

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA
DIRETORIA DE DESENVOLVIMENTO DE ENSINO
COORDENAÇÃO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

VIVALDO VÍTOR MACEDO FRANÇA

RELATÓRIO TÉCNICO

Documentação de Produto Implementado por Software: LimpAeh

GUARABIRA – PB

2025

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA
DIRETORIA DE DESENVOLVIMENTO DE ENSINO
COORDENAÇÃO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

RELATÓRIO TÉCNICO

Documentação de Produto Implementado por Software: LimpAeh

Vivaldo Vítor Macedo França

Orientador (IFPB)

Prof. Dr. Rodrigo Leone Alves

Coorientador (IFPB)

Prof. Dr. Nasson Paulo Sales Neves

Trabalho de Conclusão de Curso apresentado ao
Curso de Tecnologia em Sistemas para Internet, do
Instituto Federal da Paraíba – Campus Guarabira, em
cumprimento às exigências parciais para a obtenção
do título Tecnólogo em Sistemas para Internet.

GUARABIRA - PB

2025

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFPB - GUARABIRA

F815d

França, Vivaldo Vítor Macedo

Documentação de Produto Implementado por Software: LimpAeh / Vivaldo Vítor Macedo França.- Guarabira, 2025.

32 f.: il; color.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Sistemas para Internet). – Instituto Federal da Paraíba, Campus Guarabira, 2025.

"Orientação: Prof. Dr. Rodrigo Leone Alves"

"Co-orientador: Prof. Dr. Nasson Paulo Sales Neves."

Referências.

1. Sistemas para Internet. 2. Gerenciamento de limpeza. 3. Controle de atividade. I. Título.

CDU 004.4(0.067)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA

ATA 11/2025 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO
CCS de Tecnologia em Sistemas para Internet

Aos 13 de março de 2025, às 13:30, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Nasson Paulo Sales Neves (Coorientador), Rhavy Maia Guedes (Examinador Interno), Lucas Vieira de Souza (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *campus* Guarabira, desenvolvido pela aluno(a) **Vivaldo Vitor Macedo França**, matrícula de nº **202013810030**, intitulado "LimpAeh", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 90 (Noventa)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 13 de março de 2025.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves**, PROFESSOR ENS BASICO TECN TECNOLÓGICO, em 20/03/2025 13:30:34.
- **Lucas Vieira de Souza**, COORDENADOR(A) DE CURSO - FUC1 - CCSTSI-GB em 20/03/2025 16:55:39.
- **Rhavy Maia Guedes**, PROFESSOR ENS BASICO TECN TECNOLÓGICO, em 20/03/2025 19:19:38.



Documento assinado digitalmente

NASSON PAULO SALES NEVES
Data: 22/03/2025 16:26:54-0300
Verifique em <https://validar.iti.gov.br>

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 685278
Verificador: 41993803df
Código de Autenticação:



Dedico este trabalho primeiramente ao meu Deus, que nunca me abandonou e sempre me foi fiel, à minha família, à minha mãe, ao meu pai e aos que sempre estiveram e estão comigo nos momentos bons e ruins.

AGRADECIMENTOS

Quero agradecer primeiramente a Deus por este momento tão importante e especial na minha vida. Sem Ele, eu não estaria aqui e nada seria possível.

À minha família, em especial meus pais, Edenize Macedo França e Vivaldo Luis de França, que sempre me apoiaram e me incentivaram a seguir meus sonhos.

A uma pessoa muito especial que entrou na minha vida e mudou meu modo de enxergar as maravilhas deste mundo: minha amada Raquel Lucena.

Aos professores, em especial o professor Me. Rhavy Maia Guedes, que, além de um excelente docente, é também uma belíssima pessoa e me ajudou muito nesse projeto.

Aos amigos e colegas que sempre me ajudaram de alguma forma, seja tirando um tempo para diversão, seja para uma conversa sadia.

“Quem é sábio procura aprender, mas os
tolos estão satisfeitos com a sua própria
ignorância ” Provérbios 15:14

RESUMO

O objetivo deste projeto é aprimorar e otimizar o gerenciamento de limpeza das instituições públicas, denominado LimpAeh, facilitando as comunicações entre administradores, supervisores e funcionários de limpeza. O sistema permite o registro, a designação e o monitoramento das atividades, promovendo eficiência operacional e transparência no controle das tarefas realizadas. As tecnologias utilizadas no sistema incluem o desenvolvimento web com React para o front-end e Flask para o back-end, proporcionando uma solução web escalável e de fácil utilização. O sistema contribui para a centralização das informações, a redução do tempo de execução de tarefas administrativas e a geração de relatórios de desempenho. Logo, conclui-se que o LimpAeh oferece uma ferramenta eficiente no que diz respeito ao aprimoramento da gestão de serviços higiênicos, com potencial de ser adaptada às necessidades de outros contextos institucionais.

Palavras-chave: gerenciamento de limpeza, software, controle de atividades

ABSTRACT

The objective of this project is to enhance and optimize the cleaning management of public institutions, named LimpAeh, by facilitating communication between administrators, supervisors, and cleaning staff. The system enables the registration, assignment, and monitoring of activities, promoting operational efficiency and transparency in task management. The technologies used include web development with React for the front-end and Flask for the back-end, providing a scalable and user-friendly web solution. The system contributes to the centralization of information, reduction of administrative task execution time, and generation of performance reports. Therefore, it is concluded that LimpAeh offers an efficient tool for improving the management of cleaning services, with the potential to be adapted to the needs of other institutional contexts.

Keywords: cleaning management, software, task control

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Dados Gerais	9
1.2 Tema	9
1.3 Objetivo do Projeto	9
1.4 Delimitação do Problema	10
1.5 Justificativa	10
1.6 Método de Trabalho	10
1.7 Organização do Trabalho	11
1.8 Sobre os Objetivos de Desenvolvimento Sustentável	12
1.9 Sobre a Lei Geral de Proteção de Dados	12
 2 DESCRIÇÃO GERAL DO PROBLEMA	 13
2.1 Descrição do Problema	13
2.2 Perfis de Usuários	13
2.3 Requisitos	14
2.3.1 Requisitos Funcionais.....	14
2.3.2 Requisitos Não Funcionais	16
 3 ANÁLISE E DESIGN	 18
3.1 Diagrama de Classes	18
3.2 Diagrama de Atividades	19
3.3 Modelo de Dados	20
3.3.1 Modelo Lógico da Base de Dados	20
3.3.2 Criação Física do Modelo de Dados	22
3.3.3 Dicionário de Dados	23
3.4 Ambiente de Desenvolvimento	26
 4 IMPLEMENTAÇÃO	 28
 5 CONCLUSÃO	 31
 REFERÊNCIAS	 32

1 INTRODUÇÃO

A gestão eficiente de atividades de limpeza em instituições públicas é essencial para garantir ambientes saudáveis e organizados. No entanto, o gerenciamento manual dessas tarefas muitas vezes resulta em falta de controle, comunicação ineficaz e baixa produtividade. Para superar esses desafios, o projeto LimpAeh visa solucionar tecnologicamente esses problemas, organizando atividades de limpeza por meio de um sistema de registro e monitoramento eficiente.

1.1 Dados Gerais

Atualmente, o gerenciamento de atividades de limpeza em muitas instituições públicas é feito de forma manual, o que dificulta o controle das tarefas e a comunicação eficiente entre os supervisores e funcionários. Com isso, o software LimpAeh surge como uma alternativa eficaz para solucionar essa problemática rotineira nos institutos.

1.2 Tema

O LimpAeh é uma ferramenta desenvolvida para otimizar a comunicação automatizada entre direção administrativa, supervisores e funcionários. Essa solução é fundamental no que diz respeito ao monitoramento das atividades, uma vez que o registro de designações e atividades estarão armazenados no banco de dados da aplicação. Além de garantir um controle eficaz sobre as operações, o software também assegura a padronização e a qualidade das informações geradas.

1.3 Objetivo do Projeto

1.3.1 Objetivo Geral:

O objetivo geral do projeto é desenvolver uma plataforma que permita a comunicação automatizada entre direção administrativa, supervisores e funcionários para o gerenciamento das atividades de limpeza nas instituições públicas de ensino. Além disso, a plataforma oferecerá registro e monitoramento dessas atividades.

1.3.2 Objetivo Específicos:

- Implementar funcionalidades de registro e designação de atividades de limpeza;
- Desenvolver mecanismos de comunicação para automatizar o fluxo de informações entre administradores, supervisores e funcionários;
- Integrar relatórios sobre as atividades feitas;
- Assegurar um processo de login seguro, preservando a privacidade das informações;

1.4 Delimitação do Problema

A delimitação deste projeto concentra-se na solução do problema enfrentado pelos supervisores e funcionários acerca da comunicação informal e ineficiente entre eles e a administração. O software proposto visa proporcionar uma comunicação fluida e eficaz entre os funcionários do campus. Porém, o sistema não incluirá funcionalidades como gestão de inventário de produtos de limpeza ou automação de processos físicos. O foco está na otimização da comunicação entre as partes envolvidas, bem como no registro e monitoramento de designações e atividades.

1.5 Justificativa

Este software foi inspirado na sugestão do professor de Programação Web II, Me. Rhavy Maia Guedes, que propôs o desenvolvimento de um sistema que fosse capaz de gerenciar as atividades de limpeza do campus. A proposta é atender à necessidade de aprimorar a comunicação e o controle dessas atividades, gerando maior eficiência no controle dessas tarefas realizadas.

1.6 Método de Trabalho

A metodologia utilizada para desenvolver o projeto LimpAeh foi o Scrum, um método ágil que facilita a compreensão e o desenvolvimento de um trabalho. Essa metodologia é uma estrutura de gestão de projetos que auxilia as equipes a se organizarem e colaborarem para

alcançar um objetivo em comum. Ela é baseada em princípios ágeis e é uma das formas mais populares de gerenciamento de projetos a curto prazo.

O Scrum trabalha em sprints, com duração padrão de 15 (quinze) dias, e durante esse período, as tarefas para implementação do software eram atribuídas. Além de acelerar a entrega, essa abordagem incremental e iterativa também garante flexibilização e adaptação no que diz respeito às mudanças ou necessidades do sistema.

Para a modelagem do banco de dados, foi adotada a Programação Orientada a Objetos (POO), que permite a construção de sistemas mais organizados e robustos. A POO é um paradigma de programação que organiza o código em torno de objetos — instâncias de classes — facilitando a representação de elementos do mundo real, com objetos que possuem atributos e métodos. Esse paradigma promove modularidade, reuso de código e melhor manutenção do sistema, ao permitir a criação de objetos com atributos e métodos específicos.

1.7 Sobre os Objetivos de Desenvolvimento Sustentável

É importante destacar a contribuição do software para a sustentabilidade e a promoção da qualidade de vida nas instituições de ensino, em alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS) definidos pela Organização das Nações Unidas (ONU). Para isso, destacam-se nesse projeto o ODS 3 (Saúde e Bem-Estar), ao garantir ambientes mais limpos e saudáveis por meio de uma gestão eficiente de limpeza; o ODS 8 (Trabalho Decente e Crescimento Econômico), por melhorar a comunicação e a produtividade dos funcionários; o ODS 9 (Indústria, Inovação e Infraestrutura), ao implementar uma solução tecnológica moderna; e o ODS 12 (Consumo e Produção Responsáveis), por contribuir para a redução do uso de papéis e outros recursos.

Ao adotar esse conjunto de normas, o projeto reforça seu compromisso com a qualidade de vida sustentável, promovendo maior eficiência operacional no dia a dia dos funcionários e gestores do campus.

1.8 Sobre a Lei Geral de Proteção de Dados

A Lei Geral de Proteção de Dados (LGPD) foi estabelecida para proteger as informações pessoais e garantir maior segurança no tratamento de dados no Brasil. Este projeto adota medidas para assegurar a conformidade com essa lei, implementando autenticação segura para proteger dados e credenciais contra acessos não autorizados (BRASIL, 2018). Os principais pontos de adequação do sistema LimpAeh à LGPD incluem:

- **Consentimento:** O acesso ao sistema requer que os usuários forneçam consentimento explícito para o armazenamento de suas credenciais no banco de dados.
- **Finalidade e Necessidade:** Os dados coletados são usados exclusivamente para as funcionalidades de gerenciamento e comunicação do sistema, sem qualquer coleta ou compartilhamento para ações externas.
- **Segurança:** O sistema oferece segurança utilizando tecnologia de autenticação JWT (Json Web Token) com senha criptografada e conexões seguras.
- **Direito de Acesso e Exclusão:** Somente administradores têm permissão para gerenciar os dados dos funcionários, podendo cadastrar, editar ou excluir informações desatualizadas ou desnecessárias conforme apropriado.

A inclusão da LGPD no sistema não apenas reforça a segurança dos dados, mas também evidencia um compromisso ético e moral com a proteção da privacidade dos usuários.

1.9 Organização do Trabalho

O documento foi estruturado para detalhar os principais aspectos do projeto, incluindo a problemática que o software busca resolver, os requisitos funcionais e não funcionais, e os diferentes perfis de usuários. Também são apresentados o design do sistema, abrangendo diagramas de classes e atividades, a modelagem do banco de dados, além da organização e configuração do ambiente de desenvolvimento.

2 DESCRIÇÃO GERAL DO SISTEMA

O sistema “LimpAeh” foi projetado para automatizar os meios de comunicação entre direção administrativa, supervisores e funcionários, proporcionando um fluxo de informações mais sistemático, rápido e organizado. A solução aborda a problemática da comunicação informal e ineficiente, que impacta negativamente a supervisão e a gestão das atividades de limpeza em ambientes institucionais.

Além disso, a organização eficiente das atividades de limpeza requer práticas sistematizadas e uso de técnicas específicas para a uma execução limpa e eficaz das tarefas. Segundo Hygibras (2024), a limpeza de alta performance atrelada a criação de um roteiro de limpeza é essencial para manter a saúde e a segurança dos alunos e funcionários de uma escola. A automatização do fluxo de comunicação do LimpAeh vai integrar essa roteirização, garantindo uma melhor qualidade de vida para os gestores, funcionários e alunos da instituição.

2.1 Descrição do Problema

O fluxo de comunicação informal entre direção administrativa e funcionários pode ocasionar falhas de comunicação e informações imprecisas. Essa ineficiência pode levar à execução de atividades incorretas, tendo como consequência a falta de higienização em alguns ambientes, comprometendo a qualidade e a segurança local da instituição. Em virtude disso, fica evidente que uma plataforma como o LimpAeh facilitaria a gestão dos administradores e funcionários de limpeza.

2.2 Perfis de Usuário

O sistema *LimpAeh* define três perfis de usuário: administrador, supervisor e funcionário. Cada perfil desempenha um papel específico e essencial para a gestão eficiente das atividades de limpeza.

Perfil 01 - Administrador - O administrador tem a função de gerenciar o sistema em sua totalidade. Ele poderá cadastrar, editar e excluir dados ou registros. Além disso, o

administrador é encarregado de notificar o supervisor sobre a necessidade de uma nova atividade de limpeza.

Perfil 02 - Supervisor - O supervisor gerencia diretamente as atividades de limpeza. Ele recebe as solicitações enviadas pelo administrador, cria as atividades necessárias e designa cada uma delas a um funcionário para execução.

Perfil 03 - Funcionário - O funcionário executa as atividades de limpeza designadas pelo supervisor. Sua principal função no sistema é registrar os relatórios detalhados das atividades concluídas, garantindo um acompanhamento preciso das tarefas realizadas.

2.3 Requisitos

2.3.1 Requisitos Funcionais

RF_01 - Cadastro de usuário - O sistema deve permitir que o administrador cadastre novos usuários. O cadastro pode incluir diferentes perfis: administrador, supervisor ou funcionário. Durante o processo, o administrador deve preencher os seguintes campos obrigatórios: nome completo, e-mail, senha, tipo de usuário e nome da empresa (para o perfil de funcionário).

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_02 - Cadastro de ambiente - O sistema deve permitir que o administrador cadastre novos ambientes para o gerenciamento de atividades de limpeza.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_03 - Cadastro de atividade de limpeza - O sistema deve permitir que, tanto o administrador quanto o supervisor, cadastrem novas atividades de limpeza.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_04 - Cadastro de empresa - O sistema deve permitir que o administrador cadastre a empresa terceirizada referente aos funcionários.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

RF_05 - Relatórios - O sistema deve permitir que os funcionários de limpeza criem relatórios após a conclusão de uma atividade.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

RF_06 - Solicitações - O sistema deve permitir que o administrador envie solicitações ao supervisor.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_07 - Login de usuário - O sistema deve permitir que todos os usuários cadastrados efetuem login utilizando suas credenciais (e-mail e senha) para acessar suas respectivas funcionalidades, de acordo com seu perfil (administrador, supervisor ou funcionário).

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_08 - Exibir os funcionários - O sistema deve conter uma tabela exibindo os funcionários cadastrados, disponível para visualização nas interfaces de administrador e supervisor.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

RF_09 - Exibir as atividades de limpeza - O sistema deve exibir as atividades de limpeza designadas para cada funcionário.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

RF_10 - Exibir os relatórios de atividades concluídas - O sistema deve exibir relatórios detalhando as atividades de limpeza concluídas por cada funcionário.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_11 - Validação de dados - O sistema deve realizar validação para cada dado nos campos de entrada, garantindo a integridade das informações fornecidas. As validações incluem verificar o formato correto para e-mail, além de validar a presença de campos obrigatórios.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_12 - Autenticação e autorização - O sistema deve implementar mecanismos de autenticação para garantir que apenas usuários registrados possam acessar o sistema. Além disso, deve fornecer autorização baseada em perfis de usuário, restringindo o acesso a funcionalidades específicas conforme o tipo de usuário (administrador, supervisor ou funcionário).

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RF_13 - Publicação de fotos antes e depois das atividades - O sistema deve permitir que os funcionários de limpeza publiquem fotos que representem o estado do ambiente antes e depois da execução de uma atividade de limpeza, vinculando essas imagens ao relatório de atividade correspondente.

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

2.3.2 Requisitos Não Funcionais

RNF_01 - Notificações - O sistema deve enviar notificações aos supervisores e aos funcionários sobre solicitações e designações de tarefas.

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

RNF_02 - Segurança - O sistema deve garantir a segurança dos dados pessoais dos usuários, utilizando tecnologias de criptografia para a segurança das informações.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RNF_03 - Usabilidade - O sistema deve ser de fácil entendimento para o usuário, oferecendo uma interface amigável, intuitiva e acessível.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RNF_04 - Escalabilidade - O sistema deve ser projetado para suportar o crescimento no número de usuários e atividades, podendo ser facilmente adaptado para atender a demandas futuras.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

RNF_05 - Desempenho - O sistema deve ser capaz de suportar múltiplas operações simultâneas sem comprometer o tempo de resposta, garantindo que o tempo de carregamento não ultrapasse 2 segundos para as principais funcionalidades.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

RNF_06 - Compatibilidade - O sistema deve ser acessível a partir de navegadores modernos e dispositivos móveis, oferecendo responsividade e compatibilidade com diferentes plataformas.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

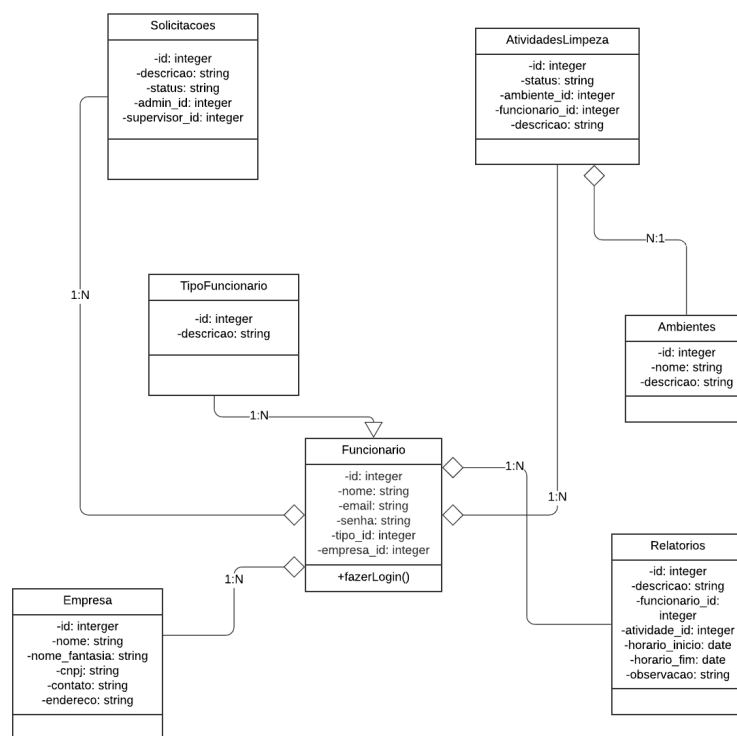
3 ANÁLISE E DESIGN

Nesta etapa do projeto, serão abordados os principais elementos de análise e design do sistema. Essa análise é fundamental para definir e alinhar os escopos do projeto e identificar os requisitos já abordados, utilizando *Unified Modeling Language* (UML) para gerar os diagramas do sistema.

3.1 Diagrama de Classes

“Bastante usado por engenheiros de software para documentar arquiteturas de software, os diagramas de classes são um tipo de diagrama da estrutura porque descrevem o que deve estar presente no sistema a ser modelado.” (LUCIDCHART, s.d.). Resumidamente, um diagrama de classes representa visualmente a estrutura de um sistema de software. Com base nisso, a figura 1 a seguir mostra a estrutura do projeto LimpAeh.

Figura 1 - Diagrama de Classes

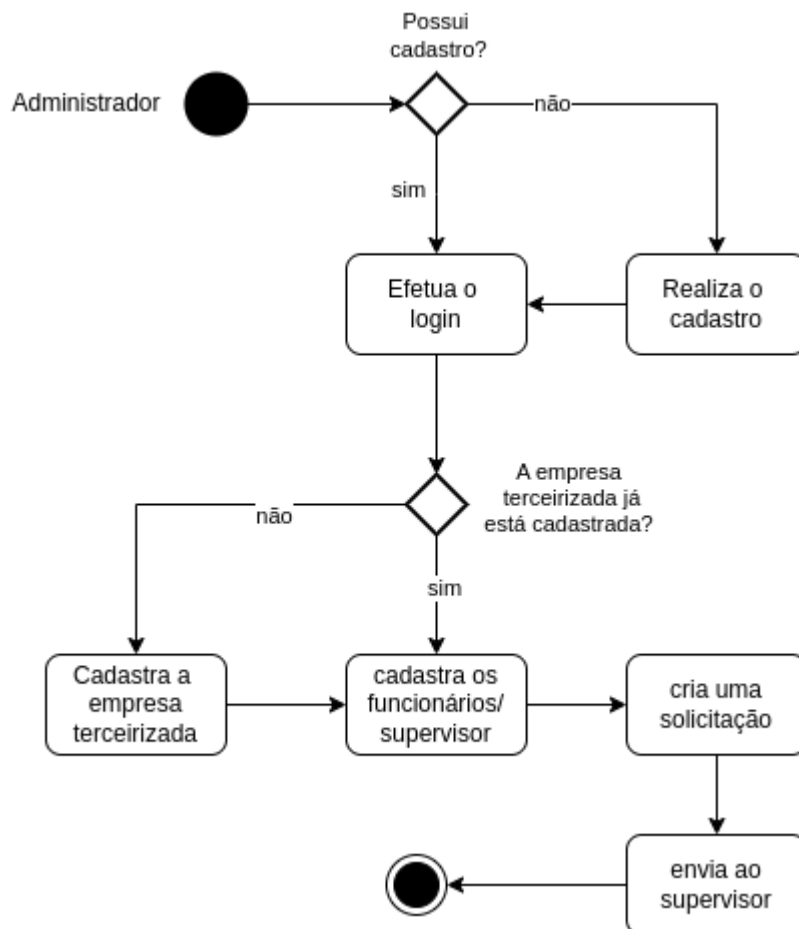


Fonte: Autoria própria (2025)

3.2 Diagrama de Atividades

“Na UML, um diagrama de atividade fornece uma visualização do comportamento de um sistema descrevendo a sequência de ações em um processo.” (IBM, 2021). Conforme afirma a International Business Machines Corporation (IBM), um diagrama de atividades é uma ferramenta visual poderosa usada para descrever o fluxo de trabalho ou o comportamento de um sistema. Na figura 2 a seguir, é possível visualizar como um administrador acessa e realiza as suas funções no sistema.

Figura 2 - Diagrama de Atividades



Fonte: Autoria própria (2025)

3.3 Modelo de Dados

“Modelagem de dados é o processo de criar uma representação visual, ou esquema, que define os sistemas de coleta e gerenciamento de informações de qualquer organização.” (Amazon Web Services [AWS], s.d.). No LimpAeh, a função primordial dos modelos de dados é apresentar visualmente as entidades, seus atributos e os relacionamentos que os ligam.

O modelo de dados tem papel fundamental no desenvolvimento de um software, pois oferece diversos benefícios ao sistema. A modelagem facilita a comunicação e a compreensão, além de garantir a organização e a consistência das informações.

3.3.1 Modelo Lógico da Base de Dados

O Modelo Lógico da Base de Dados define a estrutura conceitual dos dados, com foco nos relacionamentos e nas regras de negócios. Esse modelo foi baseado nos requisitos funcionais e não funcionais do sistema e é completamente independente do sistema de gerenciamento de banco de dados.

- **Tabela Ambientes**

Atributos: id, nome, descrição

Chave primária: id

- **Tabela Atividades de Limpeza**

Atributos: id, status, ambiente_id, funcionario_id, descricao

Chave primária: id

Chave estrangeira: ambiente_id, funcionario_id

- **Tabela Empresas**

Atributos: id, nome, nome_fantasia, cnpj, contato, endereco

Chave primária: id

- **Tabela Funcionários**

Atributos: id, nome, email, senha, tipo_id, empresa_id

Chave primária: id

Chave estrangeira: tipo_id, empresa_id

- **Tabela Relatórios**

Atributos: id, descricao, funcionario_id, atividade_id, horario_inicio, horario_fim, observacao

Chave primária: id

Chave estrangeira: funcionario_id, atividade_id

- **Tabela Solicitações**

Atributos: id, descricao, status, supervisor_id

Chave primária: id

Chave estrangeira: supervisor_id

- **Tabela Tipos de Funcionários**

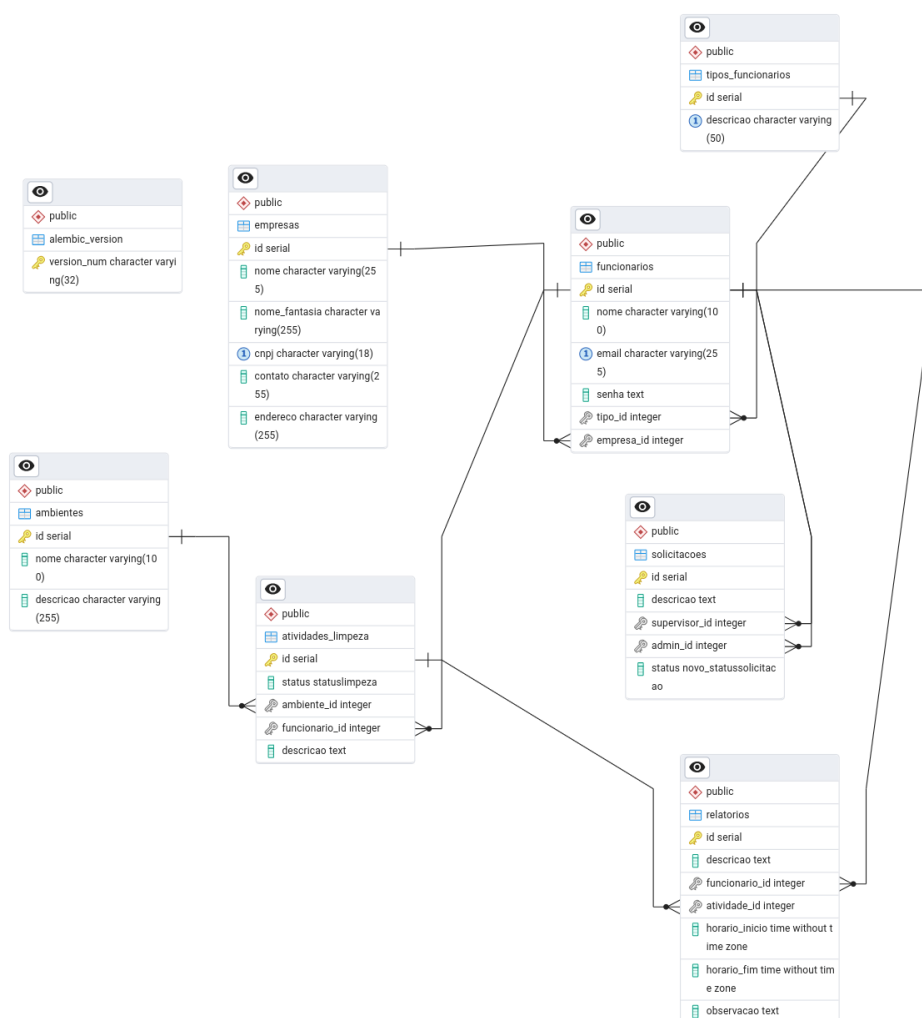
Atributos: id, descricao

Chave primária: id

3.3.2 Criação Física do Modelo de Dados

O Modelo Físico da Base de Dados representa a implementação do modelo conceitual. A modelagem do LimpAeh foi construída através de migrações no framework Flask, por meio da extensão Flask-Migrate. As migrações têm a função de traduzir o modelo lógico para físico, transformando o conceitual em scripts SQL compatíveis com o SGBD PostgreSQL, utilizado no projeto. Na figura abaixo, é possível visualizar a estrutura que o PgAdmin4 fornece, através da opção “ERD For Database”, acerca da modelagem física do banco de dados.

Figura 3 - Modelo Físico da Base de Dados



Fonte: PgAdmin4

3.3.3 Dicionário de Dados

O Dicionário de Dados é um documento essencial para detalhar os dados do modelo lógico e físico do sistema. Ele é responsável por descrever o significado, as características e a relação entre os dados. Nesta seção, serão apresentados os detalhes de cada atributo utilizado no banco de dados, incluindo seus tipos de dados, restrições e relacionamentos.

Ambientes

Campo	Descrição	Tipo
id	Identificador único de ambiente	Inteiro
nome	Nome do ambiente	Texto
descricao	Descrição sobre o local do ambiente	Texto

AtividadesLimpeza

Campo	Descrição	Tipo
id	Identificador único de atividades	Inteiro
descricao	Descreve o que deve ser feito na atividade	Texto
status	Representa o estado atual da atividade	Texto
ambiente_id	Identificar do ambiente em que a atividade será executada	Inteiro
funcionario_id	Identificador do funcionário que cumprirá a atividade	Inteiro

Empresas

Campo	Descrição	Tipo
id	Identificador único da empresa	Inteiro
nome	Nome da empresa	Texto
nome_fantasia	Nome fantasia da empresa	Texto
cnpj	CNPJ da empresa	Texto
contato	Contato da empresa	Texto
endereco	Endereço da empresa	Texto

Funcionários

Campo	Descrição	Tipo
id	Identificador único do funcionário	Inteiro
nome	Nome do funcionário	Texto
email	Email do funcionário	Texto
senha	Senha do funcionário	Texto
tipo_id	Identificador de tipo de funcionário	Inteiro
empresa_id	Identificador da empresa que o funcionário faz parte	Inteiro

Relatórios

Campo	Descrição	Tipo
id	Identificador único de relatórios	Inteiro

descricao	Descrição sobre o relatório	Texto
funcionario_id	Identificador do funcionário que fez o relatório	Inteiro
atividade_id	Identificador da atividade que foi realizada	Inteiro
horario_inicio	Marca o horário que o funcionário começou a atividade	Data
horario_fim	Marca o horário que o funcionário terminou a atividade	Data

Solicitações

Campo	Descrição	Tipo
id	Identificador único de solicitações	Inteiro
descricao	Descrição sobre a solicitação	Texto
status	Representa o estado atual da solicitação	Texto
supervisor_id	Identificador do supervisor que recebeu a solicitação	Inteiro

TiposFuncionários

Campo	Descrição	Tipo
id	Identificador único de ambiente	Inteiro
descricao	Descrição sobre o tipo de usuário	Texto

3.4 Ambiente de Desenvolvimento

O ambiente de desenvolvimento é a parte principal do projeto, pois é onde é definido as linguagens de programação que serão utilizadas e o banco de dados para o sistema. Além disso, também é no ambiente de desenvolvimento que são escolhidos os frameworks, tanto para o front-end quanto para o back-end.

Para isso, foi utilizado a poderosa IDE (Ambiente de Desenvolvimento Integrado) Visual Studio Code, que é uma ferramenta para desenvolvedores de software, que permite editar, depurar e executar tarefas em código. Ele é otimizado para a compilação e depuração de aplicativos web e de nuvem.

A ferramenta de versionamento utilizada foi o Git, que é um sistema que registra e controla as alterações feitas em um projeto, como código-fonte, um modelo ou um sistema. Ele permite rastrear o histórico de modificações e facilita a colaboração entre desenvolvedores.

Back-end:

A linguagem de programação utilizada para o desenvolvimento do back-end do sistema foi Python, uma linguagem de alto nível conhecida por sua clareza e simplicidade, além de oferecer recursos tecnológicos avançados e ser interpretada, o que facilita a portabilidade e a manutenção do código.

O framework utilizado no desenvolvimento do sistema foi o Flask, que é um micro framework de desenvolvimento web escrito em Python. No entanto, o principal recurso empregado foi a extensão Flask-RESTful, que permite a construção de API's RESTful, permitindo uma integração eficiente entre o front-end e o back-end do sistema.

O banco de dados utilizado foi o PostgreSQL, que é um sistema de gerenciamento de banco de dados relacional. O PostgreSQL é conhecido por sua robustez, extensibilidade e conformidade com padrões SQL. Ele oferece suporte avançado para operações transacionais, tipos de dados complexos e extensões personalizadas, tornando-se uma escolha ideal para sistemas de alta complexidade e desempenho

Front-end:

No front-end, a linguagem de programação utilizada foi o JavaScript, uma linguagem de alto nível e dinâmica, muito utilizada para o desenvolvimento web, além de ser uma linguagem interpretada.

Para a construção das interfaces de usuário, foi utilizada a biblioteca React, uma poderosa ferramenta para o desenvolvimento de componentes reutilizáveis e interfaces dinâmicas e responsivas.

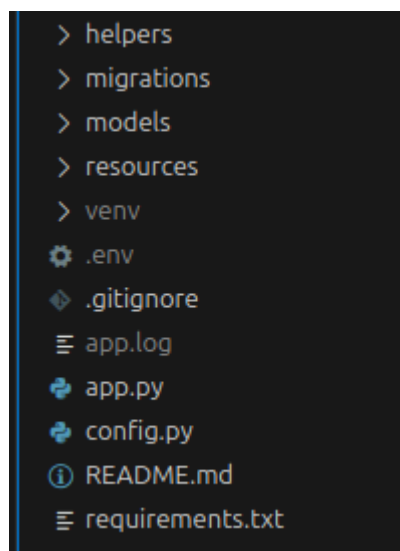
O framework escolhido para o design da interface foi o Material UI, que oferece uma coleção de componentes estilizados para React seguindo os princípios de design do Material Design do Google, proporcionando uma experiência visual consistente e moderna.

4 IMPLEMENTAÇÃO

A implementação do LimpAeh foi realizada utilizando duas estruturas de arquivos similares, uma para cada parte da aplicação: back-end e front-end. Para a parte interna do sistema, foi utilizado o padrão de estrutura de arquivos Flask. E para a parte externa, foi montado a estrutura de arquivos no padrão React.

Back-end:

Figura 4 - Arquivos do back-end



Fonte: Autoria própria

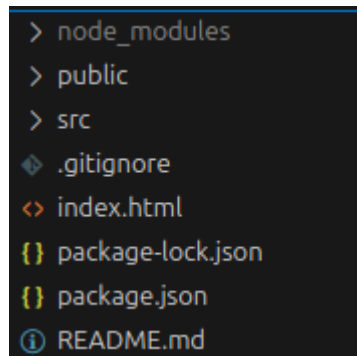
No diretório /limpaeh-back-end/ estão:

- /helpers: Armazena funções e classes utilitárias que são usadas em várias partes da aplicação.
- /migrations: No diretório *migrations* estão os arquivos de migração do banco de dados, utilizando Flask-Migrate.
- /models: Contém os arquivos responsáveis por modularizar o banco de dados.
- /resources: Implementa as operações CRUD (Create, Read, Update, Delete) para os principais recursos do sistema, facilitando a manipulação de dados.

- /venv: Módulo do Python usado para criar o ambiente virtual.
- /.env: Arquivo de configuração que armazena variáveis de ambiente, como chaves de API e credenciais do banco de dados.
- /.gitignore: Arquivo responsável por não permitir que alguns arquivos subam para a produção git.
- /app.log: Arquivo que registra os logs do sistema, útil para monitoramento e depuração de erros.
- /app.py: Arquivo principal, onde o sistema é inicializado e gerenciado.
- /config.py: Armazena as configurações globais do sistema, incluindo a URL de conexão com o banco de dados.
- /README.md: Documento de introdução e documentação básica do projeto, contendo informações sobre a configuração, execução e dependências da aplicação.
- /requirements.txt: Lista todas as dependências do projeto, facilitando a instalação e o gerenciamento de bibliotecas.

Front-end:

Figura 5 - Arquivos do front-end



Fonte: Autoria própria

No diretório /limpaeh-front-end/ estão:

- /node_modules: diretório criado pelo Node.js quando é instalado os pacotes ou bibliotecas usando o gerenciador de pacotes
- /public: Serve como um local onde recursos estáticos do projeto são armazenados.

- /src: Contém os componentes da aplicação, além do arquivo principal App.jsx.
- /,gitignore: Arquivo responsável por não permitir que alguns arquivos subam para a produção git.
- /index.html: Serve como a base onde a interface do usuário será renderizada pelo JavaScript
- /package-lock-json: É criado automaticamente quando você executa o comando npm install em um projeto Node.js. Ele complementa o package.json e tem funções importantes para a gestão de dependências.
- /package.json: Contém as dependências que foram instaladas para o projeto.
- /README.md: Documento de introdução e documentação básica do projeto, contendo informações sobre a configuração, execução e dependências da aplicação.

5 CONCLUSÃO

Dessa maneira, o LimpAeh surge como uma solução inovadora e tecnológica, apresentando vantagens técnicas e eficientes para as atividades diárias dos funcionários de limpeza do IFPB - Campus Guarabira-PB. Ao otimizar a comunicação entre gestores, supervisores e funcionários, o LimpAeh garante uma melhor qualidade na operação de serviços e na troca de informações.

Embora o sistema ofereça funcionalidades robustas para o gerenciamento de atividades, a plataforma ainda não possui sistema de notificação em tempo real e nem registro de imagens antes e depois da execução das tarefas de limpeza. A implementação dessas funcionalidades futuras pode ampliar a usabilidade da plataforma, aprimorando ainda mais a comunicação e a supervisão das atividades. Também vale ressaltar que a implementação de um novo tipo de usuário, um ‘solicitante’, que teria a função de reportar ao supervisor locais que não foram devidamente limpos, também está previsto como uma melhoria futura. Esse usuário poderia ser um docente ou algum técnico administrativo.

Atualmente o sistema está avançado, podendo realizar cadastro, edições e exclusões de dados. O sistema já consegue gerenciar solicitações, atividades e relatórios, podendo ser utilizado tanto pelos administradores quanto pelos funcionários do campus.

REFERÊNCIAS

AMAZON WEB SERVICES [AWS]. O que é modelagem de dados? Disponível em: <https://aws.amazon.com/pt/what-is/data-modeling/>. Acesso em: 18 de jan. 2025

BRASIL. Lei Geral de Proteção de Dados Pessoais (LGPD). 2018. Disponível em: <http://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd>. Acesso em: 16 de jan. 2025.

HYGIBRAS. Roteiro de limpeza escolar: técnicas e produtos corretos. 2024. Disponível em: https://www.hygibras.com/artigos/roteiro-limpeza-escolar/?utm_source=chatgpt.com#T2. Acesso em: 13 jan. 2025.

IBM. Diagrama de Atividades. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity>. Acesso em: 16 de jan. 2025. Activity diagrams

LUCIDCHART. O que é um diagrama de classe UML? Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 15 jan. 2025.

PACTO GLOBAL. Objetivos de Desenvolvimento Sustentável. 2023. Disponível em: <https://www.pactoglobal.org.br/ods>. Acesso em: 12 jan. 2025.