

Sistema de Gerenciamento de Acervos Bibliotecários

Lavínia Dantas de Mesquita
PEX1271 - Teste de Software
UFERSA

Pau dos Ferros, Brasil
lavinia.mesquita@alunos.ufersa.edu.br

Pedro Vinícius de Andrade Queiroz
PEX1271 - Teste de Software
UFERSA

Pau dos Ferros, Brasil
pedro.queiroz@alunos.ufersa.edu.br

Pollyana Dias Paiva
PEX1271 - Teste de Software
UFERSA

Pau dos Ferros, Brasil
pollyana.paiva@alunos.ufersa.edu.br

Resumo—Este documento é o resultado do estudo e da concepção do projeto SGAB (Sistema de Gerenciamento de Acervos Bibliotecários), uma solução destinada a aprimorar o gerenciamento acervos literários de bibliotecas, abrangendo tanto acervos privados como públicos. É também o projeto avaliativo da 3ª unidade da disciplina Teste de Software. O foco é estender nossa influência e utilidade a uma vasta região, com destaque para a Biblioteca Campus Pau dos Ferros, localizada na UFERSA da cidade de Pau dos Ferros. O principal objetivo é promover o estudo e a leitura além das barreiras físicas das bibliotecas, facilitando o empréstimo de exemplares de livros e simplificando a administração do acervo para os responsáveis pelo mesmo, tornando a jornada de pesquisa, aprendizado, manutenção e preservação de livros de forma eficiente para todos os envolvidos. Principalmente, este artigo mostrará a pesquisa e os resultados dos testes executados neste software, que são: —

Palavras-Chave—Biblioteca, Sistema de Gerenciamento, Teste de Software, Banco de Dados.

Abstract—This document is the result of the study and design of the SGAB project (Library Collection Management System), a solution aimed at enhancing the management of literary collections in libraries, encompassing both private and public collections. It also serves as the evaluative project for the 3rd unit of the Object-Oriented Programming course. The focus is to expand our influence and utility to a wide region, with a special emphasis on the Campus Pau dos Ferros Library located at UFERSA in the city of Pau dos Ferros. The main objective is to promote study and reading beyond the physical boundaries of libraries, making it easier to borrow book copies and simplifying the management of the collection for those responsible for it. This will make the journey of research, learning, maintenance, and book preservation efficient for all parties involved. Primarily, this article will showcase the research and the results of the tests conducted on this software, which include: —

Keywords—Library, Management System, Software Testing, Database.

I. INTRODUÇÃO

No âmbito dos sistemas de gerenciamento, o projeto SGAB (Sistema de Gerenciamento de Acervos Bibliotecários) emerge como uma solução promissora para documentar, organizar e supervisionar eficazmente as coleções de bibliotecas. Este documento representa o resultado de uma pesquisa intensiva e da concepção de uma ferramenta de software simples e eficaz. Projetado para aprimorar a gestão de coleções de bibliotecas, tanto privadas quanto públicas, o SGAB é um testemunho

das capacidades da Programação Orientada a Objetos e dos princípios que sustentam operações eficientes de bibliotecas.

Primeiramente, exploramos os conceitos fundamentais necessários para o desenvolvimento e utilização ótima do sistema SGAB. No cerne deste projeto está a combinação de tecnologia, alocação de recursos e planejamento estratégico para harmonizar processos reais com os registros virtuais aprimorando a organização de acervos bibliotecários.

A escolha da tecnologia, incluindo a versátil linguagem de programação Java e o robusto sistema de banco de dados PostgreSQL mantido em nuvem com o auxílio do ElephantSQL é explorada. A justificativa por trás dessas escolhas reside em sua capacidade de criar um sistema confiável, seguro e verificado para o armazenamento e manipulação de dados de bibliotecas, garantindo operações sem intercorrências e a satisfação dos usuários.

II. FUNDAMENTAÇÃO TEÓRICA

O SGAB é uma ferramenta de Software capaz de documentar, gerenciar e organizar acervos bibliotecários. Aqui, abordaremos os princípios e conceitos necessários para seu desenvolvimento e uso eficaz:

Primeiramente, segundo Richard L. Darf (DAFT, 2016), um sistema de gerenciamento é, basicamente, uma estrutura organizacional que integra processos, recursos e tecnologia com o intuito de alcançar alguma meta numa organização. Primeiro, planeja-se o processo começa com o estabelecimento de metas, no nosso caso, é o auxílio nas ações de empréstimo e organização geral do acervo de uma biblioteca. Depois, entra a etapa de organização, onde são definidas as estruturas e atribuições de responsabilidades, que serão vistas posteriormente.

Posteriormente, com os recursos alocados, o sistema coordena a execução das atividades de acordo com as metas. Vale salientar a necessidade de um administrador que mantenha o controle do gerenciamento para garantir que as metas sejam cumpridas, o que isso significa? Significa que após o sistema ser desenvolvido, os administradores precisam alinhar as ações na biblioteca com registros dentro do sistema, se um livro novo chegar, ele deve ser cadastrado, se um livro for emprestado, ele deve ser marcado como tal, entre outros pontos.

Por fim, quando o sistema for testado por um tempo, ele deve ser avaliado para que possíveis melhorias sejam adicionadas. No nosso caso, as melhorias serão listadas na conclusão deste documento. Agora, é essencial mostrar os pontos principais do projeto:

- Alinhamento com os princípios de uma biblioteca: Estes princípios incluem o fornecimento de acesso a materiais de pesquisa, apoio ao ensino e aprendizagem, promoção da cultura e a leitura, manutenção e preservação do acervo bibliotecário, organização das informações dos visitantes em prol do monitoramento do estoque e entre outros.
- Eficiência: Com a criação do sistema, busca-se melhorar a efetividade na prestação de serviços aos usuários.
- Gestão das Informações:
 - Catalogação e Classificação: De forma adequada, permitindo que os usuários busquem os materiais desejados com facilidade, que saibam de forma transparente a disponibilidade e para que seja possível manter controle de todos os exemplares.
- Padrões e Normas:
 - Padrão de Catalogação: Os códigos dos livros devem ser gerados de forma padrão e distinta para garantir consistência de descrição e busca.
- Gestão de Usuários:
 - Autenticação e Permissões: Permitir a autenticação dos usuários e estabelecer níveis distintos de permissão de acesso em prol da segurança e privacidade individual.
 - Segurança de dados: A partir de criptografia de informações pessoais.
 - Ética na obtenção de dados: Apenas informações relevantes serão solicitadas.

Para que o sistema seja construído e atenda todas as exigências acima, ele dependerá da criação e gerenciamento de um banco de dados com a finalidade de armazenar os dados de todos os livros, usuários e administradores registrados. O gerenciamento inclui alteração e organização das informações, a criação inclui o cadastro de todos os tipos de variáveis. Ademais, deve-se ter a possibilidade de excluir informações do banco, movendo-as para um backup temporário antes da remoção definitiva.

Ademais, os administradores são os usuários principais do sistema, eles são os responsáveis pelas informações do acervo e empréstimo cadastrando e registrando todas as informações, já os usuários são responsáveis apenas pelas suas próprias contas e pela solicitação de empréstimos. Dentre suas funcionalidades estão a criação de perfis para usuários e administradores, manutenção do acervo, empréstimo e devolução de exemplares, e dentre outras funcionalidades relacionadas.

Tendo isto em mente, para a criação deste projeto será utilizada a linguagem de programação Java (Oracle Corporation, 2023), o PostgreSQL (PostgreSQL Global Development Group, 2023) para a criação e hospedagem do banco de dados. Elas são duas tecnologias amplamente utilizadas no desenvolvimento de aplicações de software. Java é uma linguagem

de programação versátil e robusta que é conhecida por sua portabilidade, o que significa que as aplicações Java podem ser executadas em diferentes plataformas sem modificações significativas. Por outro lado, o PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto que oferece recursos avançados de armazenamento e consulta de dados, vale destacar que o sistema de banco de dados vai ser mantido com o auxílio do ElephantSQL (ElephantSQL, 2023), que oferece uma solução hospedada e gerenciada para PostgreSQL, tornando mais fácil para desenvolvedores e empresas criar, implantar e manter bancos de dados PostgreSQL na nuvem, sem a necessidade de se preocupar com a configuração de servidores, escalabilidade, segurança e tarefas de manutenção.

A combinação de Java e PostgreSQL com o ElephantSQL foi escolhida também com o objetivo de criar um sistema confiável, permitindo que os dados sejam armazenados, acessados e manipulados de maneira eficiente e segura. Para os testes, foi escolhido executar-los à moda antiga: Testes manuais. Utilizando o conceito de caixa cinza.

Testes de caixa cinza, segundo Spillner (SPILLNER; LINZ, 2021) são uma técnica de teste de software que combina elementos de testes de caixa preta e testes de caixa branca. Nesse tipo de teste, os testadores têm algum conhecimento parcial da estrutura interna do sistema, como o design ou a arquitetura do software, além de informações sobre os requisitos funcionais. Isso permite que eles projetem casos de teste mais direcionados, considerando as especificações de entrada e saída, mas também levando em conta a lógica interna do sistema.

III. TRABALHOS RELACIONADOS

O primeiro trabalho a ser destacado é o OpenLibris, desenvolvido por Márcia Goretti Ribeiro Grossi (GROSSI, 2008) para sua tese de doutorado. Este sistema, segundo sua desenvolvedora, foi criado por meio de programação para internet e foi baseado nos sistemas mais utilizados pelas universidades, dando destaque ao Pergamum, que será citado posteriormente. Após consultar 52 bibliotecas universitárias e receber informações de 31 delas, ela buscou implementar o melhor de cada sistema no seu próprio projeto: Funções de automação das rotinas administrativas e técnicas da biblioteca, serviços de referência digital síncronos e assíncronos e o maior motivador para o desenvolvimento do SGAB: A possibilidade de desenvolver um software completo usando apenas soluções livres.

Os pontos principais do seu projeto são: Preservação de registros da informação, organização e disseminação da informação. Que se alinham perfeitamente ao SGAB. Entretanto, temos uma pequena vantagem, a possibilidade do SGAB funcionar sem acesso a rede, ou seja, diferente do projeto de Goretti, o SGAB poderá armazenar suas informações localmente com a troca de credenciais da conexão do banco, ademais, pode-se fazer o salvamento de alterações localmente para depois serem enviados quando a conexão voltasse.

O segundo trabalho a ser destacado é o Pergamum (PUCPR, 2023), uma empresa que oferta vários sistemas de gerenciamento de informações, destes, podemos destacar: Eventum, para eventos institucionais, Archivum, par documentações acadêmicas, Analysis, para avaliações institucionais e Biblioteca, que gerencia acervos bibliotecários. Ele já está no mercado há vinte anos, e é utilizado também para arquivos e museus. Seus serviços são: Catalogação, aquisição, controle de usuários, circulação de materiais, emissão de relatórios e consulta ao catálogo. Basicamente, tudo que o SGAB tem como objetivo ofertar.

A vantagem do SGAB em relação ao Pergamum seria o desenvolvimento usando softwares livres e, também, ser uma opção econômica financeiramente. O Pergamum, segundo o site do GOV (GOV, 2019), custa 629,40 reais ao mês incluindo manutenção e suporte. No nosso caso, o produto teria uma licença vitalícia, onde um único pagamento garante acesso ao programa e as suas atualizações com o uso de um banco local, tornando-se uma opção econômica e acessível para instituições menores, e uma assinatura mensal à parte para manter o banco de dados online.

IV. FUNCIONALIDADES

Para facilitar o entendimento deste tópico, iremos listar as funções que estão no diagrama de classes e foram implementadas e falar sobre cada uma delas individualmente:

- O banco de dados foi alterado 8 vezes durante todo o desenvolvimento, em todas elas foi necessário atualizar as funções de todos os objetos.
- PostgreSQLConnection: Responsável pela conexão com o banco de dados. Possui as informações de login com o banco, as funções de conectar, fechar o banco e sincronização.
- Menu: Ele é quem permite a utilização correta de todas as funções abaixo, fazendo com que a administração da biblioteca e o empréstimo de livros seja executado de forma dinâmica.
- Adm - CriarAdm: Esta função exige como parâmetro todas as informações do administrador para serem registradas e não possui retorno. Primeiramente, ela tenta se conectar ao banco. Se for possível se conectar, cria uma query (linha em sql que será executada no banco de dados para armazenar as informações) destinada a inserir na tabela de administrador os seus novos itens. Depois, cria-se um state que é uma declaração que vai ser executada no banco, que irá receber a query. Após a substituição das informações, o state é executado no banco e o administrador está registrado no banco de dados. Esta função não apresentou desafios para o grupo de desenvolvedores.
- Adm - BuscarAdm: Esta função exige o id do administrador, através desse id é possível retornar o objeto administrador após uma busca no banco de dados. Esta função não apresentou desafios para o grupo de desenvolvedores.
- Adm - ExcluirAdm: Esta função exige o id do administrador, através dele, é possível executar uma declaração para excluir o administrador, não possui retorno. Esta função não apresentou desafios para o grupo.
- Adm - ListaAdm: Uma função sem retorno que retorna um arraylist de todos os administradores. Esta função foi desafiadora graças ao arraylist, que foi bem difícil de preencher.
- Adm - EditarAdm: Esta função recebe todas as informações do adm e executa um update na tabela de administradores, atualizando o administrador com base no seu cpf, não possui retorno. Esta função apresentou uma dificuldade; Como editar os telefones? A solução foi criar um editar especial para os telefones, armazenado no usuário.
- Usuário - CriarConta: Basicamente a mesma coisa do criarAdm, a diferença é a falta do idAdm, variável importante para a confirmação de empréstimos a usuários padrão. É uma função sem retorno e não apresentou dificuldades em sua criação.
- Usuário - ExcluirConta: Mesma coisa que a excluirAdm, mas ela funciona com base no cpf do usuário. Sem retorno, não apresentou dificuldades em sua criação.
- Usuário - BuscaUsuário: Retorna o usuário usando como parâmetro o seu cpf. Não apresentou dificuldades em sua criação.
- Usuário - BuscaTelefone: Retorna um resultset (funciona como um arraylist, é um retorno de tabela usada em banco de dados) com todos os telefones registrados em um cpf. A dificuldade foi a de entender o funcionamento do resultset, entretanto, foi mais eficiente que o uso de um arraylist, que foi sugerido anteriormente.
- Usuário - LoginUsuário: Basicamente, busca uma correspondência numa mesma linha de um email e uma senha, se for possível, retorna o usuário. É utilizada para login na plataforma, não apresentou dificuldades na sua criação.
- Usuário - EditarUsuario: Mesma coisa de editarAdm, e mesma dificuldade.
- Usuário - EditarTelefone: Solução na dificuldade encontrada no editarAdm e editarUsuario. Usando o cpf do usuário, ela busca quais telefones são dele na tabela, os apresenta para o usuário escolher qual quer editar e edita o número correto com base no id dele. Foi confusa de se elaborar pela equipe, demandando tempo para a elaboração da lógica e aplicação.
- Usuário - ListaUsuario: Lista todos os usuários registrados, uma função fácil de ser implementada.
- Empréstimo - BuscaEmprestimoCpf: Usa o cpf do usuário como parâmetro de busca, retornando todos os empréstimos com aquele cpf. A dificuldade foi organizar o retorno em um arraylist.
- Empréstimo - ListaEmprestimoCpf: Lista todos os empréstimos de um cpf, uma função fácil de ser implementada.
- Empréstimo - BuscaEmprestimoId: Usa o id do empréstimo para retornar-lo. Sem dificuldades para a equipe.
- Empréstimo - InserirEmprestimo: Registra um

empréstimo no banco de dados, como as funções de criar contas. Uma função análoga se torna fácil de ser implementada.

- Empréstimo - DevolverLivro: Edita o empréstimo com a data de devolução.
- Empréstimo - ListarAtrasados: Faz uma busca no banco e retorna uma lista com todos os livros que deveriam ter sido devolvidos.
- Livro - CadastrarLivro: Insere um livro no banco de dados, vale lembrar que possui a quantidade de livros, então apenas uma cópia é registrada para evitar redundância.
- Livro - BuscaLivroTitulo, BuscaLivroAutor e BuscaLivroGenero: Uma busca que retorna todas as correspondências na pesquisa de um título, autor ou gênero: “Menin” pode retornar “A menina e o porquinho” ou “O menino do pijama listrado”.
- Livro - EditarLivro: Atualiza as informações de um livro no banco de dados.
- Livro - ListarAcervo: A função lista todos os livros no acervo usando a pesquisa “Select * from Livro”.

Além do que já foi citado acima, os maior obstáculo, de longe, foi a elaboração da estrutura do banco de dados e auxiliar a mesma com a aplicação java, tanto que o diagrama de classes e o banco foram alterados diversas vezes. Demandando mais tempo para a criação correta de cada função, que foram reescritas várias vezes até a sua versão final. As funcionalidades mais desafiadoras envolveram as funções que retornam listas.

E, principalmente, a demanda geral que cada integrante possuía de outras cadeiras curriculares, unindo a dificuldade de criar reuniões foi desafiador. Mas, mesmo assim, conseguimos cumprir com todas as demandas acima.

V. TESTES E RESULTADOS

Inicialmente, os testes foram de caixa cinza. A estratégia visa examinar tanto a funcionalidade geral do sistema como sua estrutura interna e implementação. O motivo principal foi a existência do conhecimento sobre o código e que o mesmo foi testado na visão de um usuário, baseando-se nos estudos de Spillner, . (SPILLNER; LINZ, 2021).

Os testes se concentraram, primeiramente, na validação de todas as entradas de informações para garantir a execução correta das funções implementadas. O sistema é responsável por não deixar o usuário errar.

Eles foram baseados no grafo causa-efeito da função principal do programa, que orchestra as demais funções e pode ser consultado nos anexos. Ademais, após sua execução, as possíveis entradas, as possíveis saídas (oráculo) e as saídas reais foram registradas em tabelas de causa-efeito, que também estão no final deste documento.

Por fim, documentado em vídeo, testaram-se todas as funcionalidades através da execução do código, identificando seus caminhos e verificando cada um deles. Vale ressaltar que o programa não possui loops, apenas caminhos multifacetados, e sua estrutura garante que apenas dados com o tipo e formato corretos sejam aceitos pelas funções.

Os testes de caminhos seguiram a estrutura do grafo, e os oráculos foram usados para definir os resultados esperados. Para cada discrepância entre os resultados reais e os esperados um ajuste foi feito de imediato, gerando o código atual. Vale salientar que se a função com problemas for independente, os testes foram focados apenas nela. Se for uma função dependente, revisamos também as funções que a utilizam.

Por fim, os testes abrangeram tanto o fluxo de controle quanto o fluxo de dados, garantindo uma cobertura abrangente. Isso é fundamental para assegurar que os caminhos estejam sendo percorridos de maneira adequada e que as modificações no banco de dados sejam registradas devidamente. Vale salientar que não foram feitos testes combinacionais, pois o sistema possui funções de tratamento que garantem que só informações corretas serão adicionadas.

Depois de todos os testes que foram executados, foi possível observar que todos alcançaram o seu objetivo, ou seja, corresponderam ao oráculo.

	Menu Principal	Partições	T1	T2	T3	T4
Entrada	Menu Login	Ok	X			
	Menu Cadastrar	Ok		X		
	Sair	Ok			X	
	Outro caso	Ok				X
Saídas	Menu Login	-	X			
	Menu Cadastro	-		X		
	Fechar o Programa	-			X	
	Fazer Nada	-				X
Resultado:			Ok	Ok	Ok	Ok

Figura 1. Tabela Menu Principal.

	Menu Login	Partições	T1	T2	T3	T4	T5	T6
Entrada	Email	Ok	X		X		X	X
	Senha	Ok		X		X	X	X
		Sim	X	X				
	Mais de 3 tentativas?	Não			X	X	X	X
		Sim					X	
	É adm?	Não						X
Saídas	Menu Principal	-	X	X				
	Menu ADM	-					X	
	Menu Login	-						X
	Tentar Novamente	-			X	X		
Resultado:			Ok	Ok	Ok	Ok	Ok	Ok

Figura 2. Tabela Menu Login.

	Menu Cadastro	Partições	T1	T2	T3	T4	T5	T6	T7
Entrada	Nome	Ok		X	X	X	X	X	
	CPF	Ok			X	X	X	X	
	Email	Ok				X	X	X	
	Senha	Ok					X	X	
	Telefone	Ok						X	
	Sair	Ok							X
	Tentar Novamente	-	X	X	X	X	X		
Saídas	criarConta()	-						X	
	menu Principal	-							X
Resultado:			Ok	Ok	Ok	Ok	Ok	Ok	Ok

Figura 3. Tabela Menu Cadastro.

	Menu Usuário	Partições	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
Entrada	Perfil	Ok	X	X	X								
	Pesquisar Livros	Ok				X	X						
	Pedir Empréstimo	Ok						X	X	X			
	Listar Empréstimos	Ok									X		
	Sair	Ok										X	X
	Editar?	Sim	X	X									
		Não			X								
Saída	Livro Corresponde?	Sim				X		X	X				
		Não								X			
	Confirmar?	Não		X					X				X
	Editar Perfil	-	X										
	Voltar ao Menu Usuário	-		X	X	X	X	X	X	X	X		X
	Voltar ao Menu Principal	-										X	
	Exibir Livro Escolhido	-				X							
Resultado:	Exibir Empréstimos	-									X		
	Registrar Empréstimo	-						X					

Figura 4. Tabela Menu Usuário.

	Menu ADM	Partições	T1	T2	T3	T4	T5	T6
Entrada	Exibir Perfil	Ok	X	X	X			
	Gerenciar Usuários	Ok				X		
	Gerenciar Acervo	Ok					X	
	Gerenciar Empréstimos	Ok						X
	Sair	Ok						
	Editar?	Sim	X	X				
		Não			X			
Saída	Confirmar?	Sim	X					
		Não		X				
	Editar Usuário	-	X					
	Menu Gerenciar Usuário	-				X		
	Menu Gerenciar Acervo	-					X	
	Menu Gerenciar Empr	-						X
	Voltar ao Menu ADM	-						
Resultado:			X	X	X			
			Ok	Ok	Ok	Ok	Ok	Ok

Figura 5. Tabela Menu ADM.

	Menu Empréstimo	Partições	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Entrada	Receber Livro	Ok	X	X	X							
	Extender Prazo	Ok				X	X	X				
	Listar Por Usuários	Ok							X			
	Listar Atrasados	Ok								X		
	Listar Empréstimos	Ok									X	
	Sair	Ok										X
	Pesquisa Corresponde?	Sim	X	X		X	X					
Saída		Não			X			X				
	Confirmar?	Sim	X	X		X	X	X	X			
		Não		X								
	Menu Empréstimo	-	X	X	X	X	X	X	X			
	Exibir Perfil	-										
	devolverLivro()	-	X									
	extenderPrazo()	-				X						
Resultado:	listarUsuários()	-						X				
	listarEmpréstimos()	-								X		
Resultado:	listarAtrasados()	-									X	
	Menu ADM	-										X

Figura 6. Tabela Menu Empréstimo.

	Menu Acervo	Partições	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
Entrada	Cadastrar	Ok	X	X	X									
	Buscar	Ok				X	X							
	Listar	Ok						X						
	Editar	Ok							X	X				
	Excluir	Ok									X	X	X	
	Voltar	Ok												X
	Pesquisa Corresponde?	Sim	X	X		X		X		X	X			
Saída		Não			X				X			X		
	Confirmar?	Sim	X					X		X				
		Não		X							X			
	Acervo Menu	-		X	X	X	X	X	X	X	X	X	X	
	cadastrar()	-	X											
	buscar()	-				X								
	listar()	-						X						
Resultado:	editar()	-							X					
	excluir()	-									X		X	
Resultado:	Menu ADM	-												X
			Ok	Ok	Ok	Ok	Ok	Ok	Ok	Ok	Ok	Ok	Ok	Ok

Figura 7. Tabela Menu Acervo.

VI. CONCLUSÃO

Com base neste estudo e no desenvolvimento deste sistema, foi possível explorar a concepção e desenvolvimento

do Sistema de Gerenciamento de Acervos Bibliotecários (SGAB). O SGAB representa uma solução promissora para a documentação, organização e supervisão eficaz das coleções de bibliotecas, tanto públicas quanto privadas.

O projeto se baseia na combinação de tecnologia, alocação de recursos e planejamento estratégico para harmonizar os processos do mundo real com registros virtuais, aprimorando assim a organização dos acervos bibliotecários. A escolha das tecnologias, como a linguagem de programação Java e o sistema de banco de dados PostgreSQL, justificou-se pela capacidade de criar um sistema confiável, seguro e verificado para o armazenamento e manipulação de dados de bibliotecas.

A realização de testes sistemáticos, desde os estágios iniciais de desenvolvimento até as fases finais de implementação, permitiu identificar e corrigir potenciais problemas, erros de código e discrepâncias nos dados. Os testes realizados no sistema incluíram validação de entradas de dados, testes de caminhos, verificações de fluxo de controle e de dados. Os testes foram executados para garantir que o sistema funcionasse conforme o esperado, identificando discrepâncias e fazendo ajustes quando necessário.

Nesse contexto, as estratégias de teste adotadas foram fundamentais para mitigar riscos e assegurar que o SGAB fosse lançado com sucesso, atendendo às demandas e normas de bibliotecas de maneira robusta e confiável.

Em resumo, os testes desempenharam um papel crucial na concepção, desenvolvimento e aprimoramento do SGAB. Eles garantiram que o sistema fosse uma solução sólida e confiável para as instituições bibliotecárias, contribuindo para o cumprimento de seus objetivos de gestão e proporcionando uma experiência de usuário satisfatória.

Considerando o desenvolvimento contínuo do Sistema de Gerenciamento de Acervos Bibliotecários (SGAB), há várias oportunidades de aprimoramento e expansão que podem ser consideradas para trabalhos futuros. Primeiramente, a implementação de criptografia dos dados dos usuários é uma medida crítica para fortalecer a segurança e a privacidade, tornando o SGAB mais resistente a ameaças cibernéticas. Além disso, a criação de uma interface de usuário intuitiva e amigável aprimoraria a experiência dos usuários, facilitando a navegação e a interação com o sistema. A implementação de mecanismos de penalização para usuários que não cumprem com as devoluções de livros pode promover maior responsabilidade e reduzir perdas de acervos. Além disso, tornar o SGAB multiplataforma, de modo que possa ser acessado em diversos dispositivos e sistemas operacionais, ampliaria sua acessibilidade. Por fim, a incorporação de testes automatizados contínuos garantirá a estabilidade e a confiabilidade do sistema, facilitando a identificação precoce de problemas e melhorias na fase de desenvolvimento. Essas sugestões representam áreas-chave para futuras pesquisas e desenvolvimento no contexto do SGAB.

Portanto, o SGAB representa uma solução econômica e acessível para instituições que desejam melhorar a gestão de suas bibliotecas. Com a capacidade de funcionar localmente e com a flexibilidade oferecida pelas tecnologias utilizadas,

o SGAB tem o potencial de simplificar a administração de acervos bibliotecários, melhorando a eficiência na prestação de serviços aos usuários e contribuindo para o alcance dos objetivos das bibliotecas.

VII. ANEXOS

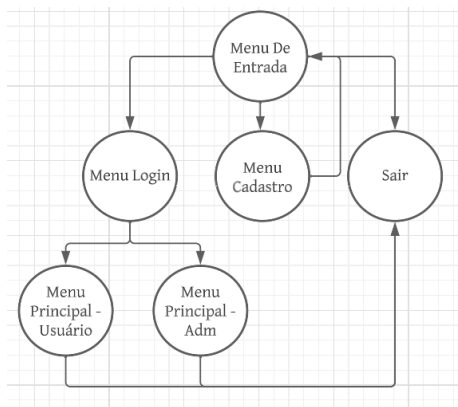


Figura 8. Grafo Principal.

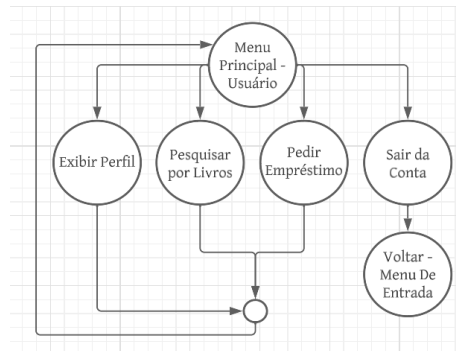


Figura 11. Menu Principal - Usuário.

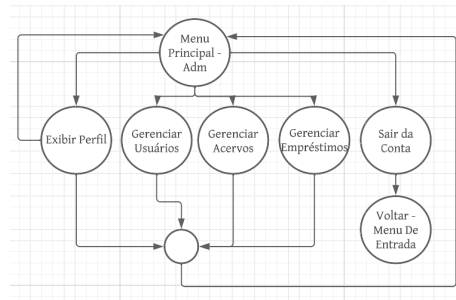


Figura 12. Menu Principal - ADM.

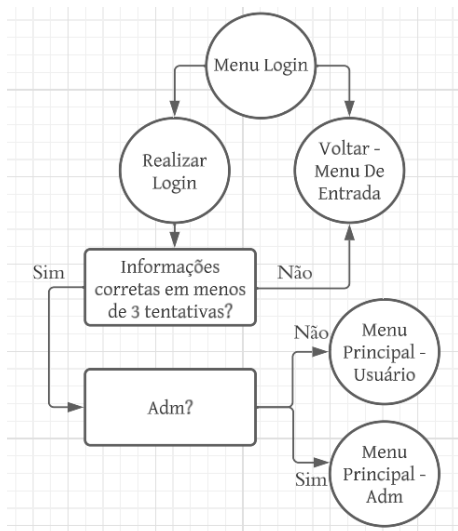


Figura 9. Menu Login.

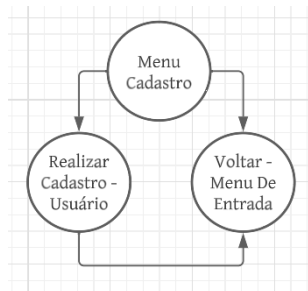


Figura 10. Menu Cadastro.

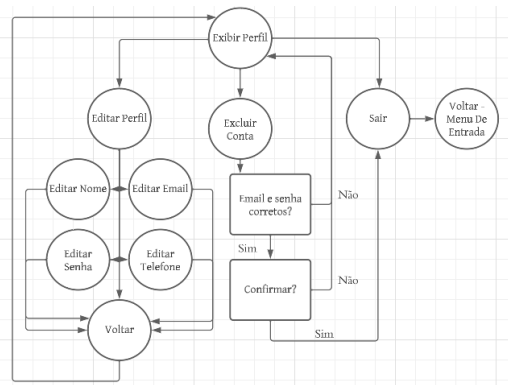


Figura 13. Exibir Perfil - Ambos.

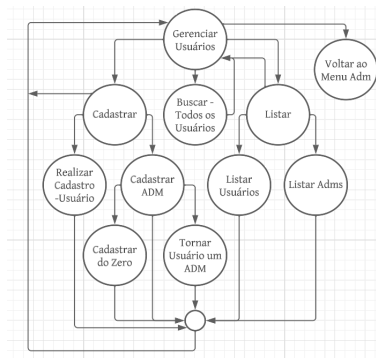


Figura 14. Gerenciar Usuários - ADM.

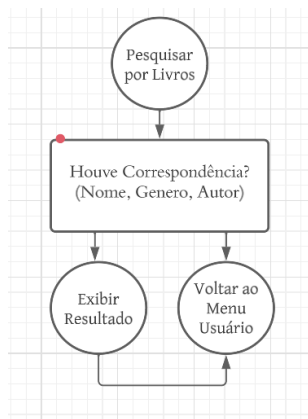


Figura 15. Pesquisa de Livros - ADM e Usuário.



Figura 16. Empréstimo - Ambos.

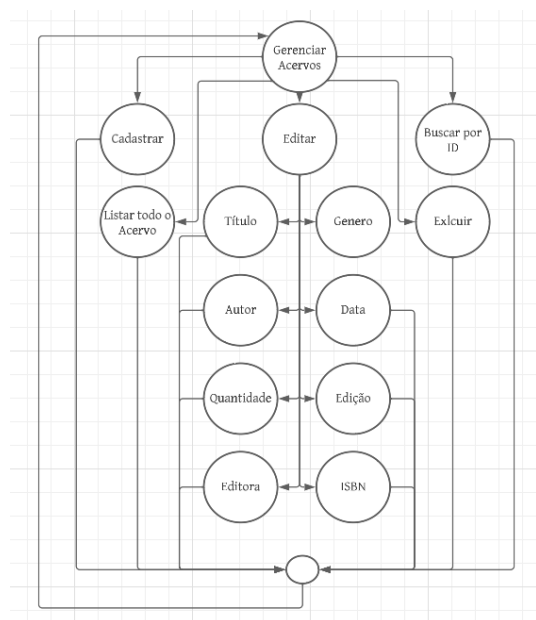


Figura 17. Gerenciar Acervo.

REFERÊNCIAS

- DAFT, R. L. **Management**. [S.l.]: Cengage Learning, 2016.
- ElephantSQL. **ElephantSQL - PostgreSQL as a Service**. 2023. Disponível em: <https://www.elephantsql.com/>.
- GOV. **Contrato de Manutenção do Sistema Integrado de Bibliotecas - Pergamum**. 2019. Disponível em: https://www.gov.br/observatorio/pt-br/aceso-a-informacao/licitacoes-e-contratos/contratos/2019/documentos/contrato_08-2019-pergamum.pdf.

[//www.gov.br/observatorio/pt-br/aceso-a-informacao/licitacoes-e-contratos/contratos/2019/documentos/contrato_08-2019-pergamum.pdf](https://www.gov.br/observatorio/pt-br/aceso-a-informacao/licitacoes-e-contratos/contratos/2019/documentos/contrato_08-2019-pergamum.pdf).

GROSSI, M. G. R. Estudo das características de software e implementação de um software livre para o sistema de gerenciamento de bibliotecas universitárias federais brasileiras. Universidade Federal de Minas Gerais, 2008.

Oracle Corporation. **Java**. 2023. Disponível em: <https://www.java.com/>.

PostgreSQL Global Development Group. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2023. Disponível em: <https://www.postgresql.org/>.

PUCPR. **Pergamum - Sistema Integrado de Bibliotecas**. 2023. Disponível em: <https://www.pergamum.pucpr.br/>.

SPELLNER, A.; LINZ, T. **Software Testing Foundations: A Study Guide for the Certified Tester Exam-Foundation Level-ISTQB® Compliant**. [S.l.]: dpunkt. verlag, 2021.