

## **EASY LIBRARY CENTER: SISTEMA DE GESTÃO PARA BIBLIOTECAS**

**Mateus Antônio Huebra Henrique<sup>1</sup>, Ezequias Ferreira de Souza<sup>2</sup>, Lidiane Hott de Fúcio Borges<sup>3</sup>, Luciana Rocha Cardoso<sup>4</sup>, Ludmilla Breder Furtado<sup>5</sup>**

<sup>1</sup>Graduado em Análise e Desenvolvimento de Sistemas, UNIFACIG, Manhuaçu-MG,  
[1910442@sempre.unifacig.edu.br](mailto:1910442@sempre.unifacig.edu.br)

<sup>2</sup>Mestre em Desenvolvimento Local, UNIFACIG, Manhuaçu-MG,  
[ezequias.souza@sempre.unifacig.edu.br](mailto:ezequias.souza@sempre.unifacig.edu.br)

<sup>3</sup>Mestre em Engenharia e Ciência dos Materiais. UNIFACIG, Manhuaçu-MG.  
[pedagogia@unifacig.edu.br](mailto:pedagogia@unifacig.edu.br)

<sup>4</sup>Mestre em Informática, UNIFACIG, Manhuaçu-MG, [ludmila@sempre.unifacig.edu.br](mailto:ludmila@sempre.unifacig.edu.br)

<sup>5</sup>Mestre em Ciência da Computação, UNIFACIG, Manhuaçu-MG, [luroca@sempre.unifacig.edu.br](mailto:luroca@sempre.unifacig.edu.br)

**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 autorial 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.

**Área do Conhecimento:** Ciências Exatas e da Terra.

## **EASY LIBRARY CENTER: MANAGEMENT SYSTEM FOR LIBRARIES**

**Abstract:** Managing files and information physically on paper is an outdated method. The developed application aims to automate and facilitate the management of all data held by a library, in addition to making it more attractive in the midst of modernity, providing a public page to check the existence and availability of books and copies. The application was developed using a simple and authorial framework in PHP language (Hypertext Preprocessor), on the server, and the Quasar framework, which uses Javascript, on the client. In addition to the use of MySQL as a database management system. UML (Unified Modeling Language) diagrams were also used for the project. The System includes tables for registering and controlling books, copies, loans, categories, authors, readers and publishers, in addition to a module for viewing some statistics.

**Keywords:** Libraries; Management; System.

### **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 tomaria 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.

## METODOLOGIA

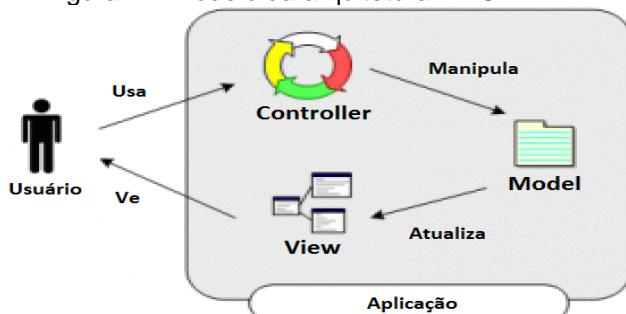
A metodologia escolhida para ser utilizada neste artigo foi a pesquisa exploratória com base em levantamento bibliográfico, como artigos e livros ao redor dos temas de biblioteca e das tecnologias empregadas no desenvolvimento do Sistema.

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).

## 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. O *controller* processa as requisições do usuário, manipulando as classes de *model*, retornando os dados para a camada *view*.

Figura 1 – Modelo da arquitetura MVC.



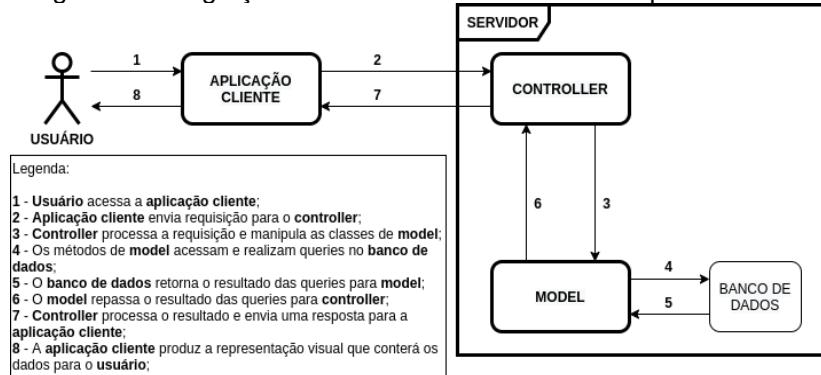
Fonte: (ORLANDO, 2009).

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

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



Fonte: Acervo pessoal.

## 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.

## 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.

## 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 ou de um par de chaves públicas e privadas, garantindo assim a sua confiabilidade.

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). 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 identificação de tal.

## **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*, 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, e exclusões em SQL, realizando o famoso CRUD dos dados.

## **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 cargo. Para usuários de nível público, o Sistema apresentará uma tela com a opção de pesquisa de livros, o usuário poderá usar alguns filtros para pesquisa como categoria, ano, *etc*. Os resultados, mostra uma descrição dos livros, dirá se há ou não exemplares para aquele livro. Para os usuários dos níveis a seguir, é necessário efetuar *login* no Sistema. Os usuários funcionários terão acesso aos módulos de cadastro de livros, empréstimos, categorias, autores, leitores e editoras, e um módulo para a visualização de algumas estatísticas. O usuário administrador terá acesso total ao Sistema e um módulo para gerenciar as contas dos outros funcionários.

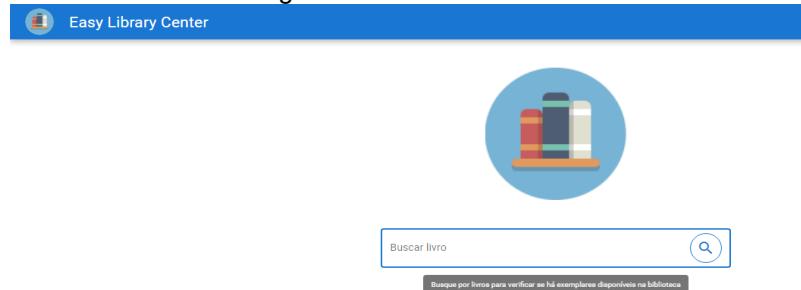
## **REQUISITOS DO SISTEMA E FUNCIONAMENTO DO SISTEMA**

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.

## **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.

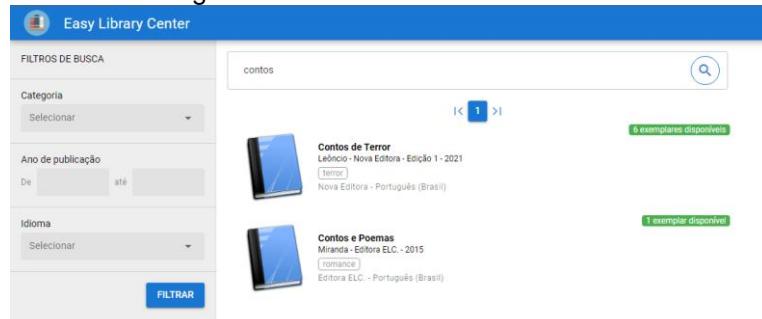
Figura 4 – Tela inicial do Sistema.



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.

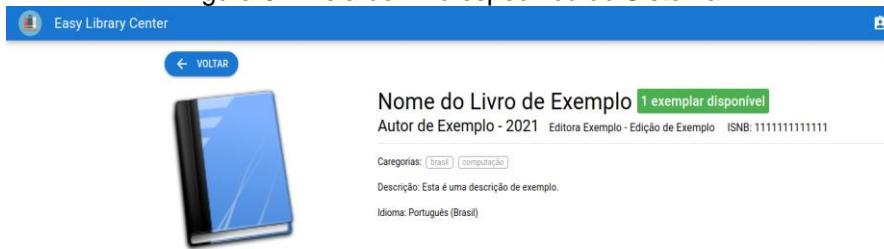
Figura 5 – Tela de busca do Sistema.



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.

A interface de usuário para login de funcionários. No topo, uma barra azul com o logo "Easy Library Center". Abaixo, uma seção intitulada "Área de login para funcionários". Um aviso diz: "Se você é apenas um frequentador da biblioteca, você pode retornar para a página inicial [clicando aqui](#)". Há campos para "Nome de usuário" e "Senha", com botões "LIMPAR" e "ENTRAR" ao lado. Abaixo dos campos, uma barra com links: "ÁREAS DO SISTEMA", "Administrador", "Gerenciador de Empréstimos", "Gerenciador de Livros", "Gerenciador de Autores", "Gerenciador de Editoras", "Visualizar Estatísticas", "Logout".

Fonte: Acervo pessoal.

## 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.

A interface de usuário para áreas do Sistema. No topo, uma barra azul com o logo "Easy Library Center" e o nome do usuário "Mateus Huebra". Abaixo, uma seção intitulada "ÁREAS DO SISTEMA" com uma lista de opções: "Administrador", "Gerenciador de Empréstimos", "Gerenciador de Livros", "Gerenciador de Autores", "Gerenciador de Editoras", "Visualizar Estatísticas", "Logout". A direita, uma seção com o nome "Bem-vindo(a) de volta Mateus Huebra!" e o link "Acesse as opções à esquerda para começar".

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.

A interface de usuário para realizar empréstimo. No topo, uma barra com o link "+ Realizar Empréstimo". Abaixo, campos para "Empréstimo: 07/11/2021" e "Devolução até: 14/11/2021". Um aviso diz: "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". Há campos para "Pessoa" (Mateus Antônio Huebra Henrique), "Nome do Livro" (Contos e Poemas) e "Identificador da Cópia" (1). Um botão "REALIZAR EMPRÉSTIMO" está no fundo.

Fonte: Acervo pessoal.

Figura 10 – Painel de realização de devolução do Sistema.

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.

Fonte: Acervo pessoal.

## FUNCIONAMENTO DO SISTEMA: ADMINISTRADOR

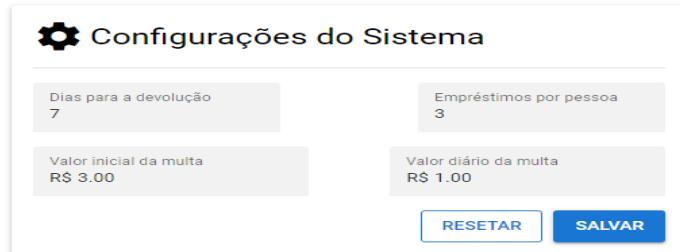
Ao logar como usuário administrador, será dado acesso a áreas exclusivas, como a tela de gerenciamento de funcionários (Figura 12). Nesta tela, pode-se visualizar, criar, editar e deletar contas para os funcionários, além de definir as permissões daquela conta. A permissão de “administrador” dá acesso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

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

Fonte: Acervo pessoal.

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.



Fonte: Acervo pessoal.

## RESULTADOS E DISCUSSÃO

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. 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. 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. O Sistema para automatizar o controle de todos os livros, exemplares, leitores e empréstimos nos arquivos da biblioteca, 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. 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 de o proprietário da organização desejar, mais futuramente, desenvolver um cliente *mobile*, isto é, um aplicativo para os dispositivos móveis.

## 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.

## 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.I.], 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.