Universidade Federal Rural do Semi-Árido
Campus Pau dos Ferros
Departamento de Engenharias e Tecnologia

FRANCISCA LORRAYNE DE LIMA SANTOS
FRANCISCA SAMIRA AQUINO FRANÇA
PEDRO VINICIUS DE ANDRADE QUEIROZ
POLLYANA DIAS PAIVA

Sistema de Gerenciamento de Acervos Bibliotecários - “SGAB”

Pau dos Ferros - RN
OUTUBRO - 2023

FRANCISCA LORRAYNE DE LIMA SANTOS
FRANCISCA SAMIRA AQUINO FRANÇA
PEDRO VINICIUS DE ANDRADE QUEIROZ
POLLYANA DIAS PAIVA

Sistema de Gerenciamento de Acervos Bibliotecários - “SGAB”

Relatório apresentado à Universidade Federal Rural do Semi-Árido, na disciplina de Banco de Dados, como requisito para avaliação na 3ª unidade, solicitado pela professora Dra. Huliane Medeiros da Silva.

Pau dos Ferros - RN
OUTUBRO - 2023

SUMÁRIO

1. Introdução

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

Nosso 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.

O SGAB foi concebido para ser uma ferramenta acessível e confiável, pensada tanto para os usuários que desejam buscar otimizar a gestão dos recursos de suas bibliotecas. O objetivo é tornar a jornada de pesquisa, aprendizado, manutenção e preservação das ferramentas fundamentais para o conhecimento de forma eficiente e agradável para todos os envolvidos.

As etapas do projeto foram projeção e desenvolvimento. Na etapa de projeção, a equipe se dedicou a elaboração da lógica geral do SGAB, realizando pesquisas gerais de como os sistemas de gerenciamento funcionam, como adequar as bibliotecas e criando os diagramas com os dados a serem implementados. A etapa de desenvolvimento, que é a mais complexa, focamos na conexão com o banco de dados, criação dos menus, objetos, métodos e funcionalidades.

1.1. Equipe de desenvolvimento

FRANCISCA LORRAYNE DE LIMA SANTOS - 2021022429
FRANCISCA SAMIRA AQUINO FRANÇA - 2021022568
PEDRO VINICIUS DE ANDRADE QUEIROZ - 2021010775
POLLYANA DIAS PAIVA - 2021011138

2. Desenvolvimento

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.

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.

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:

Objetivos:

- 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, o PostgreSQL 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 uma plataforma de código aberto para gerenciamento de banco de dados relacional que disponibiliza recursos avançados para armazenamento e consulta de informações.

A combinação de Java e PostgreSQL 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.

Inicialmente, foi criado um diagrama de classes básico para criar a idéia geral de como seria a lógica dos relacionamentos e quais informações seriam implementadas em cada classe criada. Vale salientar que após sua elaboração os objetos, getters, setters, toStrings e construtores foram criados.

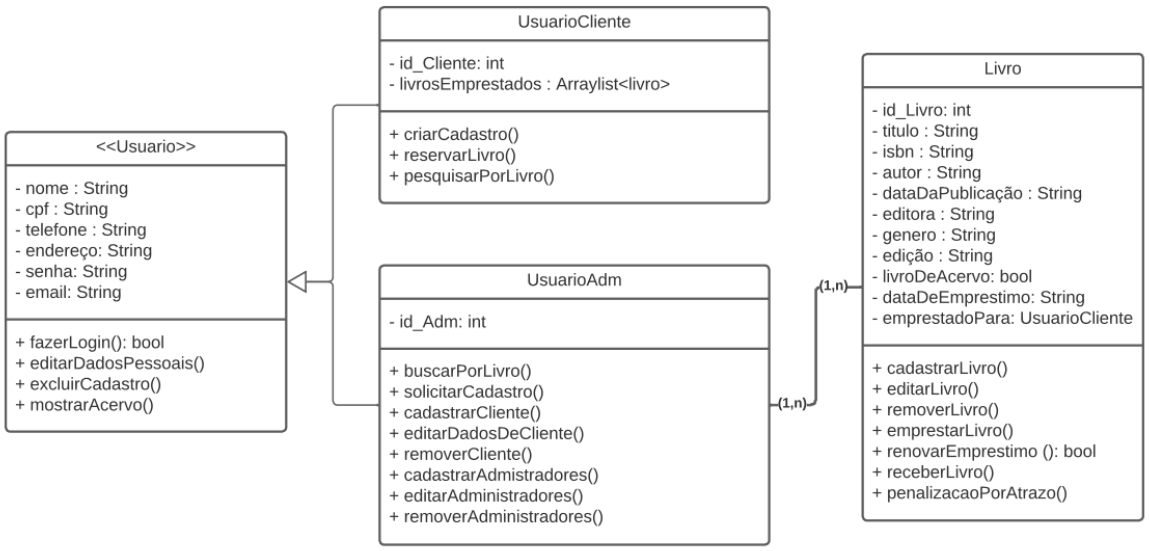


Figura 1 - Primeiro Diagrama de Classes

Em seguida, nos deparamos com o impasse principal: Como implementar o servidor? No dia 27 de Setembro, usando o site ElephantSQL, criamos uma instância gratuita para implementar um banco de dados em PostgreSQL. Ele possui três vantagens que foram os pontos principais para a sua escolha: Ele está disponível para a maioria das aplicações em nuvem ou em plataformas, armazena em tempo real as informações adicionadas, alteradas ou excluídas no banco em nuvem e possui backups automáticos, houve certa dificuldade, mas em apenas um dia foi possível implementar e elaborar o banco de dados.

Nesse contexto, começamos a estruturar o Banco de Dados para o Sistema de Gerenciamento de Acervos Bibliotecários (SGAB). Partindo deste pressuposto, iniciamos a compreensão das necessidades dos usuários e administradores do sistema, bem como pelo desenvolvimento de um plano de negócios que fornece diretrizes sólidas para o projeto. No estágio de planejamento, começamos a conceber o Diagrama de Entidade-Relacionamento

(DER) para representar as entidades que serão armazenadas no banco de dados. Entidades estas que chamamos de: "Usuário", "Administrador", "Livro" e "Empréstimo" e “Cliente”, onde cada um possui seus atributos específicos, como "CPF", "Nome", "Senha", "Email", "Data de Devolução" e outros campos pertinentes.

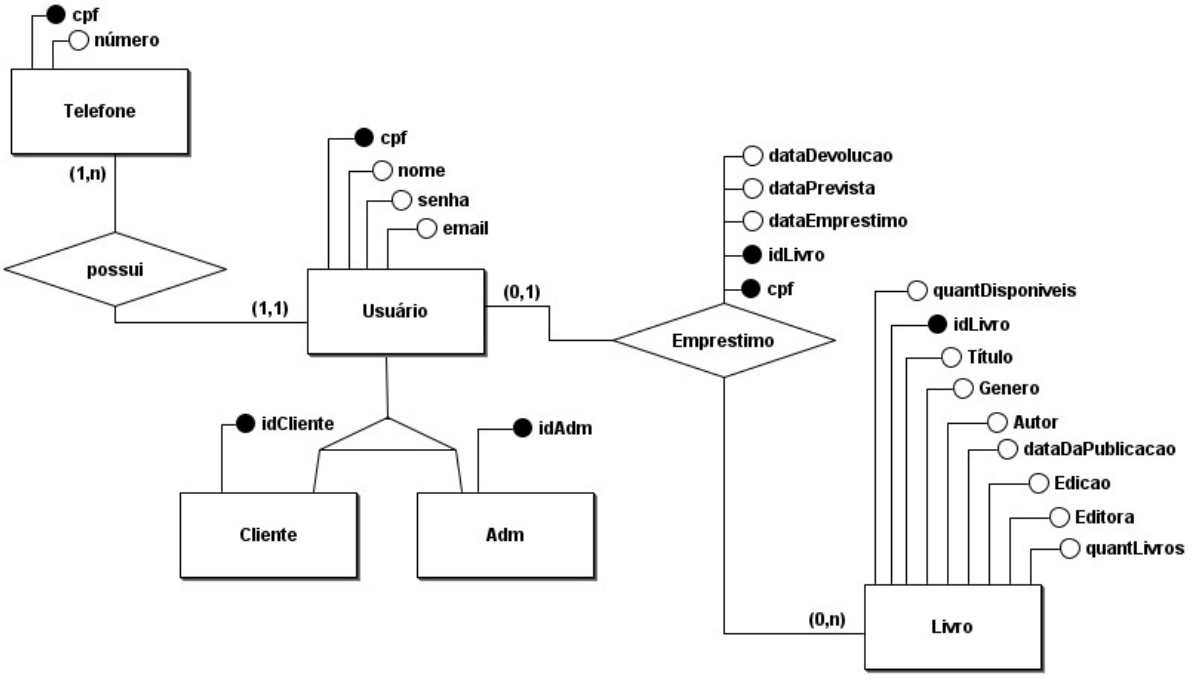


Figura 1 - de Entidade-Relacionamento (DER)

A criação do DER representou o ponto de partida para o desenvolvimento do projeto. No entanto, ao longo da elaboração do plano de negócios, percebemos que havia uma redundância entre as entidades "Usuário" e "Cliente", já que ambas desempenham a mesma função no sistema. Em resposta, optamos por eliminar a entidade "Cliente", o que demandou ajustes no DER em oito ocasiões para refletir essa decisão.

Adicionalmente, foi necessário efetuar a reclassificação da entidade "telefone" como um atributo do usuário, uma vez que a sua utilização como entidade não se mostrasse pertinente. Dessa forma, procedemos à sua inclusão como um atributo associado ao usuário, permitindo, assim, que os usuários possam fornecer suas informações de telefone durante o processo de cadastro, caso desejem. Após todas as alterações realizadas, o Diagrama de Entidade-Relacionamento (DER) evoluiu para a seguinte representação:

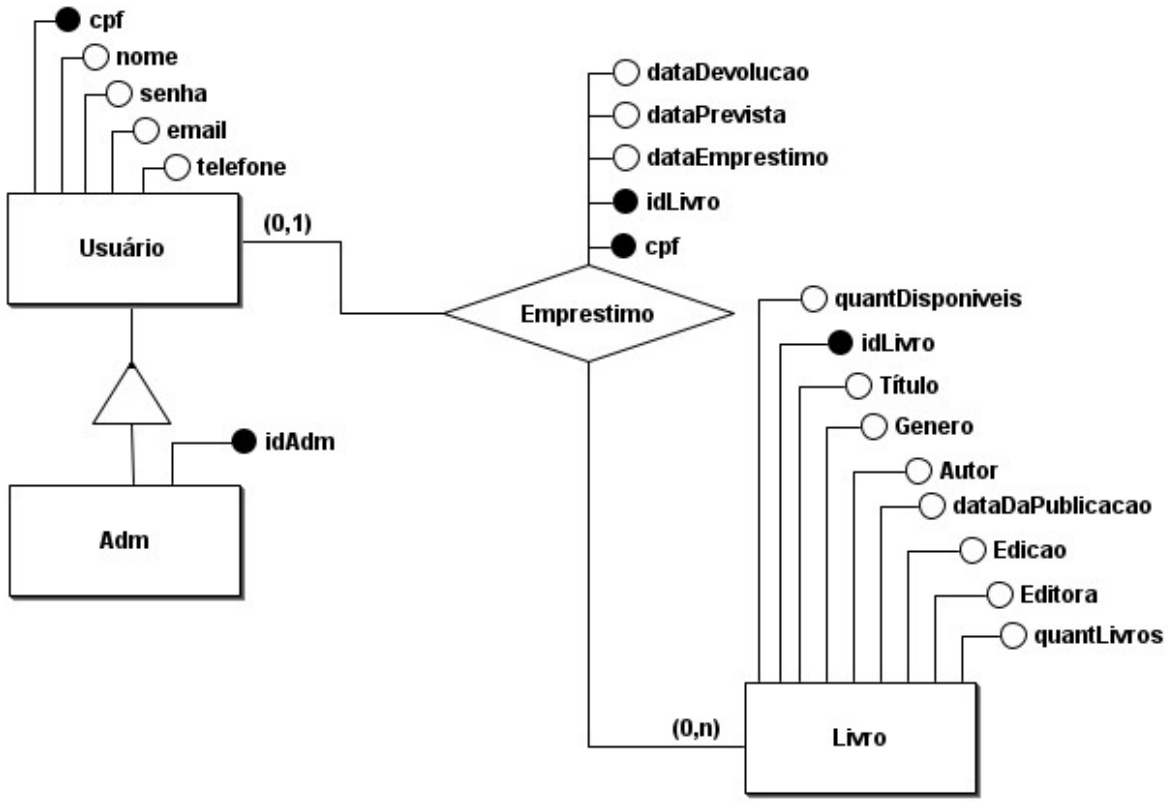


Figura 2 - Diagrama de Entidade-Relacionamento (DER) definitivo.

Além disso, foi de extrema importância estabelecer as relações entre essas entidades, uma vez que isso molda a lógica do sistema. Por exemplo, é importante especificar que um usuário pode solicitar apenas um empréstimo por vez, e esse empréstimo deve ser devolvido antes que o usuário possa fazer uma nova solicitação. Essas regras de negócios e restrições garantem a integridade e o funcionamento adequado do sistema.

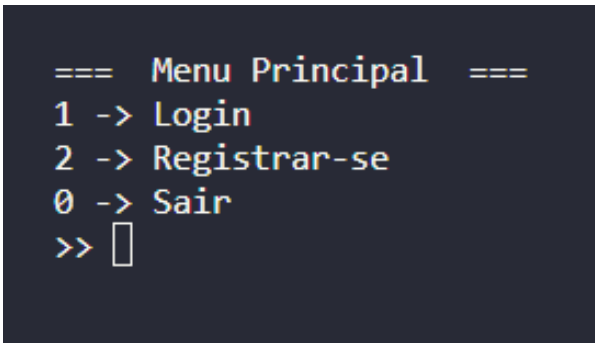
Após a elaboração e refinamento do DER conforme necessário, avançamos para a fase de implementação do banco de dados. Isso envolveu a criação das tabelas, a definição dos relacionamentos e a aplicação das regras de negócios para assegurar a funcionalidade adequada do sistema. Esse processo de desenvolvimento do SGAB visa atender de forma eficaz às necessidades de usuários e administradores da biblioteca, melhorando a gestão do acervo e simplificando o processo de empréstimo e devolução de livros.

Agora que o banco de dados e as conexões estavam prontas, no dia 28 de Setembro, iniciou-se o desenvolvimento das funções referentes ao livro, entre elas: buscarLivroId (exibe uma lista de livros com base no id inserido, apenas para administradores) e buscarLivro (exibe uma lista com base no título, autor ou gênero escolhidos, utilizado pelos usuários e administradores). Criar a busca foi desafiador por causa de problemas de sintaxe com o SQL,

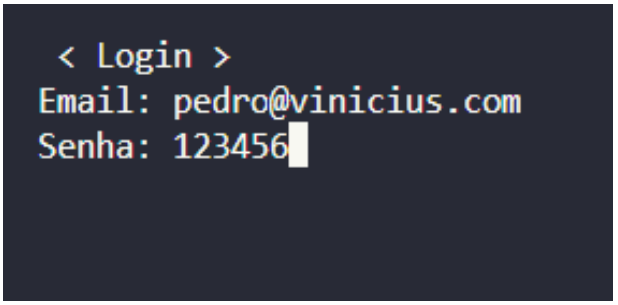
ou seja, não sabíamos que a substituição de uma interrogação por uma *string* numa *query* gerava “ ‘*string*’ ” ao invés de “*string*”. Quando descobrimos, no dia 29 de Setembro, foi possível executar as buscas no banco.

Segue abaixo as propostas de visão de menu:

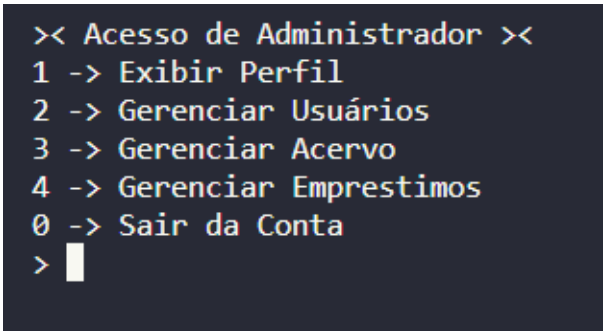
Tela principal do menu:

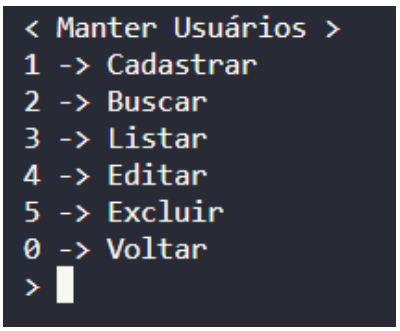


Tela de Login:

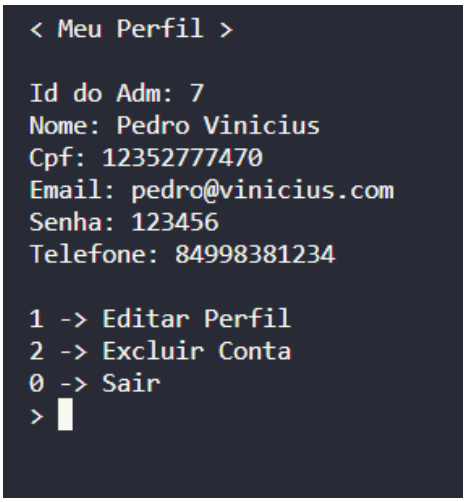


Tela do Administrador:

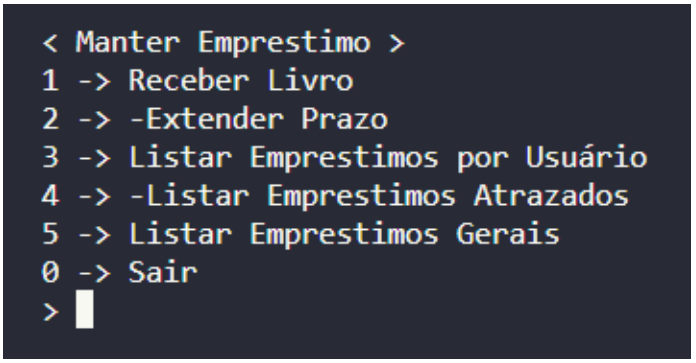




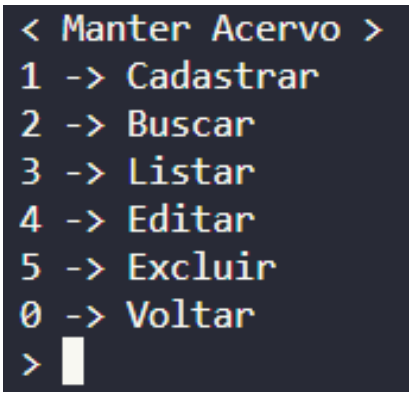
Tela de exibição dos dados do usuário/adm:



Tela de empréstimo de Livros:



Acervo:



Neste mesmo dia, começamos algumas funções referentes ao usuário, entretanto, surgiu-se a necessidade de reelaborar o diagrama de classes e o banco de dados. O motivo foi a necessidade de criar uma tabela especial para empréstimo, para que fosse possível gerenciar mais informações, como: Quantidade de livros disponíveis, livros que foram emprestados juntos para um usuário, documentação da data que o empréstimo foi criado, geração da data prevista de devolução e a data real da devolução para criar a penalização de um usuário que atrasou os livros e muitas outras modificações. Ademais, foi necessário reformular o banco de dados por completo, portanto, até o dia 30 de novembro, essas modificações foram feitas e estes são os resultados:

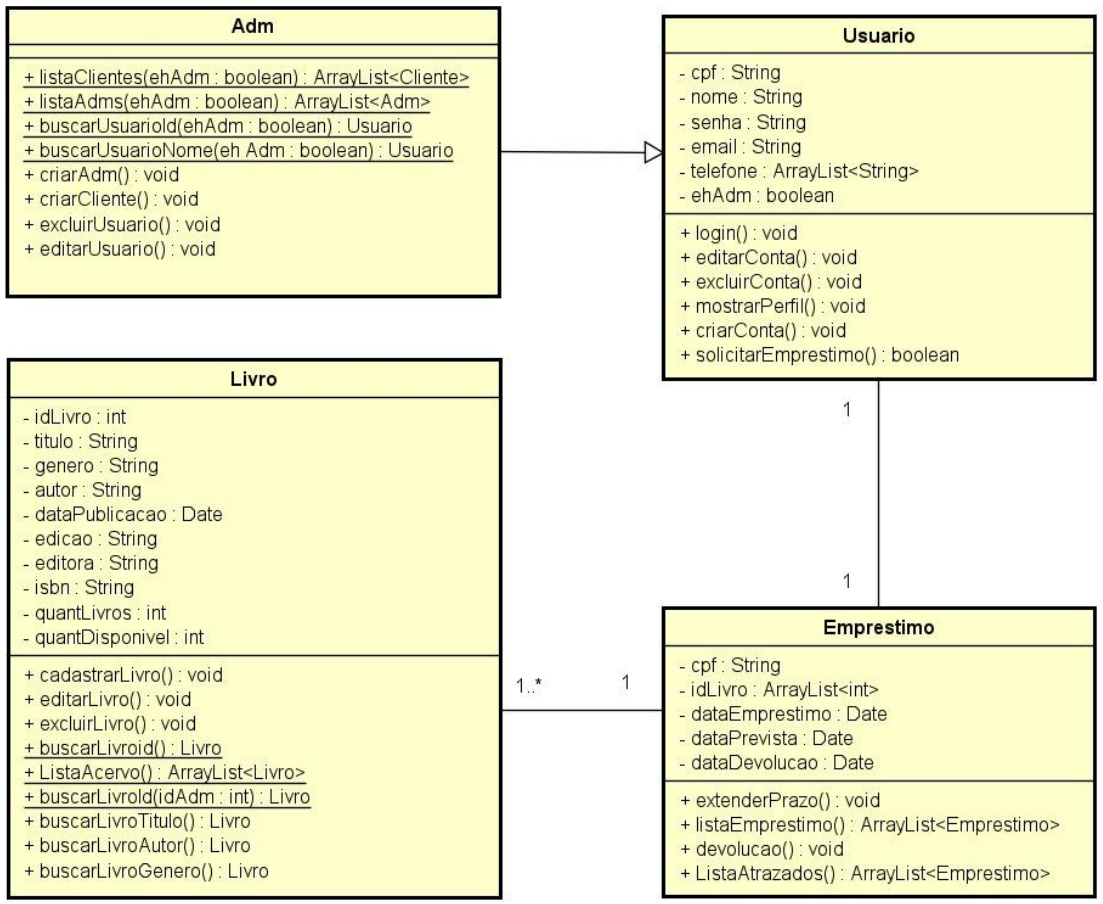


Figura 2 - Diagrama de Classes Definitivo

Após adaptar integralmente o código existente para a nova configuração, foi possível introduzir novas implementações a partir do dia 2 de outubro. Nesse período, foram desenvolvidas todas as funções relacionadas a empréstimos, bem como as funções compartilhadas entre os usuários, conforme o novo diagrama.

Após a conclusão dessas implementações, em 4 de outubro, deu-se início à criação das funções relacionadas ao administrador, à adição das últimas funcionalidades referentes aos livros e à elaboração do menu que possibilitará a utilização do programa.

Devido a restrições de tempo, optou-se por não desenvolver uma interface gráfica. Assim, o programa será acessado pelo prompt de comando. Contudo, é importante destacar que qualquer desktop com o Java devidamente instalado e configurado poderá acessar o Sistema de Gerenciamento de Acervos Bibliotecários (SGAB) e desfrutar de suas funcionalidades.

Em 5 de outubro, realizou-se uma reunião para correções finais no diagrama, no banco de dados e na criação de novas tabelas para listas que precisavam ser separadas de suas tabelas originais. Além disso, foram iniciados os testes e a finalização do menu. No dia 6, começou-se a criação das funções ainda pendentes, junto com a correção das funções existentes. Esse foi um esforço conjunto da equipe, dedicando-se a analisar e corrigir potenciais erros, a maioria relacionada à compatibilidade com o novo banco de dados, como era esperado diante das extensas modificações realizadas.

Na reta final, nos dias 17 e 18 de outubro, o foco esteve na conclusão do menu, na correção de eventuais erros, nas atualizações no GitHub, na inserção de comentários auxiliares em algumas funções e nas atualizações de documentação. Finalmente, o resultado foi entregue para análise e avaliação.

2.1. Funcionalidades implementadas

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 possuia de outras cadeiras curriculares, unindo a dificuldade de criar reuniões foi desafiador. Mas, mesmo assim, conseguimos cumprir com todas as demandas acima.

2.2. Funcionalidades não concluídas

Primeiramente, as funções que não foram implementadas foram:

- **Empréstimo -> ExtenderPrazo:** A função deveria alterar a data de devolução, aumentando em uma semana. Não foi desenvolvida por falta de tempo e pelas dificuldades encontradas na implementação das funções de busca.

Ademais, por falta de tempo, o menu não foi concluído. Mesmo que quase todas as funções do diagrama de classes e do DER tenham sido criadas, testadas e funcionando corretamente, não foi possível implementar no menu as funções referentes a empréstimo. Ou seja, o acesso as contas e os privilégios do administrador estão corretamente implementados, mas o sistema de cadastro, alteração e devolução de empréstimos não foi implementado. Ainda por causa do prazo, não foi possível fazer a geração dos relatórios de empréstimo, entretanto a busca de empréstimos e de atrasos foi criada corretamente, é ela quem permite imprimir os relatórios do banco. Vale salientar que nunca um empréstimo pode ser apagado do banco.

De fato, além de aprender a lidar com listas e com comandos do banco de dados, o maior desafio foi o tempo, dado que todos os integrantes estavam extremamente ocupados com demandas importantes de outras cadeiras, entretanto, isso não impediu a criação correta de todas as funções necessárias para que o sistema funcione corretamente.

3. Conclusão

Em resumo, o desenvolvimento do Sistema de Gerenciamento de Acervos Bibliotecários (SGAB) representou uma jornada repleta de desafios e ajustes significativos. Começamos escolhendo o PostgreSQL como nosso banco de dados, aproveitando o serviço ElephantSQL para criar uma instância gratuita, o que nos proporcionou benefícios cruciais, como disponibilidade em várias plataformas, armazenamento em tempo real e backups automáticos.

Enfrentamos desafios técnicos na implementação das funções de busca de livros, superando obstáculos de sintaxe SQL. Após solucionar essas questões, conseguimos desenvolver com sucesso as funções relacionadas aos livros.

No entanto, à medida que o projeto avançou, ficou claro que o tempo limitado foi um grande desafio, exigindo reescritas de funções e comprometendo a conclusão de algumas, como a elaboração completa do menu e funcionalidades relacionadas a empréstimos. Funcionalidades não concluídas, como Solicitar Empréstimo e Extender Prazo, refletem limitações temporais e desafios técnicos não superados. Apesar dos desafios, a equipe demonstrou resiliência e comprometimento, alcançando a criação correta das funções essenciais para o funcionamento do sistema. A principal lição aprendida transcende o domínio técnico e envolve a gestão eficaz do tempo e dos recursos, habilidades essenciais no ambiente de desenvolvimento de software.

Em conclusão, o desenvolvimento do SGAB foi uma jornada de aprendizado, colaboração e superação de obstáculos, resultando em um sistema que atende às necessidades de gerenciamento de acervos bibliotecários. A equipe enfrentou desafios técnicos e limitações de tempo, mas conseguiu implementar com sucesso as funcionalidades essenciais. O projeto oferece uma base sólida para futuras iterações e destaca a importância da gestão eficaz de recursos em projetos de software.