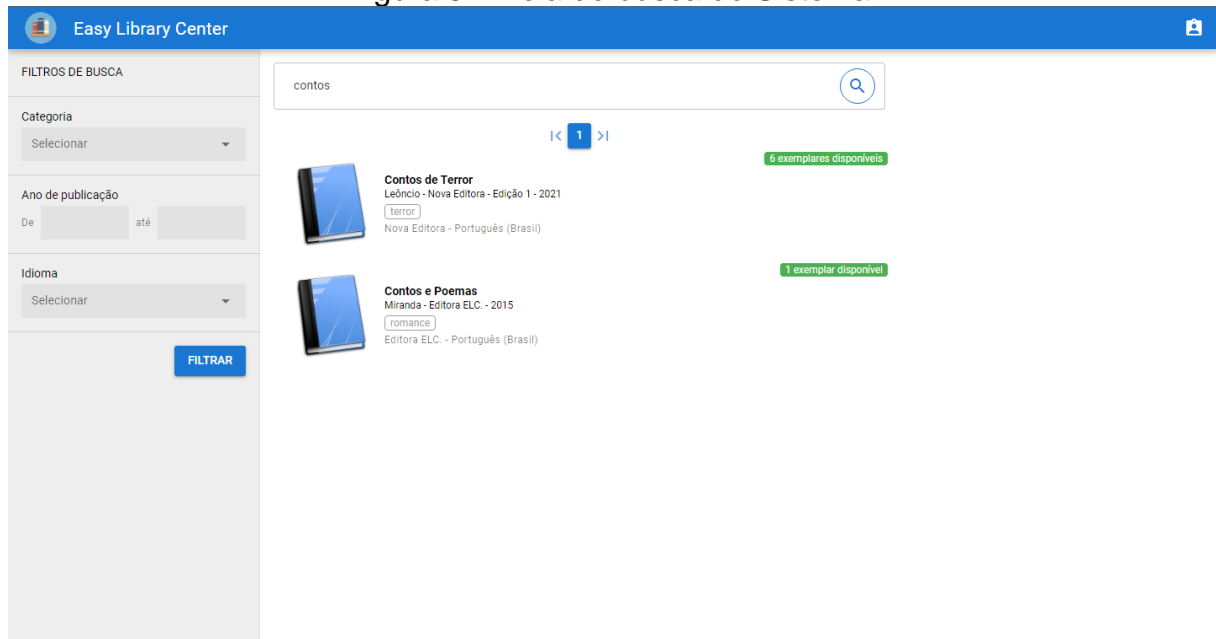
Figura 4 – Tela inicial do Sistema.



Fonte: Acervo pessoal.

Ao realizar a busca o usuário será redirecionado para a tela de busca de livros, exibida na Figura 5. O Sistema apresentará 8 livros por vez, para visualizar os resultados mais distantes, deve-se usar a paginação, encontrada abaixo do campo de busca, no topo da tela. Na esquerda, são apresentados mais filtros, como categoria e ano de publicação, para uma busca mais precisa. Clicando em qualquer um dos livros apresentados no resultado, o usuário é redirecionado para a página do livro, para uma melhor visualização das informações do mesmo.

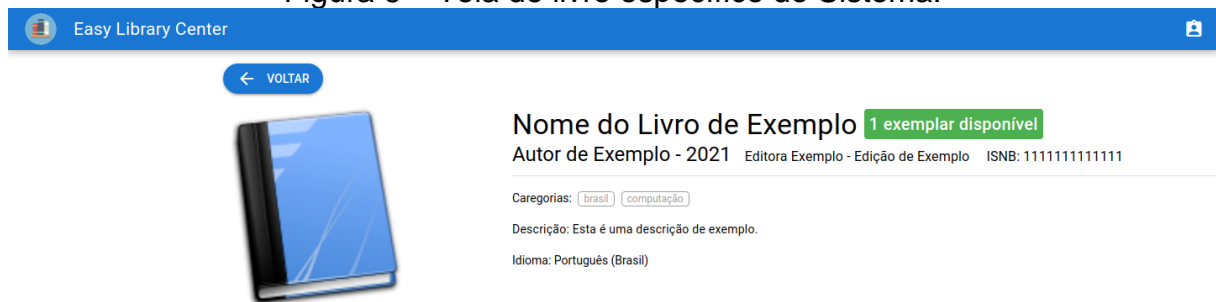
Figura 5 – Tela de busca do Sistema.



Fonte: Acervo pessoal.

Ao abrir a página de um livro, você verá algo similar à Figura 6. Nesta tela é possível visualizar todos os dados públicos do livro, além de sua descrição completa.

Figura 6 – Tela de livro específico do Sistema.



Fonte: Acervo pessoal.

O ícone de credencial no canto superior direito redireciona o usuário para a página de *login* do Sistema, que pode ser vista na Figura 7.

Figura 7 – Tela de *login* do Sistema.clicando aqui'. There are two input fields: 'Nome de usuário' and 'Senha'. At the bottom are two buttons: 'LIMPAR' and 'ENTRAR'."/>

Easy Library Center

Área de login para funcionários

Se você é apenas um frequentador da biblioteca, você pode retornar para a página inicial [clicando aqui](#)

Nome de usuário

Senha

LIMPAR ENTRAR

Fonte: Acervo pessoal.

2.5.2.2. Funcionamento do Sistema: Funcionários

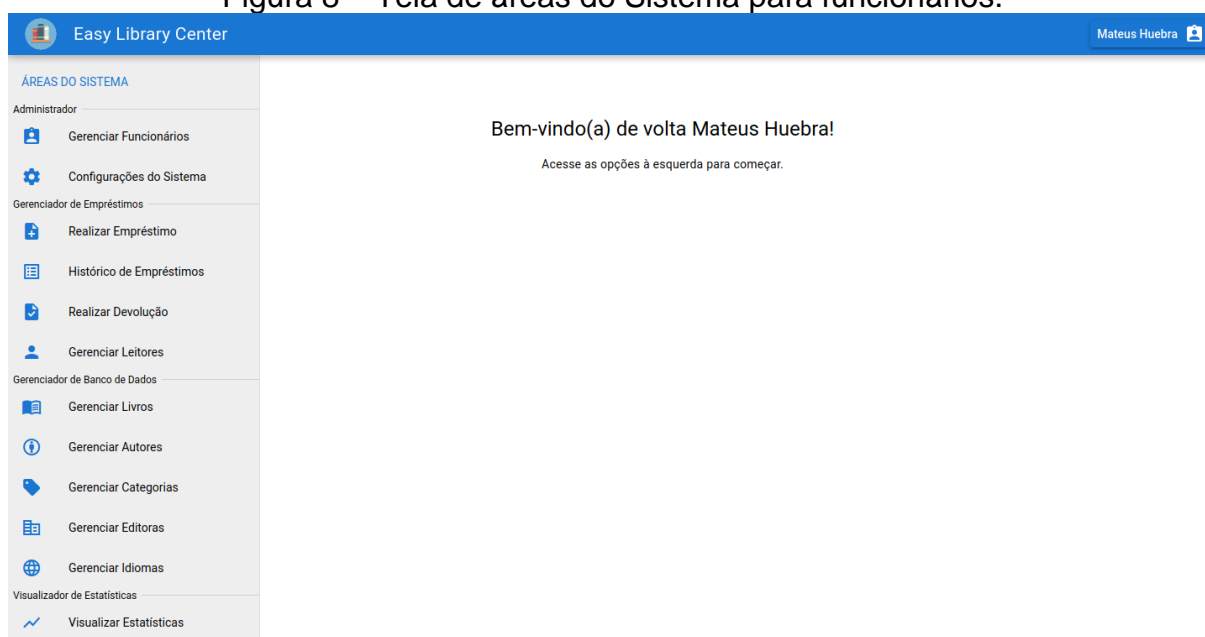
Quando o usuário realiza o *login* e é autenticado, ele é redirecionado para a tela de áreas do Sistema, exibida na Figura 8.

Na esquerda, é possível visualizar um botão para cada área separada em grupos. Cada grupo corresponde a uma permissão que o funcionário tem acesso. Por exemplo, um funcionário com a permissão de “Gerenciador de Empréstimos” poderá realizar empréstimos e devoluções, consultar exemplares e gerenciar os dados dos leitores.

Na direita, fica o painel da área, que corresponderá ao botão pressionado à esquerda. Todas as telas a partir de agora estarão inseridas dentro dele.

Além disso, quando o usuário está logado, o ícone de credencial no canto superior direito dará acesso a um *menu* flutuante. Nele é possível se redirecionar de volta às áreas do Sistema, abrir uma tela de configurações da conta e sair da mesma.

Figura 8 – Tela de áreas do Sistema para funcionários.



Fonte: Acervo pessoal.

A Figura 9 apresenta como é a tela destinada à realização de empréstimos no Sistema. A data de empréstimo e devolução são preenchidas automaticamente como a data corrente e a data limite para devolução baseada nas configurações do Sistema, respectivamente. O usuário deve entrar o nome da pessoa e, caso ela não exista no banco de dados, será dada a opção para inseri-la rapidamente. E por fim entrar o nome do livro.

Para realizar a devolução, basta entrar novamente o nome da pessoa e selecionar o livro a ser devolvido. Se houver atraso, o valor da multa será explicado logo abaixo, pedindo também uma confirmação de que a mesma foi paga, como vemos na Figura 10.

Figura 9 – Painel de realização de empréstimo do Sistema.

Realizar Empréstimo

Empréstimo: 07/11/2021 Devolução até: 14/11/2021

A devolução deve ser realizada em até 7 dias após o empréstimo. Sujeito a multa!
Valor inicial da multa: R\$ 3.00
Valor diário da multa: R\$ 1.00

Pessoa
Mateus Antônio Huebra Henrique

Nome do Livro
Contos e Poemas

Identificador da Cópia
1

REALIZAR EMPRÉSTIMO

Fonte: Acervo pessoal.

Figura 10 – Painel de realização de devolução do Sistema.

Realizar Devolução

Pessoa
Mateus Antônio Huebra Henrique

La Puerta (1)
Empréstimo: 25/08/2021 > Devolução até: 01/09/2021

The Book is on the Table 1 (2)
Empréstimo: 25/08/2021 > Devolução até: 01/09/2021

⚠ A devolução está atrasada em 67 dias
Valor inicial da multa: + R\$ 3.00
Valor diário da multa: + R\$ 1.00 x 66 dias
Valor total da multa: R\$ 69

Confirmar pagamento da multa:

REALIZAR DEVOLUÇÃO

Fonte: Acervo pessoal.

Há várias telas para o gerenciamento do banco de dados no geral. Mas a Figura 11 apresenta a principal delas, a tela de gerenciamento de livros. Como há muitas colunas para visualizar, há a opção de escolher quais você quer ver no momento, evitando poluição visual de informação desnecessária para o momento. Com o mesmo intuito, a descrição, que poluiria muito a tabela, é exibida apenas ao passar o mouse na célula de descrição de um livro.

Figura 11 – Painel para o gerenciamento de livros do Sistema.

Título ↓	Autor	Descrição	Categorias	Publicação	Editora	Ações
Zangões e Abelhas	Miranda Golveia		estudo		Editora A	[Ícone] [Ícone] [Ícone]
The Book is on th...	Albertone		estudo	2014	Publisher Limited	[Ícone] [Ícone] [Ícone]
teste		exemplo de descrição grande demais para caber na tabela.	categorias			[Ícone] [Ícone] [Ícone]
Roberto e Maria	Albertone		romance	2015	Editora Internaci...	[Ícone] [Ícone] [Ícone]
Receitas da Marcone	Marcone da Silva		culinária		Nova Editora	[Ícone] [Ícone] [Ícone]

Fonte: Acervo pessoal.

2.5.2.3. Funcionamento do Sistema: Administrador

Ao logar como usuário administrador, será dado acesso a áreas exclusivas, como a tela de gerenciamento de funcionários (Figura 12). Nesta tela, pode-se visualizar, criar, editar e deletar contar para os funcionários, além de definir as permissões daquela conta. A permissão de “administrador” dá acesso total às funcionalidades do Sistema.

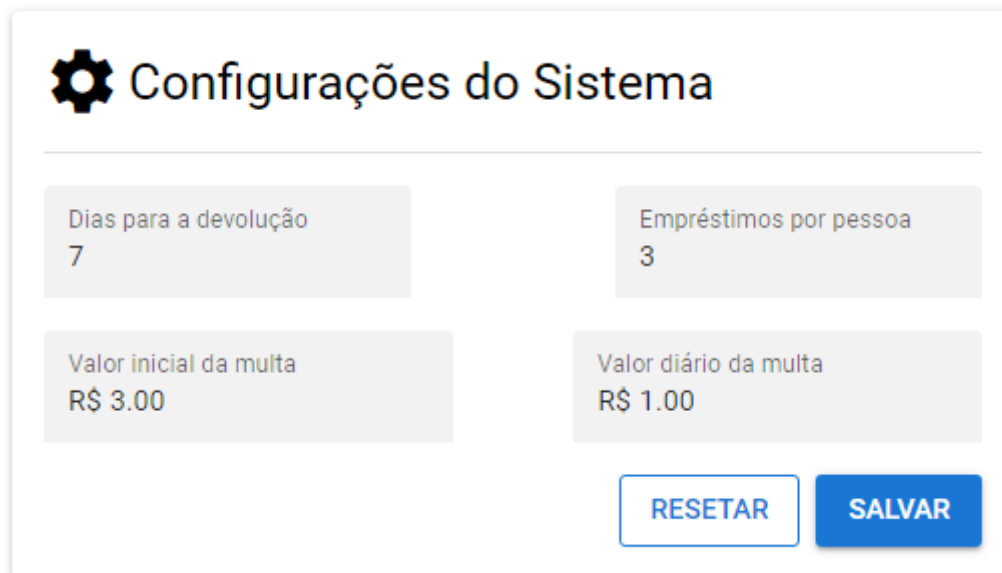
Figura 12 – Painel para o gerenciamento de funcionários do Sistema.

Nome de Usuário ↑	Nome	Permissões	Ações
admin	substituto	administrador	[Ícone] [Ícone]
analista	o analista	estatísticas	[Ícone] [Ícone]
A_test		empréstimos livros	[Ícone] [Ícone]

Fonte: Acervo pessoal.

Por fim, a Figura 13 apresenta a tela de configurações globais do Sistema. Ou seja, são configurações que ditam o funcionamento da aplicação para todos os usuários. Por exemplo, o valor inicial das multas aplicadas às devoluções em atraso.

Figura 13 – Painel de Configurações do Sistema.



A imagem mostra a interface de usuário para as configurações do sistema. No topo, há um ícone de engrenagem e o título "Configurações do Sistema". Abaixo, há quatro cartões de configuração:

- Dias para a devolução: 7
- Empréstimos por pessoa: 3
- Valor inicial da multa: R\$ 3.00
- Valor diário da multa: R\$ 1.00

Na parte inferior direita, há dois botões: "RESETAR" (branco com borda azul) e "SALVAR" (azul sólido).

Fonte: Acervo pessoal.

3. CONCLUSÃO

Todo esse Sistema serve, então, como resposta à pergunta de “como automatizar o gerenciamento das informações mantidas por uma biblioteca, além de torná-la mais atrativa em meio a modernidade?” feita na introdução do artigo.

O Sistema automatiza e simplifica toda a gestão das informações da biblioteca que serão disponibilizadas em tabelas de fácil aprendizado ao uso. Além de tornar a biblioteca um pouco mais atrativa e menos obsoleta no contexto tecnológico atual, permitindo aos frequentadores da biblioteca consultarem a existência e disponibilidades de exemplares sem sair de casa, via Internet.

Com um sistema automatizado, qualquer procedimento pode ser realizado em poucos segundos, trazendo uma facilidade e rapidez enorme quando comparado ao método antigo das bibliotecas, as fichas e arquivos de papel.

Quanto a possibilidades do futuro da aplicação, como ela é separada em cliente e servidor, é possível facilmente criar uma aplicação cliente para dispositivos móveis, como *Android* e *iOS*. Isso tornaria o Sistema ainda mais acessível ao público, alcançando uma gama ainda maior de dispositivos suportados. Outra área com possíveis melhorias seria a de estatísticas, que pode ser feita de uma forma mais customizável e com gráficos.

Seria possível também criar versões um pouco diferentes do Sistema com funções específicas para uma certa biblioteca para, por exemplo, uma biblioteca escolar a qual iria requisitar o número de matrícula de um aluno ao retirar um livro.

4. REFERÊNCIAS

BATISTA, Emerson de Oliveira. **Sistemas de informação**. Saraiva Educação SA, 2017.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Elsevier Brasil, 2006.

CARNEIRO DOS SANTOS, M. **Conversando com uma API: um estudo exploratório sobre TV social a partir da relação entre o twitter e a programação da televisão**. Revista GEMInIS, v. 4, n. 1, p. 89-107, 11 ago. 2013.

CUNHA, Murilo Bastos da. **Desafios na construção de uma biblioteca digital**. Ciência da Informação, Brasília, v. 28, n. 3, p. 257-268, set./dez. 1999. Disponível em: <https://www.scielo.br/pdf/ci/v28n3/v28n3a3>. Acesso em: 17 mar. 2021.

GAMMA, Erich. **Padrões de projetos: soluções reutilizáveis**. Bookman editora, 2009.

INCAU, Caio. **Vue.js: Construa aplicações incríveis**. Editora Casa do Código, 2017.

MONTANHEIRO, Lucas Souza; CARVALHO, Ana Maria Martins; RODRIGUES, Jackson Alves. **Utilização de JSON Web Token na Autenticação de Usuários em APIs REST**. XIII Encontro Anual de Computação. Anais do XIII Encontro Anual de Computação EnAComp, p. 186-193, 2017.

MORIGI, Valdir José; SOUTO, Luzane Ruscher. **Entre o passado e o presente: as visões de biblioteca no mundo contemporâneo**. Revista ACB, [S.l.], v. 10, n. 2, p. 189-206, jan. 2006. ISSN 1414-0594. Disponível em: <https://revista.acbsc.org.br/racb/article/view/432/551>. Acesso em: 18 mar. 2021.

ORLANDO, Alex Fernando. **Uma infra-estrutura computacional para o gerenciamento de programas de ensino individualizados**. 2009.

PULUCENO, Thiago Vieira. **Estudo de caso sobre uma API REST em Node.js**. 2012.

QUASAR. Quasar Framework: **SPA front-end on steroids**. 2021. Disponível em: <https://quasar.dev>. Acesso em: 20 set. 2021.

RAUPP, Fabiano Maury; BEUREN, Ilse Maria. Metodologia da pesquisa aplicável às ciências. **Como elaborar trabalhos monográficos em contabilidade: teoria e prática**. São Paulo: Atlas, p. 80, 2006.

RODRIGUES, Anielma Maria Marques; PRUDÊNCIO, Ricardo Bastos Cavalcante. **Automação: a inserção da biblioteca na tecnologia da informação**. Biblionline, João Pessoa, v. 5, n. 1/2, p. 16, 2009.

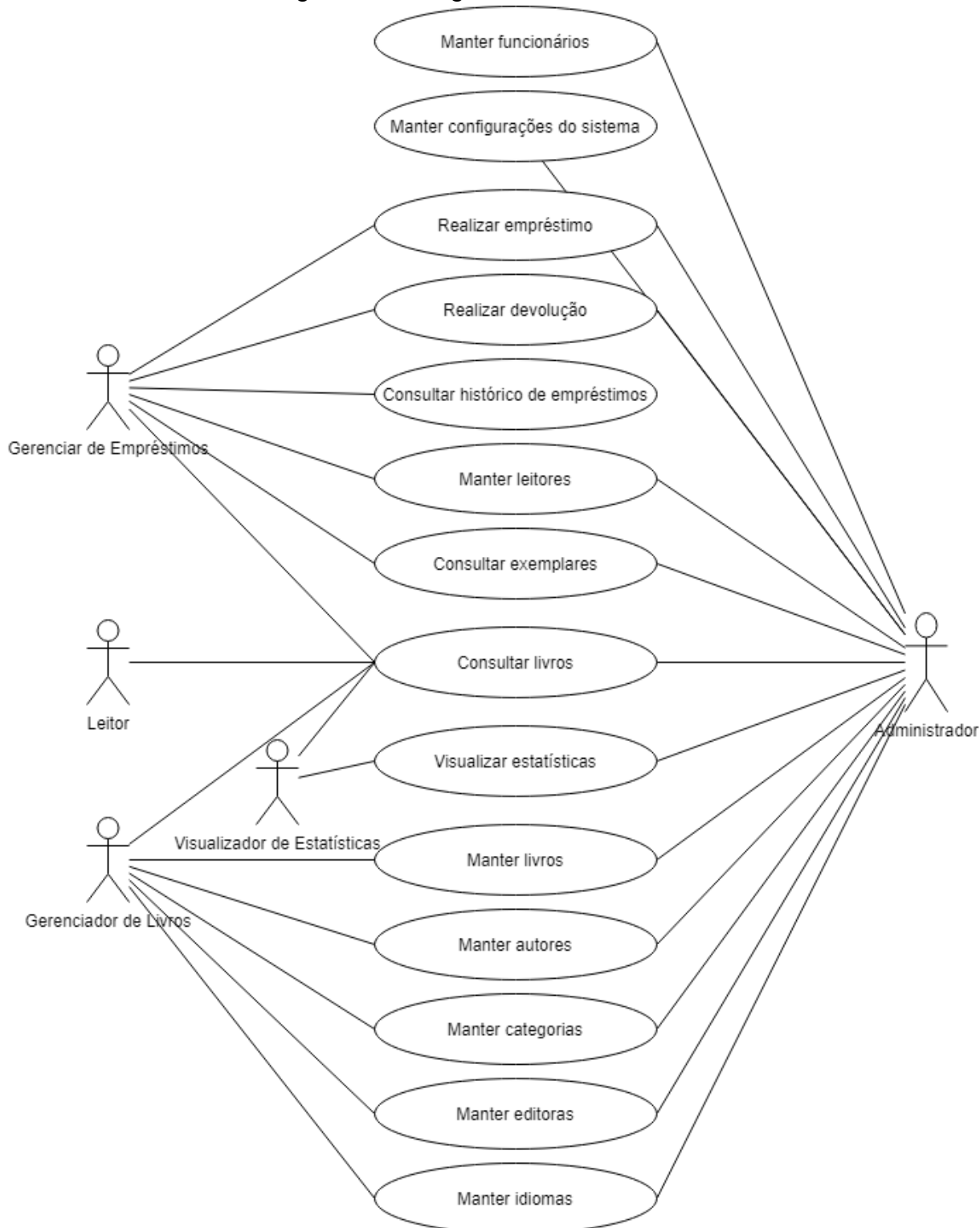
SUEHRING, Steve. **MySQL Bible**. – New York: Wiley Publishing, Inc., p. 50. 2002.

5. APÊNDICE A – DIAGRAMAS UML

Este apêndice apresenta todos os sete diagramas da UML utilizados no desenvolvimento do Sistema.

5.1. Diagrama de Caso de Uso

Figura 14 – Diagrama de caso de uso.



Fonte: Acervo pessoal.

A Figura 14 apresenta quais tipos de usuários têm acesso a o que. O leitor consegue apenas realizar uma consulta nos livros, enquanto cada um dos três funcionários à esquerda tem acesso a um grupo de ações respectivas ao seu papel no Sistema. O administrador tem acesso a todas as ações.

O diagrama de caso de uso é seguido por uma especificação da mesma, como visto abaixo:

Objetivo: Realizar o empréstimo de um exemplar de livro

Ator: Gerenciador de Empréstimos

5.1.1. Cenário principal

1. O gerenciador abre a página de realizar empréstimos no sistema.
2. O sistema recupera as configurações do sistema (valores de multa e limite de devolução).
3. O gerenciador informa o nome da pessoa.
4. O sistema recupera uma lista de pessoas com o nome.
5. O gerenciador seleciona a pessoa.
6. O gerenciador informa o nome do livro.
7. O sistema recupera uma lista de livros com o nome.
8. O gerenciador seleciona o livro.
9. O gerenciador informa o código do exemplar.
10. O sistema recupera uma lista de exemplares com o código.
11. O gerenciador seleciona o exemplar.
12. O gerenciador solicita o empréstimo.
13. O sistema registra o empréstimo.

Cenários alternativos

1. O gerenciador informou o nome da pessoa errado ou inexistente no banco de dados do sistema.

Caso afirmativo, o sistema exibe um botão para adicionar a pessoa ao sistema.

2. O gerenciador informou o nome do livro errado ou inexistente no banco de dados do sistema.

Caso afirmativo, o sistema retorna uma mensagem dizendo que o livro não existe.

3. O gerenciador informou o código do exemplar ou inexistente no banco de dados do sistema.

Caso afirmativo, o sistema retorna uma mensagem dizendo que o exemplar não existe.

4. O gerenciador não selecionou a pessoa, o livro e o exemplar e solicitou o empréstimo.

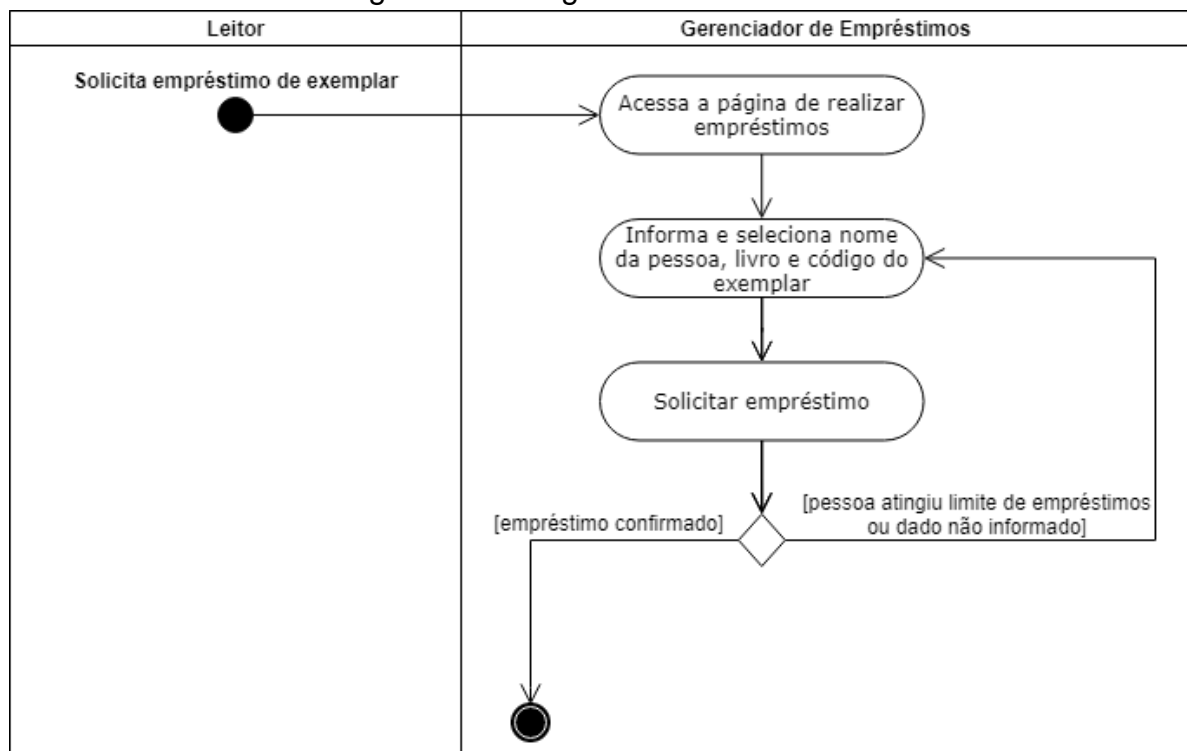
Caso afirmativo, o sistema retorna uma mensagem dizendo o que faltou selecionar.

Requisitos especiais

1. O gerenciador precisa de um navegador com acesso à Internet.

5.2. Diagrama de Atividades

Figura 15 – Diagrama de atividades.

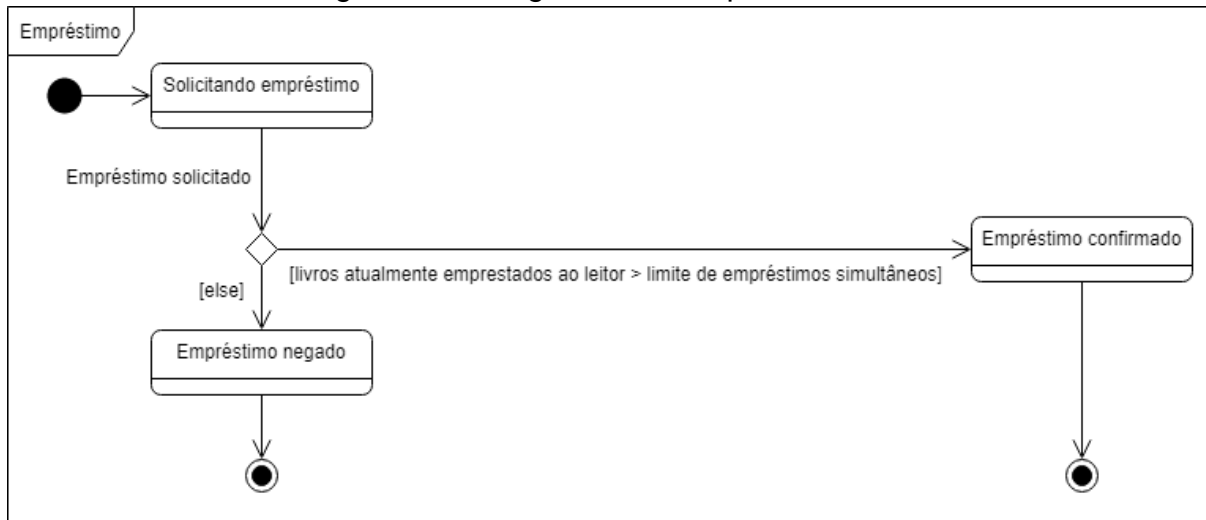


Fonte: Acervo pessoal.

A Figura 15 apresenta as atividades necessárias para a realização do empréstimo de um exemplar no Sistema. O empréstimo só é confirmado após o preenchimento de todos os dados além da validação do limite de empréstimos do leitor em questão não ter sido já atingido.

5.3. Diagrama de Máquina Estado

Figura 16 – Diagrama de máquina estado.

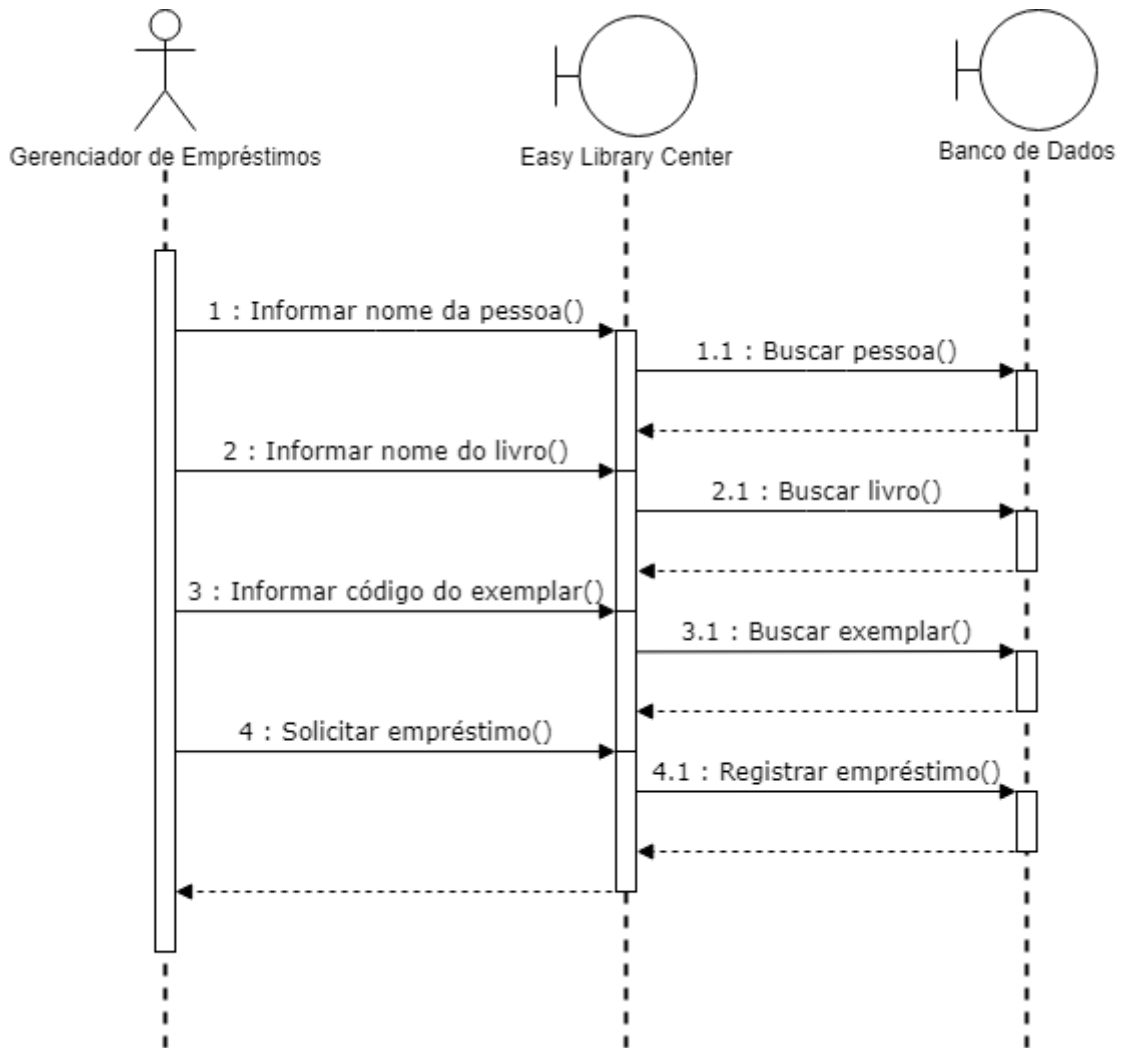


Fonte: Acervo pessoal.

A Figura 16 apresenta os estados para o mesmo processo da figura anterior, onde o empréstimo só é confirmado após as devidas validações.

5.4. Diagrama de Sequência

Figura 17 – Diagrama de sequência.

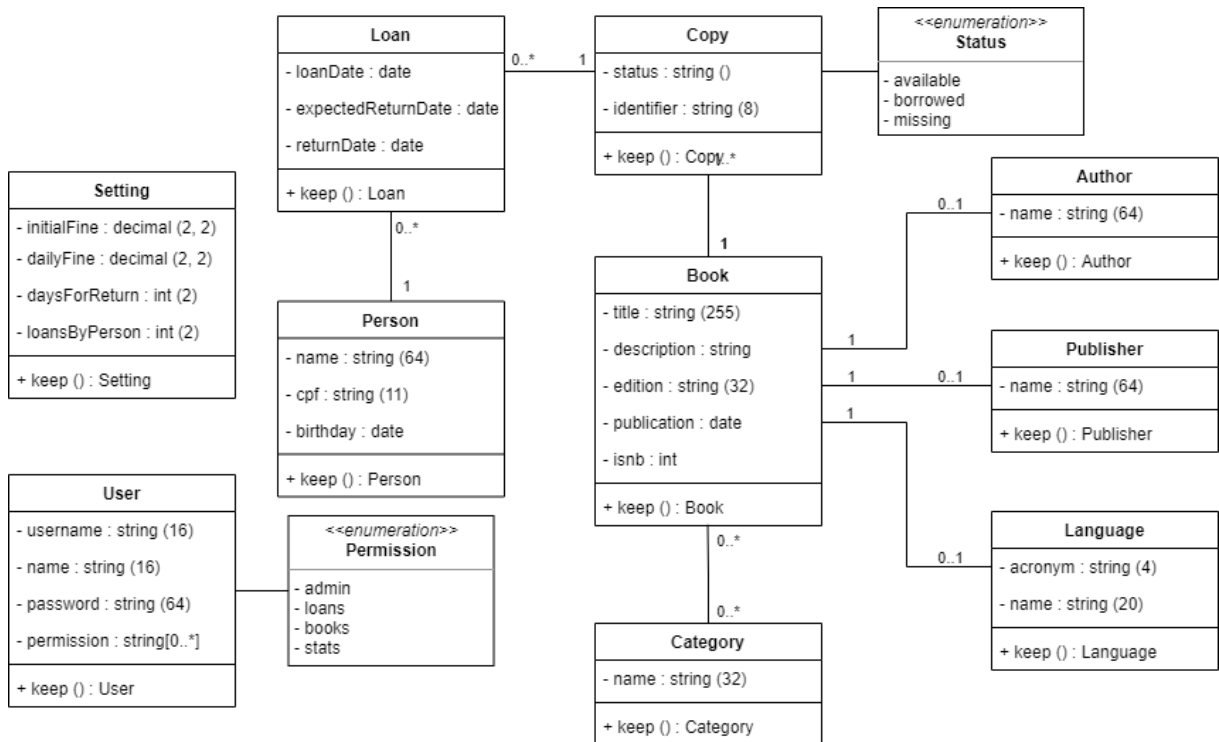


Fonte: Acervo pessoal.

A Figura 17 apresenta a sequência de atividades/processos que cada ator realiza durante a solicitação de um empréstimo.

5.5. Diagrama de Classes

Figura 18 – Diagrama de classes.

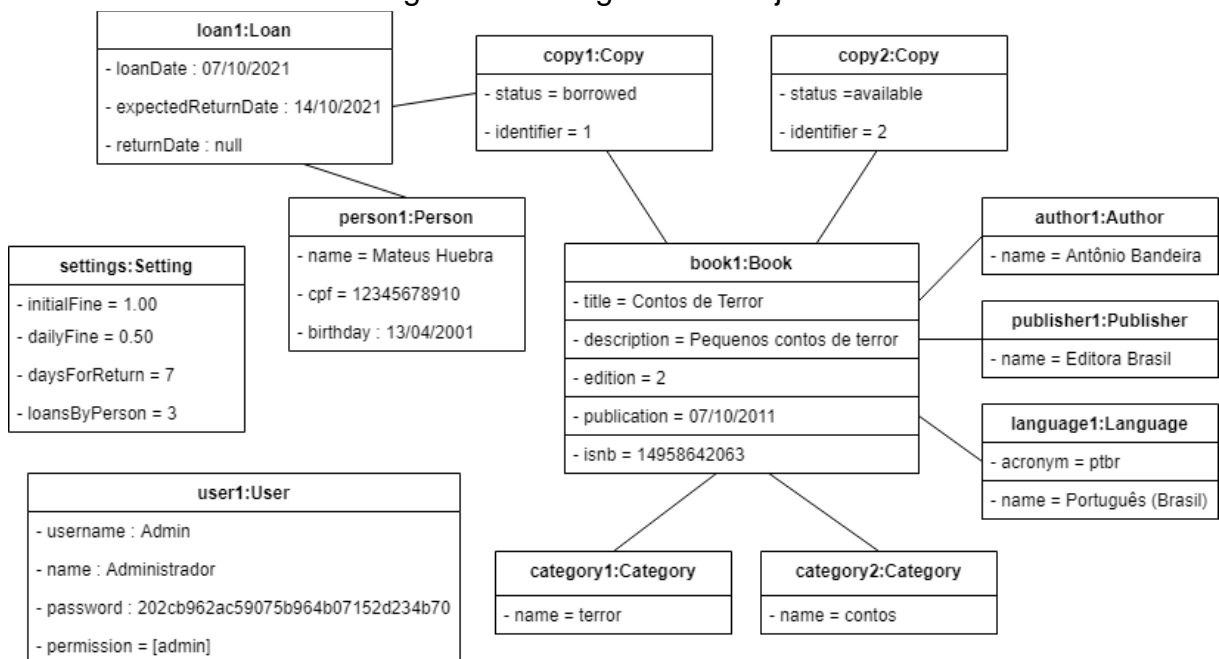


Fonte: Acervo pessoal.

A Figura 18 apresenta todas as classes do Sistema, detalhando como tudo é organizado no banco de dados do mesmo, além de suas conexões.

5.6. Diagrama de Objetos

Figura 19 – Diagrama de objetos.

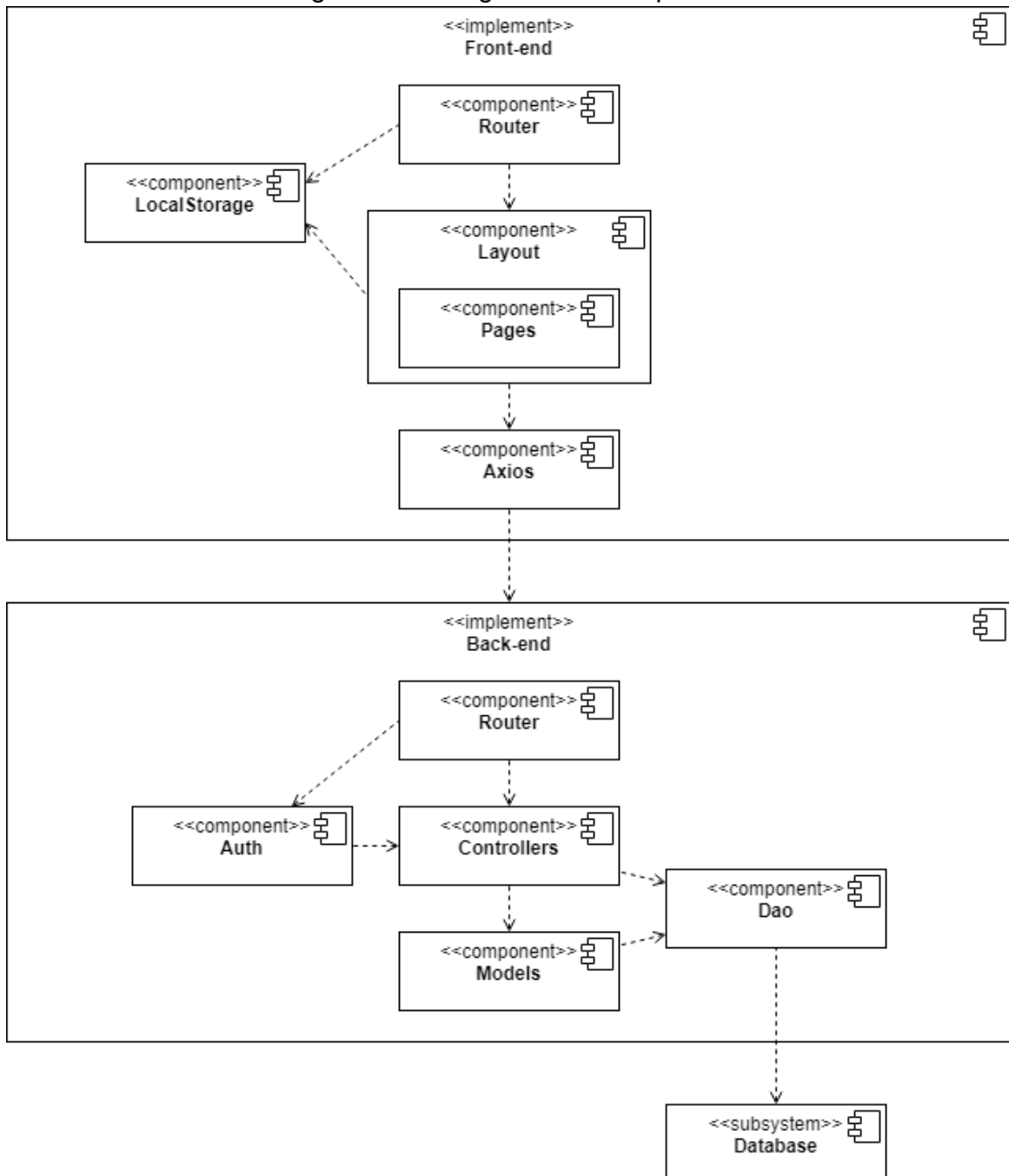


Fonte: Acervo pessoal.

A Figura 19 apresenta exemplos de objetos, com dados simulados, para todas as classes apresentadas na figura anterior.

5.7. Diagrama de Componentes

Figura 20 – Diagrama de componentes.



Fonte: Acervo pessoal.

A Figura 20 apresenta os principais componentes do Sistema, que é dividido em Front-end e Back-end. Ambos possuem um Router que redireciona para a página correspondente ao endereço solicitado.

6. APÊNDICE B – TELAS DO SISTEMA

Neste apêndice estão apresentadas algumas das telas do Sistema que não estão presentes no decorrer do artigo.

Figura 21 – Modal de adicionar livro do Sistema.

Adicionar Livro

ISBN Ano de Publicação

Título Edição

Descrição
Máximo: 512 caracteres 0

Autor
mi
Miranda

Idioma

Descrição

Máximo: 512 caracteres 0

Autor
Miranda

Categorias

Idioma

Editora

RESETAR ADICIONAR

Fonte: Acervo pessoal.

Figura 22 – Modal de exemplares do Sistema.

Exemplares do Livro

Contos e Poemas

ADICIONAR EXEMPLAR +

Identificador	Status	Ações
1	disponível	
2	disponível	

FECHAR

Fonte: Acervo pessoal.

A Figura 21 e a Figura 22 apresentam as janelas flutuantes da tabela de livros, onde é possível adicionar ou atualizar livros, além de visualizar seus exemplares. Toda tabela possui sua própria janela flutuante de adição de novo item.

Figura 23 – Painel do histórico de empréstimos do Sistema.

Empréstimo ↓	Expectativa de Devolução	Devolução	Leitor	Livro
30/11/2021	07/12/2021		Mateus Antônio Hu... ver dados	A Volta ver dados 1
30/11/2021	07/12/2021		Leitor Excluído (...) ver dados	Física Quântica ver dados 2
29/11/2021	06/12/2021		Luroca ver dados	A Volta ver dados 2
03/09/2021	10/09/2021		Leitor Excluído (...) ver dados	Contos Românticos ver dados 2

Fonte: Acervo pessoal.

Figura 24 – Painel de autores do Sistema.

Nome ↑	Livros com o Autor	Ações
Albertone	6	✎ 🗑️
Dante	3	✎ 🗑️
Gabriel Mousito	5	✎ 🗑️

Fonte: Acervo pessoal.

Figura 25 – Painel de categorias do Sistema.

Categorias Adicionar Categoria +

Colunas Visíveis Nome

Nome ↓	Livros com a Categoria	Ações
terror	4	 
romance	5	 
religião	1	 

Categorias por página 3 1-3 de 16 |< < > >|

Fonte: Acervo pessoal.

Figura 26 – Painel de editoras do Sistema.

Editoras Adicionar Editora +

Colunas Visíveis Nome

Nome	Livros com a Editora ↓	Ações
Editora Internacional	6	 
Editora ELC.	4	 
Nova Editora	4	 







Editoras por página 3 1-3 de 9 |< < > >|

Fonte: Acervo pessoal.

Figura 27 – Painel de idiomas do Sistema.

Idiomas Adicionar Idioma +

Colunas Visíveis Sigla Nome

Sigla	Nome ↓	Livros com o Idioma	Ações
ptpt	Português (Portugal)	2	 
ptbr	Português (Brasil)	19	 
enus	Inglês (EUA)	2	 


Idiomas por página 3 1-3 de 5 < >

Fonte: Acervo pessoal.







As Figuras 23, 24, 25, 26 e 27 mais exemplos de tabelas que podem ser encontradas pelo Sistema. Com destaque à Figura 23 que demonstra o histórico de empréstimos.

Figura 28 – Painel de estatísticas do Sistema.

Visualize as categorias e os livros mais emprestados na faixa de data desejada:

 Data de Empréstimo
01/08/2021 ~ 31/08/2021

Categoria	Quantidade
estudo	4
terror	2
ficção	2
culinária	1
ação	1

Ver	Livro	Quantidade
	The Book is on the Table	4
	Horror Stories from Alber	2
	O Excelentissimo	2
	Receitas da Marcone	1
	La Puerta	1
	O Terceiro Samurai	1

Fonte: Acervo pessoal.

A Figura 28 apresenta a tela de estatísticas, onde é possível visualizar as categorias e livros mais emprestados de certo período.