

**UMA ARQUITETURA DE SISTEMA DE EMPRÉSTIMOS DE MATERIAIS PARA A UFERSA**

***A MATERIAL LENDING SYSTEM ARCHITECTURE FOR UFERSA***

Emanuel Bruno Duarte de Morais<sup>1</sup>

Taísso Reni de Souza Melo<sup>2</sup>

Reudismam Rolim de Sousa<sup>3</sup>

Samara Martins Nascimento Gonçalves<sup>4</sup>

**RESUMO**

Com a crescente demanda por tecnologia, a Universidade Federal Rural do Semi-Árido (UFERSA), Campus Pau dos Ferros, busca modernizar sua experiência educacional. Dentre as necessidades, encontra-se a de gerir o empréstimo de material à comunidade acadêmica (e.g., caixas de som, projetores). Atualmente, esse processo é feito manualmente, o que é demorado e sujeito a erros. A proposta deste trabalho é o desenvolvimento de uma arquitetura para o sistema “Empréstimo Fácil UFERSA” (EFU), que visa otimizar o empréstimo de equipamentos da Universidade. Para isso, foi empregado o método pesquisa-ação, que emprega técnicas consagradas para melhorar uma prática. A técnica utilizada foi o projeto de visões arquiteturais de módulo, componente & conector e alocação. A arquitetura visa facilitar o gerenciamento de materiais, superando as limitações dos métodos tradicionais. A avaliação se baseou em cenários, que demonstraram a adequação dos elementos das visões para a implementação do sistema.

**Palavras-chave:** desenvolvimento de software; arquitetura de software; empréstimos.

**ABSTRACT**

With the growing demand for technology, the Universidade Federal Rural do Semi-Árido (UFERSA), Pau dos Ferros Campus, seeks to modernize its educational experience. Among the needs, they need to manage the loan of materials to the academic community (e.g., speakers, projectors). Currently, this process is done manually, which is time-consuming and prone to errors. This work proposes to develop an architecture for the “Empréstimo Fácil UFERSA” (EFU) system, which aims to optimize the loan of equipment from the University. For this, the action research method was employed, which employs established techniques to improve a practice. The technique used was the design of architectural views of modules, components & connectors, and allocation. The architecture aims to facilitate the management of materials, overcoming the limitations of traditional methods. The evaluation was based on scenarios, which demonstrated the suitability of the elements of the views for the implementation of the system.

---

<sup>1</sup> Emanuel Bruno Duarte de Morais – Graduando em Engenharia de Software – UFERSA – Pau dos Ferros.

<sup>2</sup> Taísso Reni de Souza Melo – Graduando em Engenharia de Software – UFERSA – Pau dos Ferros.

<sup>3</sup> Reudismam Rolim de Sousa – Doutor em Computação – UFCG – Campina Grande.

<sup>4</sup> Samara Martins Nascimento Gonçalves – Doutora em Ciência da Computação – UFC – Fortaleza.

**Keywords:** software development; software architecture; loans.

**Disponível em:** <https://periodicos.ufpb.br/index.php/tematica/article/view/74481>

## 1 INTRODUÇÃO

Com a alta demanda na atualidade por recursos tecnológicos, tem-se cada vez mais a necessidade de estratégias para facilitar o gerenciamento de problemas na sociedade. Em especial, na Universidade Federal Rural do Semi-Árido (UFERSA), Campus Pau dos Ferros existe uma demanda de alunos, professores e organizadores de eventos por recursos, como caixas de som, projetores, ferramentas, etc. tornando o acesso a esses recursos fundamental para as atividades acadêmicas e culturais. Dessa forma, é essencial que se implemente estratégias eficientes de gerenciamento destes, uma vez que as ferramentas tradicionais, como planilhas de texto, não são suficientes para ações, como a visualização rápida dos dados, emissão de relatórios, dentre outras funcionalidades, tornando o empréstimo de materiais sujeito a erros e demorado.

Para lidar com essa problemática, foi empregado o método pesquisa-ação, um processo que utiliza abordagens consagradas da literatura para solucionar um processo prático (Tripp, 2005), que neste trabalho refere-se ao empréstimo de materiais. Dentre os tipos de pesquisa-ação, este trabalho se encaixa como uma pesquisa-ação prática, em que os pesquisadores dialogam com a equipe envolvida no processo a ser melhorado para propor uma nova solução (Tripp, 2005).

A problemática desta pesquisa-ação é como reduzir erros e tempo dos envolvidos no empréstimo de materiais. A proposta para melhoria da prática é um software, o Empréstimo Fácil UFERSA (EFU). O desenvolvimento de um software envolve várias etapas (Sommerville, 2011). A proposta se concentra especificamente na construção da Arquitetura do Ambiente, que funciona como um guia para as demais etapas do ciclo de desenvolvimento do software e permite à equipe entender a viabilidade de construção do software (Bass, Clements e Kazman, 2012).

O objetivo é criar uma arquitetura para simplificar e otimizar o processo de empréstimo de equipamentos na universidade, garantindo acesso fácil e eficiente para alunos, professores e colaboradores. A proposta busca assegurar que as transações sejam realizadas de forma organizada e rastreável, melhorando a experiência do usuário e a gestão interna dos materiais.

O sistema conta com formulários para identificação das pessoas e materiais para empréstimos; ao preenchê-los, os dados são armazenados em um banco de dados, garantindo a integridade e segurança das informações. O sistema fornece um mecanismo de comunicação e interface para os usuários que desejam se conectar ao serviço. A intenção é criar uma solução abrangente que atenda às necessidades da comunidade acadêmica, promovendo um ambiente mais colaborativo e acessível.

O sistema também alerta, por e-mail, os usuários de situações diversas, tais como a passagem do prazo estipulado de empréstimo. Por meio de um painel de controle, os administradores podem controlar o uso do sistema, gerenciar e rastrear os materiais. Outras funcionalidades importantes serão o agendamento, geração de relatórios sobre o uso dos equipamentos, podendo criar análises sobre os materiais emprestados.

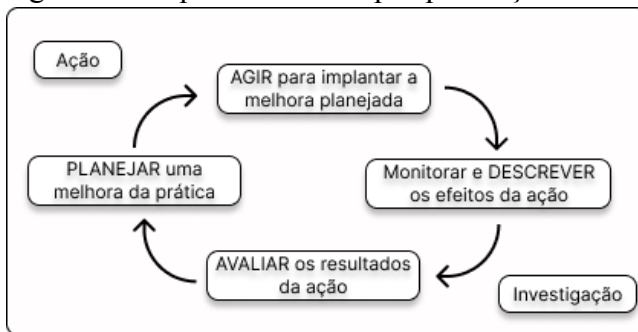
## 2 METODOLOGIA

A metodologia seguiu as etapas do método pesquisa-ação. Ele pode ser realizado de diferentes formas, aplicadas isoladamente ou combinadas para o desenvolvimento de uma

solução. Este trabalho se caracteriza como uma abordagem de pesquisa-ação prática. As abordagens consagradas utilizadas para o desenvolvimento da proposta são aquelas elencadas na Engenharia de Software, em específico, as empregadas para desenvolver Arquitetura de Software. Neste trabalho foi escolhida a abordagem por visões arquiteturais apresentada por Bass, Clements e Kazman (2012), que vem sendo aplicadas em outros trabalhos da literatura (Morais; Melo; Sousa, 2024, Fernandes; Sousa; Silva, 2024, Magalhães; Sousa, 2021, Andrade; Sousa, 2020).

A pesquisa-ação é composta por quatro diferentes etapas (Figura 1). A primeira etapa consiste em entender o contexto e definir um plano para desenvolver uma proposta de intervenção para a problemática apresentada, sendo denominada de planejamento. Por sua vez, na segunda etapa, os pesquisadores se debruçam sobre o problema para propor uma solução, seguindo o plano da etapa anterior, sendo denominada de implementação. Em seguida, vem a etapa de monitoramento da solução para compreender os efeitos da melhoria prática. Por fim, a última etapa objetiva avaliar os impactos da solução desenvolvida (Tripp, 2005). Essas quatro etapas costumam ser resumidas em três fases: planejamento, implementação e avaliação (Tripp, 2005)

Figura 1 - Etapas do método pesquisa-ação



Fonte: Tripp (2005).

Neste trabalho, na etapa de planejamento, foram realizadas ações para levantamento dos requisitos do ambiente, buscando propor intervenções para atender as necessidades dos usuários e levantar elementos a serem incluídos no ambiente proposto. Para levantar esses dados foram realizadas reuniões com os envolvidos e levantamento de dados com a comunidade acadêmica, potenciais usuários da proposta, além do levantamento de artefatos utilizados pela unidade para mapear o empréstimo de materiais. Por sua vez, a etapa de implementação consistiu no desenvolvimento da arquitetura. Na etapa de avaliação, foi realizada uma análise arquitetural por meio de cenários de uso, que exercitou os elementos das visões arquiteturais e permitiu entender a viabilidade do software proposto. Uma sumarização dessas atividades pode ser vista no Quadro 1.

Quadro 1 - representação do ciclo de pesquisa

| Etapa         | Prática                                    | Investigação                                                                          |
|---------------|--------------------------------------------|---------------------------------------------------------------------------------------|
| Planejamento  | Levantamentos dos requisitos               | (a) das necessidades dos usuários<br>(b) dos elementos a serem incluídos no ambiente. |
| Implementação | Desenvolvimento da arquitetura             | do software desenvolvido                                                              |
| Avaliação     | Apresentação do ambiente para os usuários. | (a) da mudança prática<br>(b) do novo processo de empréstimos de materiais            |

Fonte: Autoria própria (2024).

### 3 TRABALHOS RELACIONADOS

Os trabalhos relacionados a este são aqueles que utilizam o método pesquisa-ação e propõem arquiteturas. No tocante ao método pesquisa-ação, Santos e Travassos (2008) investigaram as oportunidades de usos da pesquisa-ação em Engenharia de Software, pela colaboração entre academia e indústria. Para os autores, da perspectiva da academia, há interesse em estudos experimentais, mas da perspectiva da indústria, há o interesse em tecnologias que promovam benefícios reais e de custo conhecido. Portanto, os autores investigaram as nuances entre esses dois diferentes contextos.

Por sua vez, Barbosa, Marinho e Moura (2021) realizaram uma pesquisa-ação para propor um modelo para quantificação de incertezas epistemológicas em projetos de software com metodologias ágeis, focando em estratégias para quantificação em cenários de incertezas epistemológicas.

Já Barbosa, Fonseca e Santos (2019) utilizam o método pesquisa-ação para priorização de requisitos em startups de software. Eles apontam que as startups possuem alto risco de falha, sendo essencial o desenvolvimento de estratégias nesse contexto.

No tocante ao desenvolvimento de arquitetura de software, Moraes, Melo e Sousa (2024) propõem uma arquitetura para o controle de artefatos gerados ao longo do ciclo de vida de projetos de software. Para o desenvolvimento da arquitetura, os autores desenvolvem artefatos para representar as visões arquiteturais.

Por outro lado, Fernandes, Sousa e Silva (2024) apresentam uma arquitetura de software de um jogo sério para apoiar o ensino na Educação básica. As visões arquiteturais são utilizadas para modelar um quiz educação, que pode ser utilizado por professores e estudantes, para fomentar o ensino no processo de ensino-aprendizagem.

#### 3.1 Resultados

Nesta seção são apresentados os resultados para as etapas do método pesquisa-ação.

#### 3.2 Planejamento

Nesta seção é apresentado o planejamento do método pesquisa-ação. Os resultados foram alcançados a partir de reuniões com responsáveis pelo empréstimo de materiais na unidade, que descreveram as etapas para o gerenciamento de empréstimos. Também foram coletados registros dos materiais de controle dos empréstimos.

### 4 REQUISITOS FUNCIONAIS

Para uma melhor compreensão da arquitetura, foram elaborados os requisitos funcionais (Quadro 2). Eles seguiram o padrão de nomenclatura: “[<RF><ID>] Nome do Requisito Funcional”; o RF refere-se a Requisito Funcional e o ID indica o número em sequência; o nome após o colchete simboliza o nome do requisito. Os requisitos foram definidos por meio de um questionário aplicado por Melo et al. (2024) para elencar as principais necessidades dos usuários no controle e rastreamento dos requisitos.

Quadro 2 - Lista de Requisitos funcionais. P.: Prioridade, E: Essencial, I: Importante e D: Desejável

| Requisito                       | Descrição                                                                                                                                                                            | P. |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| [RF001] Cadastrar-se            | O sistema deve notificar os usuários em relação aos status do empréstimo/agendamento.                                                                                                | E  |
| [RF002] Entrar                  | Permitir acessar a plataforma utilizando o login e senha.                                                                                                                            | E  |
| [RF003] Listar empréstimos      | Permitir visualizar os empréstimos realizados no momento.                                                                                                                            | E  |
| [RF004] Solicitar empréstimo    | O usuário pode solicitar o empréstimo do material, preenchendo um formulário. Este formulário conterá CPF ou matrícula, data inicial, data final e selecionar o material disponível. | E  |
| [RF005] Solicitar devolução     | O usuário pode solicitar antecipação para devolução do material, que após a devolução é necessária confirmação do administrador.                                                     | E  |
| [RF006] Listar de empréstimos   | O usuário administrador pode ver todos os empréstimos (em andamento, devolvidos, concluídos e solicitados).                                                                          | E  |
| [RF007] Listar materiais        | O usuário administrador pode ver todos os materiais no sistema (disponível e não disponível).                                                                                        | E  |
| [RF008] Confirmar empréstimo    | O usuário administrador pode confirmar o empréstimo do material solicitado pelo usuário.                                                                                             | E  |
| [RF009] Negar empréstimo        | O usuário administrador pode negar o empréstimo do material solicitado pelo usuário.                                                                                                 | E  |
| [RF010] Confirmar devolução     | O usuário administrador pode confirmar as devoluções do empréstimo solicitado pelo usuário.                                                                                          | E  |
| [RF011] Remover material        | O usuário administrador pode retirar o material disponível.                                                                                                                          | E  |
| [RF012] Editar material         | O usuário pode editar as informações do material disponível.                                                                                                                         | E  |
| [RF013] Cadastrar novo material | O usuário administrador pode cadastrar novos materiais, preenchendo formulário com informações como: nome, descrição e quantidade.                                                   | E  |

Fonte: Autoria própria (2024).

## 5 REQUISITOS NÃO FUNCIONAIS

Nesta seção, serão apresentados os requisitos não funcionais do sistema (Quadro 3), que seguiram o padrão de nomenclatura: “[<RNF><ID>] Nome do Requisito Não Funcional”; o RNF refere-se a Requisito Não Funcional e o ID indica o número em sequência; o nome após o colchete simboliza o nome do requisito.

Quadro 3 - Lista de requisitos não funcionais. U: Usabilidade, I: Integração, S: Segurança, ME: Manutenibilidade e Escalabilidade e A: Atuação.

| Requisito                            | Descrição                                                                                      | Categoria |
|--------------------------------------|------------------------------------------------------------------------------------------------|-----------|
| [RNF001]<br>Notificações             | O sistema deve notificar os usuários em relação aos status do empréstimo/agendamento.          | U         |
| [RNF002]<br>Segurança                | O sistema deve proteger contra acesso não autorizado.                                          | S         |
| [RNF003] Alta capacidade de usuários | O sistema deve ser capaz de distribuir o acesso dos usuários sem qualquer perda de desempenho. | A         |

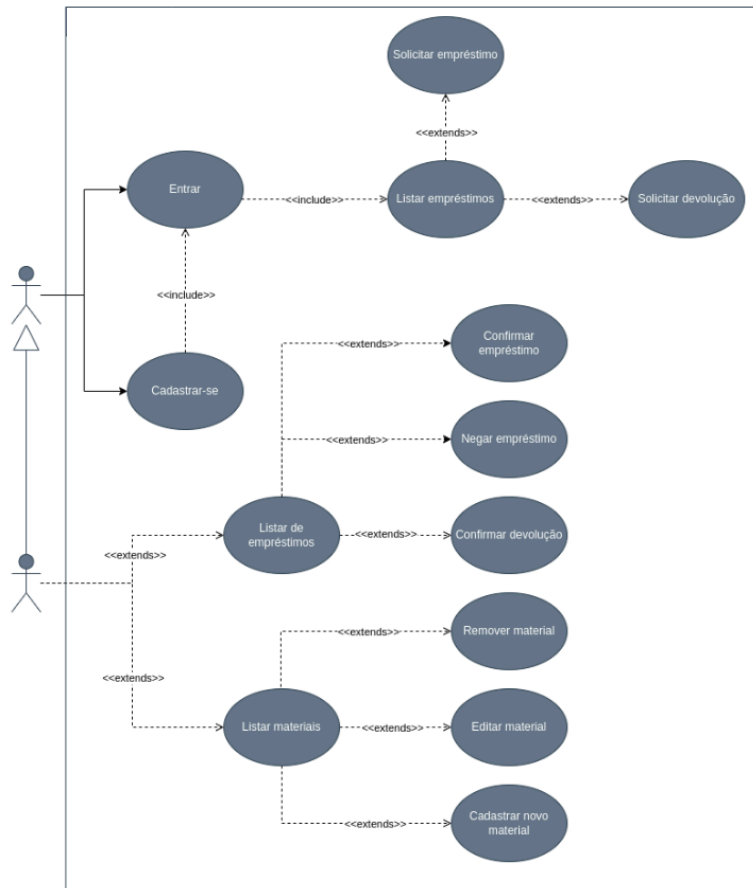
**Fonte:** Autoria própria (2024).

## 6 CASOS DE USO

Os casos de uso são fundamentais para ilustrar as interações entre os atores e o sistema (Guedes, 2018). Eles são compostos por três elementos principais: atores, casos de uso e relacionamentos (Guedes, 2018). Os atores representam os usuários que interagem com a aplicação e são simbolizados por um boneco de palito (Guedes, 2018).

Os casos de uso, que descrevem as funcionalidades do sistema, costumam ser formulados com verbos ou expressões verbais para indicar ações específicas. Eles são representados por um símbolo oval, com a descrição da ação centralizada internamente (Guedes, 2018). Além disso, os relacionamentos estabelecem as conexões entre os atores e os casos de uso, delineando como as interações ocorrem (Guedes, 2018). Na Figura 2, é possível observar o diagrama de casos de uso para o sistema EFU.

Figura 2 - Diagrama de casos de uso



Fonte: Autoria própria (2024).

## 7 IMPLEMENTAÇÃO

Nesta seção, apresenta-se a etapa de implementação do método pesquisa-ação. Essa etapa consistiu na proposta de arquitetura para o EFU, apresentando elementos que compõem a pré-criação do software, tais como documentações, diagramas, figuras e relacionamentos, que representam partes do sistema. Esta arquitetura visa consolidar o conhecimento obtido através de reuniões, para projetar e obter a melhor comunicação entre as partes interessadas, validando e escolhendo as melhores opções abordadas.

### 7.1 Visões arquiteturais

As visões arquiteturais do software são representações que mostram o ambiente de diferentes aspectos, sua compreensão e análise de suas complexidades (Bass, Clements e Kazman, 2012). Para este projeto, foram elaboradas as três principais visões: Visão de Módulo, Visão Componente & Conector e Visão de Alocação.

### 7.2 Visão de módulo

A Visão de Módulo é uma parte fundamental de um projeto de software, que busca elencar como os elementos internos do software estão estruturados (Bass, Clements e Kazman, 2012). Para representar essa visão foi utilizado um diagrama de classes da UML

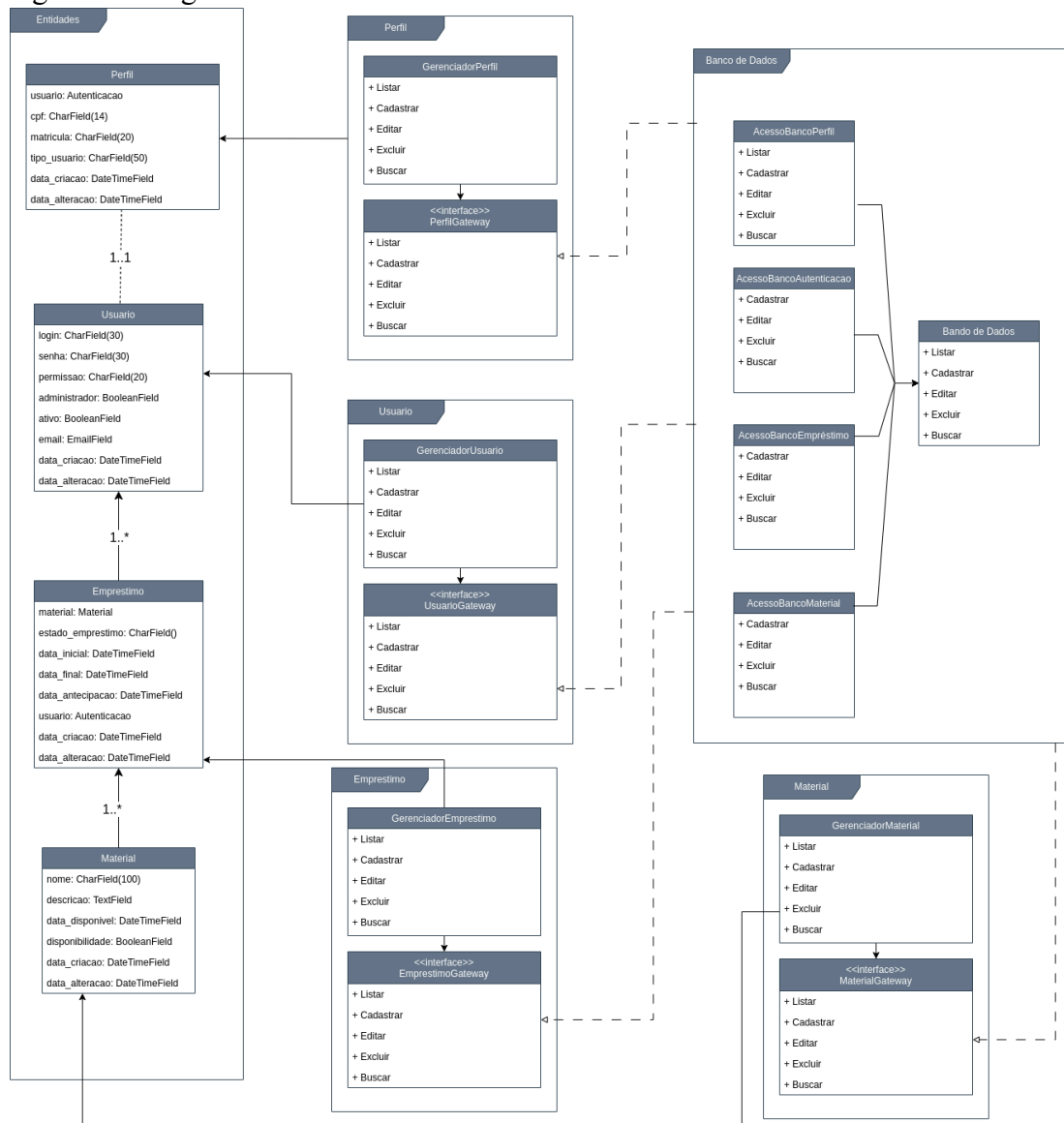
(Figura 3). Esse tipo de diagrama oferece uma visão clara e concisa das classes que compõem o sistema, bem como seus atributos, métodos e os relacionamentos entre eles (Guedes, 2018). Desta forma, proporciona uma compreensão da estrutura do sistema, facilitando a identificação de componentes essenciais e suas interações.

O diagrama é composto pelas classes material, empréstimo, usuário e perfil; cada classe representa um conjunto de informações, com seus atributos. Começando pela classe perfil, que é utilizada para guardar os dados extras do usuário e possui os seguintes atributos e relacionamentos: usuário, que funciona como identificador do relacionamento de um para um para a outra classe de usuário; CPF, que representa o número do CPF para identificar o solicitante do empréstimo; matrícula, se for vinculado a universidade; tipo de usuário, representando o nível de permissões da conta; dados de informações como data de criação e data de alteração.

Já a classe usuário é utilizada como acesso ao sistema; possui os seguintes atributos e relacionamentos: login e senha, são os campos utilizados para entrar no sistema; permissão, define qual nível de permissão o usuário possui para acessar certos recursos; administrador, indica se o usuário deve ter permissão de tudo; ativo, revela se está conta ainda pode ser utilizada; email, guarda o e-mail para comunicação com o usuário; datas, dados de informações como data de criação e data de alteração; há também um relacionamento com o perfil, que pode conter informações deste acesso de login.



Figura 3 - Diagrama de classes



**Fonte:** Autoria própria (2024).

Na classe Material, tem-se o que será emprestado, com os seguintes atributos e relacionamentos: nome e descrição, são referências para identificarmos o material; data disponível, indica quando o material ficará à disposição de novos empréstimos; disponibilidade, que indica se no exato momento em que este material é consultado, se encontra disponível; dados de informações como data de criação e data de alteração; o relacionamento de material pode estar associado a múltiplos registros com empréstimo, visto que o material será devolvido e posto a ser emprestado novamente.

E, por fim, a classe de empréstimo conterá as informações da gerência do empréstimo, com os seguintes atributos e relacionamentos: material, referência ao material emprestado; estado empréstimo, estado atual do empréstimo (ex: ativo, finalizado); data inicial, data de início do empréstimo; data final, data de término do empréstimo; data antecipada, data de antecipação do empréstimo; usuário, referência ao usuário que fez o empréstimo; dados de informações como data de criação e data de alteração

O diagrama de classes representa um sistema, que gerencia perfis, usuários, empréstimos de materiais e os próprios materiais. As relações entre as entidades garantem que

cada usuário tenha um perfil e um registro, além de permitir que eles realizem múltiplos empréstimos de diferentes materiais, desde que o material se encontre em estado de disponibilidade no momento da criação do empréstimo.

### **7.3 Visão componente & conector**

A Visão Componente & Conector é crucial para entender como os elementos se comportam em tempo de execução (Bass, Clements e Kazman, 2012) e pode ser representada por um diagrama de componentes, que ilustra os diferentes componentes e suas interações (Guedes, 2018). Neste diagrama, pode-se notar os componentes que representam as entidades, gerenciar alunos, gerenciar empréstimos e gerenciar relatórios. O diagrama correspondente ao sistema proposto está apresentado na Figura 4.

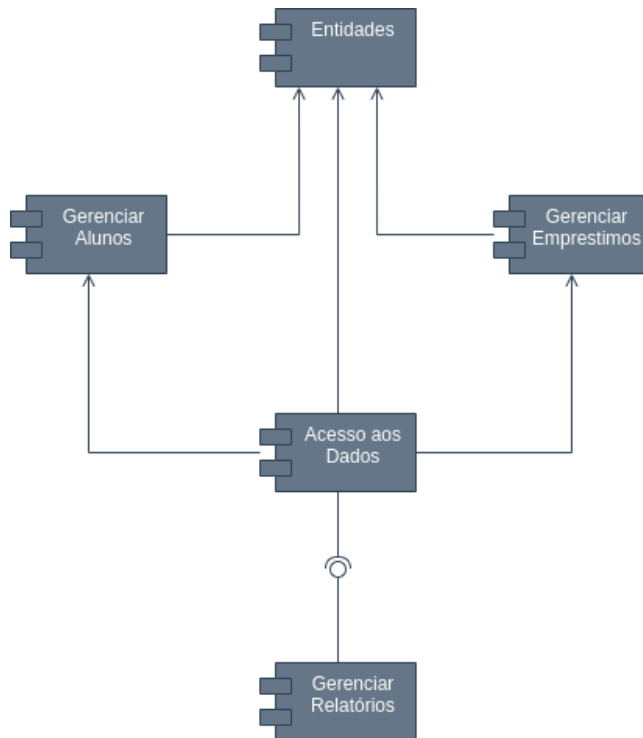
O componente Gerenciar Alunos é acessível aos usuários, permitindo que se realizem ações como criar, editar e excluir dados do usuário. O componente Gerenciar Empréstimos permite a criação, edição e exclusão dos empréstimos. Além disso, o componente Gerenciar Relatórios oferece a funcionalidade de gerar documentos, que compilam dados dos projetos e podem ser acessados pelos usuários envolvidos.

### **7.4 Visão de alocação**

A Visão de Alocação é uma representação arquitetural dos recursos alocados para o sistema, incluindo as comunicações, servidores e dispositivos (Bass, Clements e Kazman, 2012). Essa interação ocorre entre os elementos que compõem todo o projeto com o ambiente externo do software (Guedes, 2018).

Antes de se aprofundar nas ligações, é importante entender como um projeto é estruturado, especialmente, em relação à divisão entre back-end e front-end. Para atender às expectativas de um bom sistema, ele foi pensado em uma divisão de desenvolvimento entre o front-end e back-end; o front-end é a parte que os usuários interagem diretamente, geralmente através de um navegador web. A interação começa quando o usuário acessa uma página na internet. Para isso, o navegador faz uma solicitação ao servidor por um arquivo específico. O back-end, por outro lado, opera nos bastidores. É responsável pela lógica e manipulação de dados.

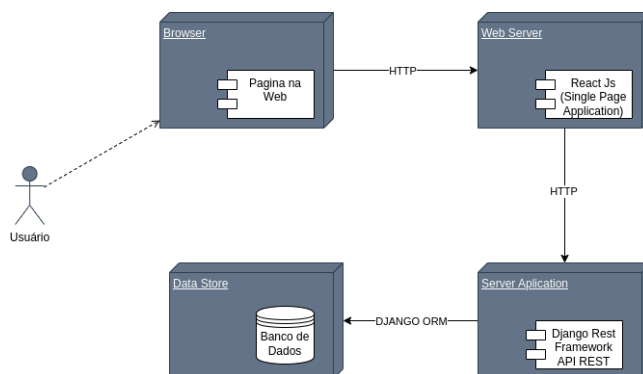
Figura 4 - Diagrama de componentes



**Fonte:** Autoria própria (2024).

A Visão de Alocação pode ser vista no diagrama da Figura 5. Neste diagrama, tem-se a infraestrutura que o usuário interage com o navegador da web, através de uma página na internet. Essa página para ser acessada, precisa solicitar ao servidor um arquivo. Esse servidor contém todos os códigos do sistema front-end (Web Server), para a obtenção de informações essa página solicita ao back-end (Server Application). Essa comunicação é feita através de requisições HTTP (HyperText Transfer Protocol), que é um protocolo de transferência de dados através de páginas Web. Quando a solicitação chega até os servidores em que o back-end foi estruturado, ele utiliza-se de consultas ao banco de dados para capturar informações e enviá-las no caminho reverso até o usuário final.

Figura 5 - Diagrama de implantação do ambiente



**Fonte:** Autoria própria (2024).

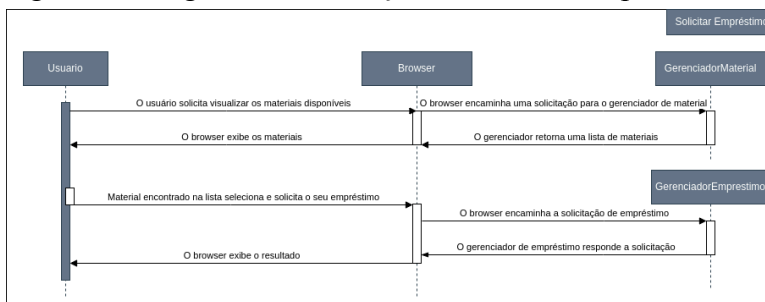
## 8 AVALIAÇÃO

A avaliação busca verificar se o que está sendo projetado atende aos objetivos e necessidades dos usuários (Bass, Clements e Kazman, 2012). Para isso, tem-se o Método de Análise Arquitetural baseado em Tradeoffs (Architecture Tradeoff Analysis Method) - ATAM (Bass, Clements e Kazman, 2012). Na avaliação foi feito uso de um de seus elementos, que se trata do uso de cenários. Através dos cenários, é possível entender se a estrutura do software atende a proposta do sistema e quais as limitações dos usuários com funcionalidades do sistema.

### 8.1 Cenários

O primeiro cenário foi baseado na interação entre o usuário, o browser e os gerenciadores de material e empréstimo. Ao requisitar um empréstimo, o usuário necessita da lista de materiais disponíveis. Ao escolher o material dessa lista, o usuário preenche um formulário, com as informações do empréstimo. O usuário submete o formulário e é criada uma solicitação de empréstimo (Figura 6).

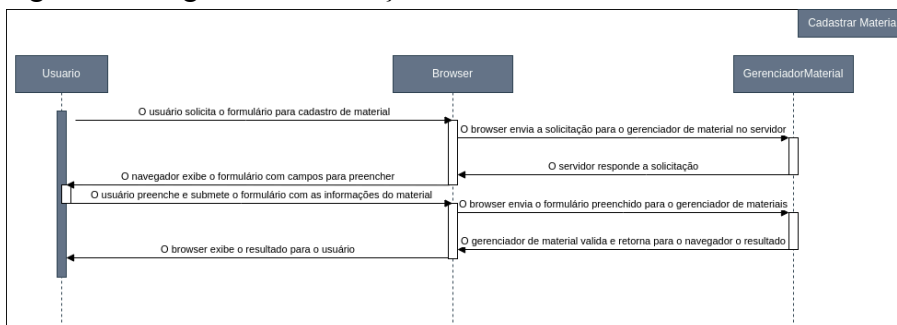
Figura 6 - Diagrama de interações - Solicitar empréstimo



Fonte: Autoria própria (2024).

No segundo cenário, tem-se o cadastro de material, que será realizado inicialmente solicitando um formulário ao gerenciador de material. Ao receber o formulário, o usuário preenche as informações necessárias do novo material e submete. O servidor avalia o formulário e retorna uma mensagem de sucesso ou de erro ao cadastrar o material a depender das informações preenchidas (Figura 7).

Figura 7 - Diagrama de interações - Cadastrar material

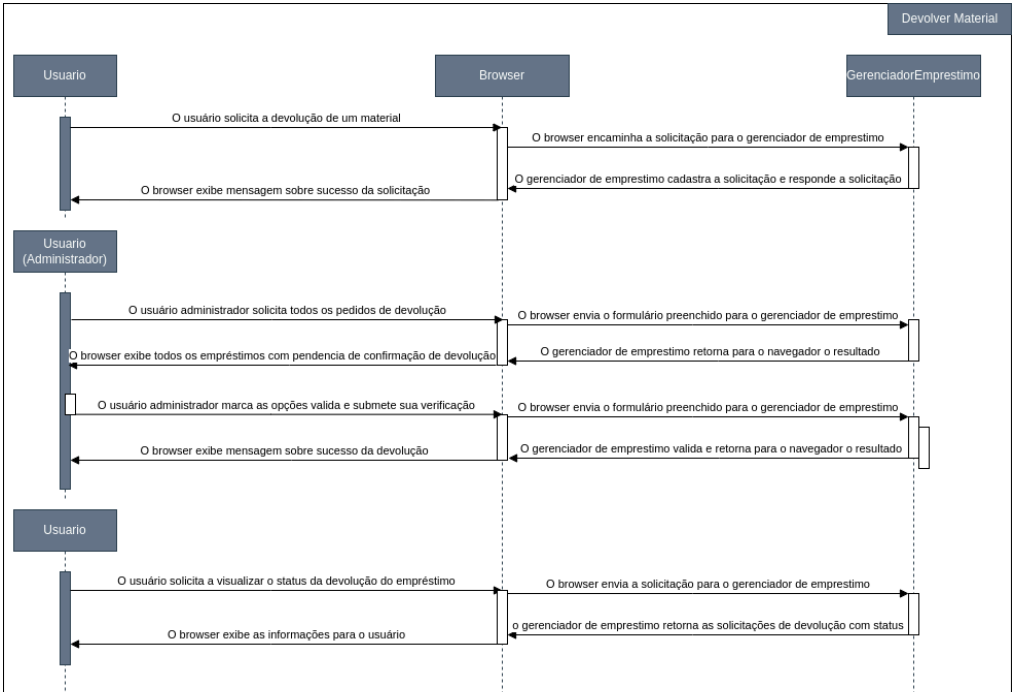


Fonte: Autoria própria (2024).

Na Figura 8, é ilustrado o processo de devolução de material, em que o usuário e administrador realizam comunicações através do browser e o gerenciador de empréstimos. O

usuário lista os materiais emprestados e seleciona a opção de devolução, deixando uma solicitação no sistema. Em seguida, o usuário administrador, ao entrar no sistema, consegue visualizar as solicitações pendentes para ser avaliada, com isso, através do browser faz um pedido ao gerenciador de empréstimo com todos os pedidos de devolução. O usuário administrador avalia o pedido confirmando ou negando a solicitação de devolução. Por fim, o usuário, após o aceite ou negação do administrador, consegue visualizar o status da sua solicitação.

Figura 8 - Diagrama de interações - Devolver material



Fonte: Autoria própria (2024).

## 9 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentada uma arquitetura para um ambiente de gerenciamento de empréstimo de materiais denominado "Empréstimo Fácil UFERSA (EFU)". O ambiente objetiva gerenciar e simplificar o processo de gestão e de solicitação de materiais. Para o desenvolvimento do ambiente, foi utilizada a metodologia pesquisa-ação, que busca melhorar um processo prático. Utilizando as etapas propostas nesse método, foi possível planejar o desenvolvimento do ambiente, implementar a arquitetura de uma solução por meio de visões arquiteturais proposta em Bass, Clements e Kazman (2012), que modela o ambiente por meio de três visões arquiteturais: visão de módulo, visão componente & conector e visão de alocação. A arquitetura foi avaliada por meio de abordagem baseada em cenários. O ambiente proposto, uma vez implementado, possui contribuições significativas para a organização geral de instituições de ensino, como a UFERSA. Como trabalhos futuros, pretende realizar a implementação do back-end e front-end do ambiente e avaliar o ambiente com usuários reais da comunidade acadêmica.

## REFERÊNCIAS

- ANDRADE, E.; SOUSA, R. R. Uma arquitetura de um sistema para interação entre alunos e professores na aprendizagem de programação. **Brazilian Journal of Development**, v. 6, n. 6, p. 40868-40890, 2020.
- ATLASSIAN. Atlassian. **Software development and collaboration tools**. Disponível em: <<https://www.atlassian.com/>>. Acesso em: 25 de fevereiro de 2024.
- BARBOSA, J. F.; MARINHO, M. L. M.; MOURA, H. P. Em direção a um modelo para quantificação da incerteza epistêmica em projetos de software: uma Pesquisa-Ação. **Revista Ibérica de Sistemas e Tecnologias de Informação**, v. 1, n. 44, p. 67-83, 2021.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. 3. ed. Boston, MA, USA: Addison-Wesley Professional, 2012.
- COHN, M.; SILVA, A. J. C. C.; PRIKLADNICKI, R. **Desenvolvimento de software com Scrum: Aplicando Métodos Ágeis com Sucesso**. 1. ed. Grand Haven, MI, USA: Bookman, 2011.
- ENGEL, G. I.. Pesquisa-ação. **Educar em Revista**, v. 66, n. 1, p. 181-191. 2020.
- FERNANDES, J. R.; SOUSA, R. R.; SILVA, V. M. L. Um jogo sério para promoção da educação no ensino básico. **Revista Científica Multidisciplinar**, v. 5, n. 10, p. e5105758-e5105758, 2024.
- FIGMA. **Figma: the collaborative interface design tool**. Disponível em: <<https://www.figma.com/>>. Acesso em: 25 fev. 2024.
- GUEDES, G. T. A. **UML 2 - uma abordagem prática**. São Paulo, SP: Novatec, 2018.
- MAGALHÃES, K. D.; SOUSA, R. R. Uma arquitetura de software para um sistema interno para comunicação entre setores. **Brazilian Journal of Development**, v. 7, n. 12, p. 112700-112713, 2021.
- MELO, T. R. S.; MORAIS, E. B. D.; SOUSA, R. R.; GONÇALVES, S. M. N. **Utilizando o método pesquisa-ação para desenvolver um protótipo de um sistema para empréstimos de materiais para a UFERSA**, Campus Pau dos Ferros. **Revista de Engenharia e Tecnologia**, v. 17, n. 1, 2025.
- MORAIS, E. B. D.; MELO, T. R. S.; SOUSA, R. R. Arquitetura de uma ferramenta para controle e rastreamento de artefatos de software. **Peer Review**, v. 6, n. 9, p. 242-256, 2024.
- SANTOS, L.; SOUSA, R. R. DE; SILVA, J. W. C. Arquitetura de um ambiente para a Associação de Apoio aos Portadores de Câncer de Mossoró e Região (AAPCMR). **Conjecturas**, v. 22, n. 15, p. 958–976, 2022.
- SANTOS, P. S. M.; TRAVASSOS, G. H. Colaboração entre academia e indústria: oportunidades para utilização da pesquisa-ação em engenharia de software. In: **Workshop de Engenharia de Software Experimental**. 2008.
- SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Pearson Prentice Hall, 2011.

TRIPP, David. Pesquisa-ação: uma introdução metodológica. **Educação e Pesquisa**, v. 31, p. 443-466, 2005.