

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC ILZA NASCIMENTO PINTUS

APLICATIVO QUALIVITA

SÃO JOSÉ DOS CAMPOS
2024

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
Etec ILZA NASCIMENTO PINTUS

LUCAS DONIZETI DOS SANTOS
JOÃO VITOR PEREIRA DE JESUS
GUSTAVO RAMOS DA SILVA SANTOS

APLICATIVO QUALIVITA

SÃO JOSÉ DOS CAMPOS
2024

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC ILZA NASCIMENTO PINTUS

LUCAS DONIZETI DOS SANTOS
JOÃO VITOR PEREIRA DE JESUS
GUSTAVO RAMOS DA SILVA SANTOS

APLICATIVO QUALIVITA

SÃO JOSÉ DOS CAMPOS

2024

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
Etec ILZA NASCIMENTO PINTUS

LUCAS DONIZETI DOS SANTOS
JOÃO VITOR PEREIRA DE JESUS
GUSTAVO RAMOS DA SILVA SANTOS

APLICATIVO QUALIVITA

Trabalho de Conclusão de Curso apresentado
à Etec Ilza Nascimento Pintus como requisito
para obtenção do grau de Técnico em
Desenvolvimento de Sistemas, sob
orientação do Professor Gildárcio Sousa
Gonçalves.

SÃO JOSÉ DOS CAMPOS
2024

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC ILZA NASCIMENTO PINTUS

LUCAS DONIZETI DOS SANTOS
JOÃO VITOR PEREIRA DE JESUS
GUSTAVO RAMOS DA SILVA SANTOS

APLICATIVO QUALIVITA

Aprovado em ____/____/____ para obtenção do curso Técnico em Informática

BANCA EXAMINADORA:

Professor:

Assinatura _____

Professor:

Assinatura _____

Professor:

Assinatura _____

Professor:

Assinatura _____

RESUMO

O projeto Qualivita tem como objetivo fornecer informações sobre a qualidade do ar para os usuários, por meio de um dispositivo medidor da poluição do ar conectado a um aplicativo. Essas informações são transmitidas ao usuário através de um mapa exibido na tela principal, que demarca a localização do sensor com um botão que, ao ser clicado, exibe diferentes dados captados pelo dispositivo medidor, como a condição da qualidade, a temperatura e a umidade do ar. Portanto, a aplicação possui funções necessárias para o fornecimento de dados sobre o ar, podendo ser utilizada em diversas situações do dia a dia dos usuários de maneira direta e objetiva. Esses fatores contribuem para a divulgação desse tipo de informação, facilitando seu acesso e podendo impactar de forma direta ou indireta o cotidiano do usuário. As tecnologias usadas para o desenvolvimento do software foram Visual Studio Code, React Native (Javascript), Firebase (noSQL), Git, GitHub e Arduino IDE (C++).

Palavras-chave: qualidade, localização, sensor, informações, objetivo

ABSTRACT

The Qualivita project aims to provide information about air quality to users through an air pollution monitoring device connected to an application. This information is transmitted to the user via a map displayed on the main screen, which marks the sensor's location with a button. When clicked, this button shows different data collected by the monitoring device, such as air quality condition, temperature, and humidity. The application thus has essential features for providing air data, making it usable in various day-to-day situations in a straightforward and objective manner. These factors contribute to spreading this type of information, facilitating access to it and potentially impacting the user's daily life, directly or indirectly. The technologies used for the software development were Visual Studio Code, React Native (JavaScript), Firebase (NoSQL), Git, GitHub, and Arduino IDE (C++).

Word Keys: quality, location, sensor, information, objective

Agradecimentos

Agradecemos aos nossos amigos, familiares, professores e guias online por toda ajuda prestada para o desenvolvimento e conclusão desse projeto.

Pensamento

“Nenhum trabalho de qualidade pode ser feito sem concentração e autossacrifício, esforço e dúvida”

Max Beerbohm

Lista de ilustrações

Figura 1: Diagrama de caso e uso	34
Figura 2: Diagrama de classe.....	35
Figura 3: Logotipo	36
Figura 4: Telas de introdução.....	36
Figura 5: Tela de cadastro.....	37
Figura 6: Tela de login.....	38
Figura 7: Tela principal com e sem imagens do satélite.....	38
Figura 8: Tela com as informações dos sensores	39
Figura 9: Tela com os dados dos sensores.....	39
Figura 10: Tela de configurações	40
Figura 11: Protótipo do sensor	40
Figura 12: Modelo Conceitual.....	42
Figura 13: Modelo Lógico	42
Figura 14: Tabela Usuários	43
Figura 15: Tabela Sensores	43
Figura 16: Tabela Administradores	44
Figura 17: Codificação do Login.....	46
Figura 18: Codificação do Cadastro	47
Figura 19: Codificação da Redefinição de Senha	48
Figura 20: Codificação do Mapa 1.....	48
Figura 21: Codificação do Mapa 2.....	49
Figura 22: Codificação das Informações	50
Figura 23: Codificação do Histórico.....	51
Figura 24: Codificação das Rotas	52

Figura 25: Codificação das Informações do Usuário	53
Figura 26: Codificação do Sensor 1	53
Figura 27: Codificação do Sensor 2	54
Figura 28: Tela de Splash	55
Figura 29: Tela de Introdução 1	56
Figura 30: Tela de Introdução 2	56
Figura 31: Tela de Introdução 3	57
Figura 32: Tela de Login.....	58
Figura 33: Tela de Cadastro.....	59
Figura 34: Tela de Recuperação de Senha.....	60
Figura 35: Tela do Mapa	61
Figura 36: Tela do Mapa Satélite	62
Figura 37: Tela de Informações.....	63
Figura 38: Tela de Histórico	64
Figura 39: Tela de Configurações	65
Figura 40: Tela de Informações do Usuário	66
Figura 41: Tela de Como Nossos Sensores Funcionam	67
Figura 42: Telas de Privacidade e Termos de Uso	68

Lista de tabelas

Tabela 1: Coletar os dados pelo Esp32.....	28
Tabela 2: Mapa funcional.....	28
Tabela 3: Integrar o Esp32 e seus módulos	29
Tabela 4: Cadastrar usuário	29
Tabela 5: Tela de login	29
Tabela 6: Exibir dados dos sensores em geral	30
Tabela 7: Histórico de dados coletados.....	30
Tabela 8: Notificar.....	30
Tabela 9: Redefinir Senha	31
Tabela 10: Configurações.....	31
Tabela 11: Permissão para acessar a localização do usuário	31
Tabela 12: Conexão à internet.....	32
Tabela 13: O aplicativo rodará no sistema operacional Android.....	32
Tabela 14: Conexão ao banco de dados	32
Tabela 15: Uso de API.....	33
Tabela 16: Suporte ao volume de dados	33
Tabela 17: Consumo de energia.....	33
Tabela 18: Cronograma	35
Tabela 19: Entidade Usuários.....	44
Tabela 20: Entidade Sensores.....	44
Tabela 21: Entidade Administradores	45

Sumário

1 INTRODUÇÃO	16
1.1 Identificação do Problema e consequências.....	16
1.2 Identificação da Solução, benefícios e principais funcionalidades.....	16
1.3 Público-alvo	17
1.4 Trabalhos Correlatos (Concorrentes).....	17
1.5 Descrição das Metodologias aplicadas.....	18
1.6 Estudo de Viabilidade	19
1.7 Objetivo Geral.....	20
1.7.1 Objetivo específico	20
1.8 Resultados Esperados.....	20
2 DESENVOLVIMENTO	21
2.1 Referencial Teórico.....	21
Pnuma	21
IQair	21
CETESB	21
2.1.2 Tecnologias Utilizadas	22
2.2 Ciclo de Desenvolvimento do Sistema.....	27
2.2.1 Anteprojeto ou Análise de Projeto	27
2.2.1.1 Levantamento de Requisitos	28
2.3 Projeto	34
2.3.1 Diagrama de caso de uso, classe e Cronograma	34
2.3.2 Design do Sistema (Telas e Navegabilidade)	36
2.3.2.1 Estudo das cores, imagens, fontes e identidade visual a serem aplicadas ao sistema	41

2.4 Projeto de Banco de Dados	42
2.4.1 Modelo Conceitual e Lógico	42
2.4.2 Modelo Físico (Não Relacional)	43
2.4.3 Dicionário de Dados	44
2.5 Codificação do sistema.....	46
3 RESULTADOS.....	55
4 CONCLUSÃO	69
5 REFERÊNCIAS.....	70
APÊNDICE – Plano de Teste.....	73
1 INTRODUÇÃO.....	77
2 SOBRE O PROJETO.....	77
3 OBJETIVOS.....	77
3.1 Sobre o sistema	78
3.2 Escopo	78
4 REQUISITOS A SEREM TESTADOS – REQUISITOS DE QUALIDADE ..	79
4.1 Teste de Banco de Dados.....	79
4.2 Teste Funcional.....	79
4.3 Teste da Interface de Usuário	80
4.4 Teste de Desempenho	80
4.5 Teste de Estresse	81
4.6 Teste de Segurança	81
4.7 Teste de Instalação e Compatibilidade	81
4.8 Teste de Usabilidade.....	82
5 REQUISITOS DO CLIENTE	82
6 REQUISITOS DO SOFTWARE	83
6.1 Requisitos Funcionais	83
6.2 Requisitos Não Funcionais.....	86

7 ESTRATÉGIAS E FERRAMENTAS DE TESTE	89
7.1 Teste do Banco de Dados.....	89
7.2 Teste Funcional.....	89
7.3 Teste da Interface do Usuário	90
7.4 Teste de Performance.....	90
7.5 Teste de Carga	90
7.6 Teste de Segurança e Controle de Acesso.....	91
7.7 Teste de Instalação	91
8 EQUIPE E INFRAESTRUTUTRA	91
9 CRONOGRAMA	92
10 CONCLUSÃO	93

1 INTRODUÇÃO

1.1 Identificação do Problema e consequências

Com o crescimento urbano e o aumento da poluição, a busca por uma vida mais saudável torna-se cada vez mais urgente. Segundo a Organização Mundial da Saúde (OMS), a poluição do ar é responsável por cerca de 7 milhões de mortes anuais em todo o mundo.

De acordo com o pneumologista João Batista, do Hospital Universitário da Universidade Federal do Maranhão (HU-UFMA), a exposição a ambientes poluídos pode agravar ou até mesmo desencadear doenças cardiovasculares, um aspecto crucial a ser considerado na escolha de um local para morar.

Apesar disso, ainda faltam recursos simples, como aplicativos que disponibilizem informações sobre a qualidade do ar em diferentes regiões da cidade. Essa lacuna dificulta o acesso das pessoas a dados importantes, levando muitas a desistirem de buscar ambientes mais saudáveis e a conviverem com elevados níveis de poluição.

1.2 Identificação da Solução, benefícios e principais funcionalidades

Pensando em todos os malefícios que uma vida sujeita à poluição traz, uma possível solução ideal seria uma forma simples para que pessoas comuns consigam ter informações importantes, como a qualidade do ar em pontos específicos da cidade, de forma mais detalhada.

Com isso, a criação de um aplicativo que dê as informações sobre o nível de poluição do ar na cidade com certeza seria uma excelente ajuda para pessoas que procuram uma vida melhor, mais natural e livre dos males causados pela sociedade atual.

Os usuários poderão utilizar os dados fornecidos para fazer escolhas mais assertivas sobre a qualidade de vida. Por exemplo, uma família que vai se mudar de casa e tem um membro com problemas respiratórios poderia usar o aplicativo para escolher um local com melhor qualidade do ar na cidade, proporcionando uma vida mais saudável para todos. Outro bom uso da aplicação seria para uma pessoa que gosta de fazer caminhadas e deseja encontrar os melhores locais para aproveitar o ar puro. Com o aplicativo, essa pessoa conseguiria planejar uma rota com pontos ideais para maior bem-estar durante a caminhada.

As utilidades desse recurso não se resumem a apenas esses exemplos, há várias outras maneiras de se utilizar as informações que serão fornecidas e com isso com certeza muitos problemas a respeito da poluição urbana e qualidade de vida seriam solucionados.

1.3 Público-alvo

O público-alvo deste projeto são pessoas que necessitam saber o estado da qualidade do ar no ambiente em que vivem, frequentam ou que até mesmo onde planejam começar a viver, no entanto o aplicativo será útil até mesmo para usuários que somente tem curiosidade de saber sobre essas informações.

Todo esse projeto visa ajudar pessoas que procuram ter uma vida mais natural e saudável, auxiliando na escolha de um ambiente com menos poluição e desse modo proporcionando uma qualidade de vida melhor.

1.4 Trabalhos Correlatos (Concorrentes)

De acordo com a pesquisa feita existem aplicações e sites que fazem a coleta e a exposição desses dados de maneira semelhante ao projeto apresentado neste trabalho, sendo elas o aplicativo da CETESB (Companhia Ambiental do Estado de São Paulo), o PNUMA uma plataforma sobre o

monitoramento da qualidade do ar feita em parceria pela ONU (Organização das Nações Unidas) e pela IQ Air, além do CanAirIO.

O aplicativo da Companhia Ambiental do Estado de São Paulo (CETESB) monitora a qualidade do ar por meio de alguns pontos espalhados em certos pontos do estado, mostrando no aplicativo não apenas as informações coletadas, mas também informações de saúde relacionadas à qualidade do ar.

Já o PNUMA plataforma da ONU (Organização das Nações Unidas) e da IQ Air mostram no mapa as informações coletadas pelos sensores, a localização desses dispositivos, e cobre com mapa de calor informando a qualidade do ar as áreas monitoradas, embora a proposta do PNUMA seja similar à do projeto o PNUMA também pretende usar desses dados para entender e realizar melhorias em diversas áreas a fim de diminuir o impacto causados por elas na poluição do ar, também pretende capacitar governos, comunidades e cidadãos para ajudar n.

O CanAirIO criado por Álvaro Antonio Vanegas e Daniel Bernal, desenvolveu um sensor caseiro de qualidade de ar que é basicamente a ideia deste trabalho de conclusão de curso, porém os dados obtidos pelo sensor da dupla colombiana são transmitidos por uma tela led acoplada ao sensor, enquanto a proposta deste trabalho é mostrar os dados coletados pelo dispositivo por meio de um mapa que será visível no aplicativo.

1.5 Descrição das Metodologias aplicadas

As metodologias e métodos ágeis usados nesse projeto serão o Scrum e Kanban, devido à versatilidade e agilidade que ambas oferecem, além de serem os métodos mais adotados no mercado atualmente, sendo usados por empresas como a Telefônica Vivo, NeoAssist, Pluga, Microsoft, Siemens, Globo, OUL e Google por exemplo.

O Scrum é uma metodologia ágil, ou seja, é um conjunto de técnicas de desenvolvimento de inovações que preza a velocidade nos processos envolvidos

no gerenciamento dinâmico de processos, sendo composto por 4 metodologias principais:

- Os indivíduos e as interações são mais importantes que os processos e as ferramentas;
- Um programa funcional e prático é mais relevante que uma documentação completa;
- É muito mais relevante a colaboração e o contato com o cliente durante todo o processo do que apenas a negociação de contratos e toda a burocracia envolvida. O que é crucial é satisfazer o cliente;
- Responder às mudanças e adaptações no processo é mais relevante do que seguir um plano específico.

Já o Kanban é um método de gestão de projetos utilizado por empresas de diferentes setores tendo como objetivo aumentar a eficiência dos processos produtivos, reduzindo o desperdício de tempo por meio da organização do fluxo de trabalho, sendo baseado em princípios como visualização do fluxo, limitação do trabalho em progresso, estabelecimento de ordens de prioridade e melhoria contínua.

1.6 Estudo de Viabilidade

Após analisar os meios para realização do projeto, várias pesquisas foram feitas para saber o que seria necessário. Para que toda essa ideia seja completa será preciso conhecimento de circuitos elétricos para utilização de um sensor físico para medição da qualidade do ar junto de uma placa ESP32, felizmente existem vários guias e tutoriais para integração deste circuito e até mesmo de sua codificação.

Outro requisito muito importante será o conhecimento sobre algumas linguagens de programação, pois isso será fundamental para toda criação do aplicativo tanto para a criação da interface quanto para a parte lógica.

E por fim a integração de todos os sistemas, para que tudo esteja pronto para o usuário. Pensando em todos os requisitos para produzir um aplicativo deste porte e depois de muitas análises foi concluído que a equipe tem capacidade para realizar esta aplicação.

1.7 Objetivo Geral

Desenvolver um aplicativo que monitore a qualidade do ar em diferentes pontos da cidade, visando a conscientização e melhoria na qualidade de vida dos usuários do sistema.

1.7.1 Objetivo específico

- Avaliar formas de conscientizar o usuário de como seus hábitos de vida podem influenciar nos dados apresentados e como isso pode afetá-lo;
- Pesquisar os locais da cidade que necessitam de um ponto de monitoramento;
- Analisar diferentes métodos para integração do dispositivo ao aplicativo.

1.8 Resultados Esperados

Fazer um sistema que por meio da monitoração da qualidade do ar permita a melhoria da qualidade de vida dos usuários e a fiscalização do meio ambiente no local em que o dispositivo estiver instalado.

2 DESENVOLVIMENTO

2.1 Referencial Teórico

Pnuma

Programa das Nações Unidas para o Meio Ambiente (Pnuma) é a principal autoridade ambiental global sendo responsável por determinar a agenda internacional do tema, promovendo a implementação da dimensão ambiental do desenvolvimento sustentável no Sistema das Nações Unidas servindo como um defensor do meio ambiente no mundo.

O PNUMA possui como missão prover a liderança, encorajando parcerias na proteção ao meio ambiente com o intuito de melhorar a qualidade de vida de pessoas sem comprometer as futuras gerações de diversos países em torno do mundo.

IQair

A IQair é uma empresa Suíça que desde 1963 atua na qualidade do ar com o intuito de fornecer dados sobre por meio de soluções tecnológicas, capacitando indivíduos, organizações e comunidades com o objetivo desses indivíduos poderem respirar um ar mais limpo melhorando sua qualidade de vida.

CETESB

A Companhia Ambiental do Estado de São Paulo (CETESB), agência ambiental do Governo do Estado de São Paulo, é responsável pelo licenciamento, controle e fiscalização de atividades poluidoras, além de avaliar e monitorar a qualidade ambiental, garantindo que os padrões de qualidade das águas, do ar e do solo sejam respeitados em todo o estado. A agência também assegura a preservação dos recursos naturais e a manutenção da saúde pública.

Além de suas funções fiscalizadora e licenciadora, incluindo atividades listadas no Anexo C da Convenção de Estocolmo, a CETESB se dedica à gestão

de resíduos sólidos e áreas contaminadas. A instituição dispõe de modernos laboratórios equipados com tecnologia de ponta, realizando análises dentro dos rigorosos padrões de Garantia de Qualidade e Controle de Qualidade (QA/QC), conforme a norma da Organização Internacional para Normalização / Comissão Eletrotécnica Internacional (ISO/IEC) 17025:2005. Seu escopo de acreditação abrange cerca de 900 ensaios ambientais, incluindo análises orgânicas, inorgânicas, microbiológicas, ecotoxicológicas e de emissões de veículos. Oferece ainda serviços de calibração e ensaios de proficiência Inter laboratoriais.

A CETESB possui o único laboratório público na América Latina para análises de dioxinas e furanos, equipado com espectrômetro de massa de alta resolução para análises refinadas. Paralelamente, presta serviços e suporte técnico a órgãos governamentais, Ministério Público e setor privado, mantendo relações com agências de cooperação internacional de diversos países. A agência intervém em emergências ambientais e realiza treinamentos especializados pela Escola Superior da CETESB.

Com 50 anos de experiência, a CETESB é uma das cinco instituições mundiais da Organização Mundial da Saúde (OMS) para abastecimento de água e saneamento, além de referência do Programa das Nações Unidas para o Desenvolvimento (PNUD) para resíduos perigosos na América Latina. Participa da Rede Latino-Americana de Prevenção e Gerenciamento de Áreas Contaminadas (ReLASC), apoia o Mecanismo de Desenvolvimento Limpo (MDL) do Protocolo de Quioto, colabora com o Programa Estadual de Mudanças Climáticas (PROCLIMA) e coordena a Rede Pan-Americana de Informação em Saúde Ambiental (REPIDISCA).

2.1.2 Tecnologias Utilizadas

- **Plataforma Arduino IDE**

A plataforma Arduino IDE é um ecossistema de hardware e software que facilita a criação de projetos eletrônicos. O hardware consiste em microcontroladores de placa única e kits de desenvolvimento, que podem ser programados para interagir com uma variedade de sensores e atuadores. O

software inclui o Arduino IDE, um ambiente de desenvolvimento integrado, que utiliza uma linguagem de programação baseada em C/C++. O Arduino é amplamente utilizado em projetos educacionais, prototipagem rápida e no desenvolvimento de produtos comerciais devido à sua simplicidade e versatilidade.

- **Ambiente de desenvolvimento Visual Studio Code**

Visual Studio Code (VS Code) é um editor de código-fonte desenvolvido pela Microsoft, lançado em 2015. É um ambiente de desenvolvimento leve, mas poderoso, que suporta uma ampla gama de linguagens de programação. Entre suas características destacam-se o suporte à depuração, controle de versionamento Git integrado, e uma vasta coleção de extensões que permitem personalizar e expandir suas funcionalidades. VS Code é especialmente popular entre desenvolvedores web e de software devido à sua flexibilidade e robustez.

- **Plataforma de desenvolvimento Expo**

Expo é uma plataforma de desenvolvimento que facilita a criação de aplicativos móveis com React Native. Ela fornece um conjunto de ferramentas e serviços que simplificam o desenvolvimento, a construção e a implantação de aplicativos. Expo permite que os desenvolvedores escrevam aplicativos em JavaScript/TypeScript e utilizem APIs de dispositivos móveis sem a necessidade de configuração nativa complexa. É amplamente utilizado por sua capacidade de acelerar o desenvolvimento de aplicativos móveis multiplataforma.

- **Linguagem de programação Javascript**

JavaScript é uma linguagem de programação de alto nível, interpretada e baseada em eventos, amplamente utilizada no desenvolvimento web. Originalmente criada para adicionar interatividade às páginas web, hoje JavaScript é uma linguagem versátil que pode ser executada tanto no lado do cliente quanto no servidor, graças a ambientes como Node.js. Com a introdução do ECMAScript, JavaScript passou a ter uma padronização que facilita a interoperabilidade entre diferentes plataformas e navegadores.

- **Framework Javascript React Native**

React Native é um framework de código aberto desenvolvido pelo Facebook que permite a criação de aplicativos móveis usando JavaScript e React. Ele permite que os desenvolvedores criem aplicativos nativos para iOS e Android utilizando uma única base de código, compartilhando grande parte da lógica entre as plataformas. React Native combina a facilidade de desenvolvimento web com a performance de aplicativos nativos, sendo uma escolha popular para o desenvolvimento de aplicativos móveis modernos.

- **Framework Javascript Node js**

Node.js é um ambiente de execução JavaScript que permite a execução de código JavaScript no lado do servidor. Desenvolvido sobre o motor V8 do Google Chrome, Node.js é conhecido por seu desempenho e escalabilidade. Ele utiliza um modelo de E/S não bloqueante e orientado a eventos, ideal para a construção de aplicações de rede e servidores web. Node.js é amplamente adotado no desenvolvimento de aplicativos web e APIs devido à sua eficiência e robustez.

- **Ferramenta para versionamento de código GIT**

Git é um sistema de controle de versão distribuído, criado por Linus Torvalds em 2005. Ele permite que desenvolvedores rastreiem alterações no código-fonte, colaborem de maneira eficiente e mantenham um histórico detalhado de modificações. Git é conhecido por sua robustez, flexibilidade e desempenho, sendo a ferramenta de controle de versão mais utilizada no desenvolvimento de software atualmente. Ele facilita a colaboração em equipes de desenvolvimento distribuídas e suporta fluxos de trabalho complexos de desenvolvimento.

- **Ferramenta para compartilhamento de código GitHub**

GitHub é uma plataforma de hospedagem de código-fonte baseada em Git, adquirida pela Microsoft em 2018. Ela fornece uma interface web amigável para repositórios Git, facilitando a colaboração entre desenvolvedores. GitHub oferece recursos como gerenciamento de projetos, revisão de código, integração contínua, e documentação, além de uma comunidade vibrante de

desenvolvedores. A plataforma é amplamente utilizada para projetos de código aberto e comerciais, permitindo a colaboração eficiente e o compartilhamento de conhecimento.

- **Pacote Office**

O Pacote Office da Microsoft é essencial para projetos, oferecendo ferramentas que facilitam a criação, edição, organização e apresentação de informações. O Microsoft Word é utilizado para documentos formais, enquanto o Excel é indispensável para a análise de dados. O PowerPoint ajuda na comunicação de ideias através de apresentações visuais, e o Outlook organiza a comunicação e as atividades do projeto. O OneNote, por sua vez, é ideal para a tomada de notas e organização de informações.

Segundo um estudo da TechRepublic, o uso do Word pode aumentar a produtividade em até 30% devido às suas ferramentas de automação e colaboração.

Essas ferramentas aumentam a produtividade, precisão e eficiência da equipe, contribuindo significativamente para o sucesso do projeto. Estudos mostram que o uso do Word pode aumentar a produtividade em até 30%, enquanto o Excel é crucial para decisões baseadas em dados. Apresentações bem estruturadas no PowerPoint podem melhorar a retenção de informações, e o Outlook pode aumentar a eficiência da equipe em até 25%. O OneNote é essencial para a organização pessoal e profissional, integrando-se perfeitamente com os outros aplicativos do Pacote Office.

- **Firebase**

O Google Firebase é uma plataforma criada pelo Google para auxiliar no desenvolvimento e gerenciamento de aplicativos móveis e web de maneira mais eficiente. Ela disponibiliza uma série de ferramentas e serviços que facilitam tanto o desenvolvimento quanto a análise e o gerenciamento de aplicativos.

Entre os recursos oferecidos pelo Firebase estão o banco de dados, que inclui o Realtime Database e o Firestore, permitindo o armazenamento e a sincronização de dados em tempo real. A plataforma também fornece soluções

de autenticação para usuários, com opções como login por senha, número de telefone e provedores de identidade como Google e Facebook. Além disso, o Firebase permite a hospedagem de aplicativos web com segurança e praticidade, e oferece funções em nuvem que possibilitam a execução de código backend em resposta a eventos, sem a necessidade de gerenciar servidores.

Outro ponto importante do Firebase é a integração com o Google Analytics, que ajuda a entender o comportamento dos usuários e a otimizar a experiência do aplicativo. A plataforma também conta com serviços de notificações push, permitindo o envio de alertas e mensagens para engajar os usuários.

Com sua abordagem escalável e flexível, o Firebase permite que os desenvolvedores se concentrem no desenvolvimento de aplicativos de alta qualidade, sem a preocupação com a infraestrutura subjacente.

- **Firebase Firestore**

O Firestore é um banco de dados NoSQL baseado em documentos. Diferente de um banco de dados SQL, não utiliza tabelas ou linhas; os dados são organizados em documentos, que estão agrupados em coleções.

Cada documento é composto por pares chave-valor, e o Firestore é projetado para gerenciar grandes coleções de documentos pequenos.

Todos os documentos devem estar contidos em coleções, e esses documentos podem incluir subcoleções e objetos aninhados. Os campos podem ser primitivos, como strings, ou mais complexos, como listas.

Coleções e documentos no Firestore são criados de forma implícita. Basta adicionar dados a um documento em uma coleção; se a coleção ou o documento não existir, o Firestore os criará automaticamente.

- **Firebase Authentication**

A maioria dos aplicativos precisa identificar seus usuários. Ter essa informação permite que um app armazene os dados na nuvem de forma segura e ofereça uma experiência personalizada em todos os dispositivos do usuário.

O Firebase Authentication fornece serviços de back-end, SDKs intuitivos e bibliotecas de interface do usuário prontas para facilitar a autenticação dos usuários no seu aplicativo. Ele suporta autenticação por meio de senhas, números de telefone e provedores de identidade federados, como Google, Facebook e Twitter, entre outros.

Além disso, o Firebase Authentication se integra a outros serviços do Firebase e utiliza padrões do setor, como OAuth 2.0 e OpenID Connect, o que facilita a integração com seu back-end personalizado.

- **Banco de dados Firebase Real Time Database**

O Firebase Realtime Database é um banco de dados baseado na nuvem. Os dados são armazenados em formato JSON e sincronizados em tempo real para todos os clientes conectados. Ao desenvolver aplicativos multiplataforma com nossos SDKs para Apple, Android e JavaScript, todos os seus usuários compartilham uma única instância do Realtime Database e recebem automaticamente as atualizações com os dados mais recentes.

2.2 Ciclo de Desenvolvimento do Sistema

2.2.1 Anteprojeto ou Análise de Projeto

No planejamento do aplicativo de monitoramento da qualidade do ar, foram estabelecidos objetivos voltados para aprimorar a experiência dos usuários, proporcionando um acompanhamento preciso e detalhado das informações captadas pelo sensor desenvolvido. O aplicativo exibe em um mapa a localização exata do sensor, facilitando a visualização em tempo real dos dados ambientais. Esses recursos são essenciais para oferecer uma experiência completa e informativa, promovendo um entendimento mais abrangente da qualidade do ar na região monitorada.

2.2.1.1 Levantamento de Requisitos

Requisitos Funcionais

Relacionados as atividades que o próprio sistema realizará, definindo as funções que cada tipo de usuário terá dentro do sistema ao ser acessado, alguns requisitos podem ser essenciais ou desejáveis, dependendo da necessidade deles no sistema.

Tabela 1: Coletar os dados pelo Esp32

Identificador	Nome	Tipo	
RF01	Coletar os dados pelo Esp32	✓ Essencial	Desejável
Descrição			
Nosso aplicativo irá coletar as informações sobre a qualidade de ar fornecida pelo Esp32 para passar ao usuário.			

Fonte: Elaborado pelos autores

Tabela 2: Mapa funcional

Identificador	Nome	Tipo	
RF02	Mapa funcional	✓ Essencial	Desejável
Descrição			
O sistema deve conter um mapa funcional com os dados dos sensores			

Fonte: Elaborado pelos autores

Tabela 3: Integrar o Esp32 e seus módulos

Identificador	Nome	Tipo	
RF03	Integrar o Esp32 e seus módulos	✓ Essencial	Desejável
Descrição			
O Esp32 será necessário como forma de integração dos sensores mq-135, mq-7 e DHT11 ao aplicativo, os módulos vão coletar os dados e o esp32 será responsável pela exibição e exportação para o MySQL			

Fonte: Elaborado pelos autores

Tabela 4: Cadastrar usuário

Identificador	Nome	Tipo	
RF04	Cadastrar usuário	✓ Essencial	Desejável
Descrição			
Deverá ser fornecido uma tela de cadastro para o usuário, o cadastro no nosso aplicativo será apenas para identificação, pois não terá utilidade prática.			

Fonte: Elaborado pelos autores

Tabela 5: Tela de login

Identificador	Nome	Tipo	
RF05	Tela de login	✓ Essencial	Desejável
Descrição			
A tela de login do aplicativo será necessária para identificar um usuário já existente, permitindo a entrada no aplicativo, nesta tela o usuário irá inserir seus dados e depois o login será possível, caso ele ainda não exista no banco de dados basta voltar a tela de cadastro			

Fonte: Elaborado pelos autores

Tabela 6: Exibir dados dos sensores em geral

Identificador	Nome	Tipo	
RF6	Exibir dados dos sensores em geral	✓ Essencial	Desejável
Descrição			
A aplicação também vai exibir as informações sobre o ar coletadas em uma tela com a informação de vários sensores juntas, como uma lista.			

Fonte: Elaborado pelos autores

Tabela 7: Histórico de dados coletados

Identificador	Nome	Tipo	
RF7	Tela para visualização de outros dados coletados anteriormente	✓ Essencial	Desejável
Descrição			
Tela onde os usuários poderão visualizar as informações antigas do mesmo sensor.			

Fonte: Elaborado pelos autores

Tabela 8: Notificar

Identificador	Nome	Tipo	
RF8	Notificar	Essencial	✓ Desejável
Descrição			
O usuário deve receber uma notificação caso a qualidade do ar mais próxima a ele estiver em nível crítico			

Fonte: Elaborado pelos autores

Tabela 9: Redefinir Senha

Identificador	Nome	Tipo	
RF9	Redefinir Senha	✓ Essencial	Desejável
Descrição			
Tela para o usuário redefinir sua senha em caso de perda.			

Fonte: Elaborado pelos autores

Tabela 10: Configurações

Identificador	Nome	Tipo	
RF10	Configurações	Essencial	✓ Desejável
Descrição			
Tela de configurações onde usuário poderá ver informações de seu cadastro e o funcionamento dos sensores.			

Fonte: Elaborado pelos autores

Requisitos não funcionais

Os requisitos Não Funcionais determinam o comportamento do sistema. Eles podem decidir qual o alcance de eficácia do sistema e é dividido em requisitos Essencial e Desejável como dito anteriormente nos requisitos funcionais.

Tabela 11: Permissão para acessar a localização do usuário

Identificador	Nome	Tipo	
RNF01	Permissão para acessar a localização do usuário	Essencial	✓ Desejável
Descrição			
Permissão do usuário para acessar sua localização por meio de seu dispositivo.			

Fonte: Elaborado pelos autores

Tabela 12: Conexão à internet

Identificador	Nome	Tipo	
RNF02	Conexão à internet	✓ Essencial	Desejável
Descrição			
Será necessário o usuário estar conectado à internet.			

Fonte: Elaborado pelos autores

Tabela 13: O aplicativo rodará no sistema operacional Android

Identificador	Nome	Tipo	
RNF03	O aplicativo rodará apenas no Android	✓ Essencial	Desejável
Descrição			
O usuário deverá ter um dispositivo com sistema operacional Android para usar o aplicativo e monitorar a qualidade de ar de sua região.			

Fonte: Elaborado pelos autores

Tabela 14: Conexão ao banco de dados

Identificador	Nome	Tipo	
RNF04	Conexão ao banco de dados	✓ Essencial	Desejável
Descrição			
O usuário e o sensor deverão estar conectados ao banco de dados do sistema para acessar e usufruir das funcionalidades do aplicativo.			

Fonte: Elaborado pelos autores

Tabela 15: Uso de API

Identificador	Nome	Tipo	
RNF05	Uso de API	✓ Essencial	Desejável
Descrição			
Uso de API (Application Programming Interface) para fornecer o mapa do sistema, e para integrar os dados dos sensores ao sistema.			

Fonte: Elaborado pelos autores

Tabela 16: Suporte ao volume de dados

Identificador	Nome	Tipo	
RNF06	Suporte ao volume de dados	✓ Essencial	Desejável
Descrição			
O dispositivo e o aplicativo deverão ser capazes de lidar com o volume de dados sem comprometer o desempenho ou a precisão dos dados coletados.			

Fonte: Elaborado pelos autores

Tabela 17: Consumo de energia

Identificador	Nome	Tipo	
RNF07	Consumo de energia	✓ Essencial	Desejável
Descrição			
O aplicativo deve ser otimizado para minimizar o consumo de energia do dispositivo móvel e do microprocessador Esp32, prolongando a vida útil da bateria e garantindo o funcionamento contínuo do sistema de monitoramento.			

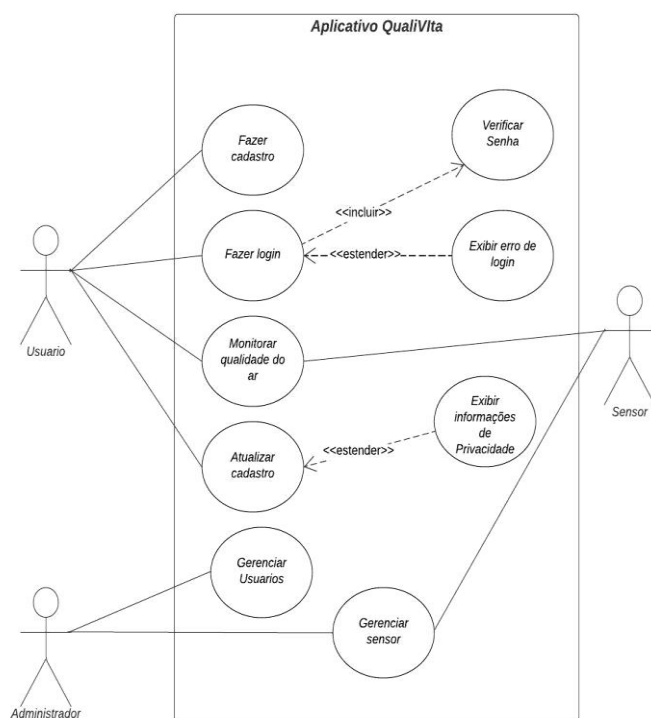
Fonte: Elaborado pelos autores

2.3 Projeto

2.3.1 Diagrama de caso de uso, classe e Cronograma

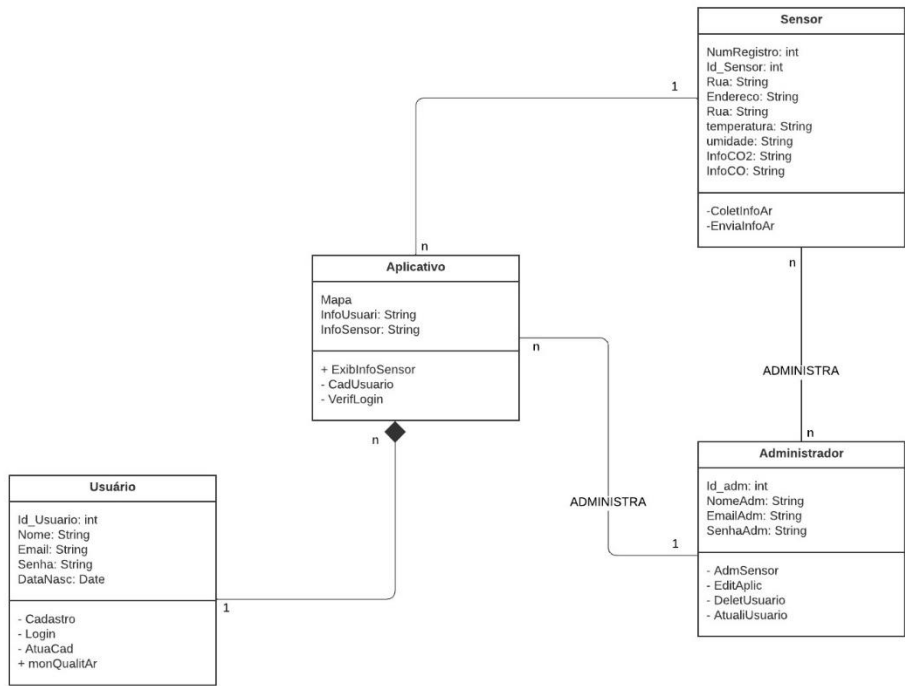
A seguir, o Diagrama de Caso de Uso, conforme a Figura 1, e o Diagrama de Classe, apresentado na Figura 2, ambos em UML, representam as principais relações entre as entidades que utilizarão o sistema, seu funcionamento e o relacionamento entre as classes:

Figura 1: Diagrama de caso e uso



Fonte: Desenvolvido pelos autores

Figura 2: Diagrama de classe



Fonte: Desenvolvido pelos autores

Tabela 18: Cronograma

RECURSOS ENVOLVIDOS			
TAREFA / ATIVIDADE	DATA INICIO	DATA FIM	RESPONSÁVEIS
INTRODUÇÃO	08/02/2024	07/03/2024	Equipe
PLANEJAMENTO DO PROJETO	07/03/2023	30/03/2023	Equipe
FUNDAMENTAÇÃO TEÓRICA	09/05/2024	30/05/2024	Equipe
OBJETIVO GERAL	07/03/2024	07/03/2024	Lucas
OBJETIVO ESPECÍFICO	07/03/2024	07/03/2024	Gustavo
LOGO DO PROJETO	23/04/2023	03/05/2024	Gustavo
LEVANTAMENTO DE REQUISITOS	22/03/2024	15/04/2024	João
PROTÓTIPO DE TELAS	15/04/2024	31/05/2024	Lucas
METODOLOGIA	27/02/2024	29/02/2024	João
MODELAGEM DE SOFTWARE	15/05/2024	7/06/2024	Equipe
DIAGRAMA DE CLASSES	01/04/2024	30/04/2024	Lucas
DIAGRAMA DE CASO DE USO	01/04/2024	30/04/2024	Lucas
CODIFICAÇÃO	15/06/2024	20/11/2024	Equipe
TESTE DO SOFTWARE	07/11/2024	25/11/2024	Equipe
SLIDES PARA APRESENTAÇÃO	25/10/2024	19/11/2024	Equipe
CONSIDERAÇÕES FINAIS	15/11/2024	20/11/2024	Lucas
REFERÊNCIAS	27/10/2024	01/11/2024	Equipe
APRESENTAÇÃO DO TCC	03/12/2024	03/12/2024	Equipe

Fonte: Desenvolvido pelos autores

2.3.2 Design do Sistema (Telas e Navegabilidade)

Figura 3: Logotipo



Fonte: Desenvolvida pelos autores

Figura 4: Telas de introdução



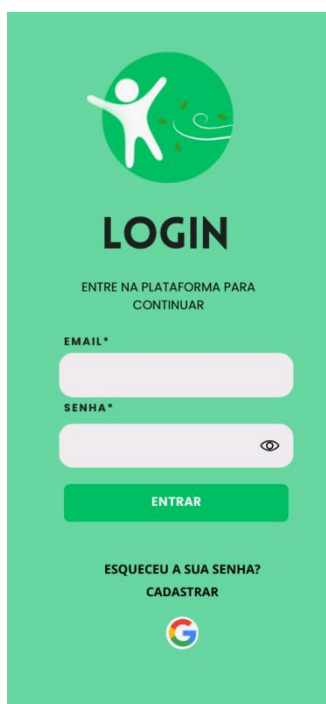


Fonte: Elaborado pelos autores

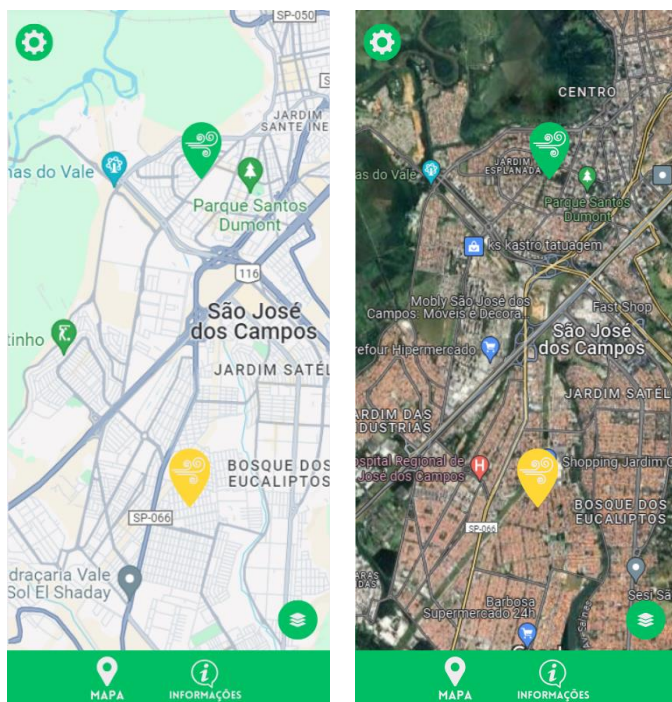
Figura 5: Tela de cadastro

A interface de criação de conta possui um fundo verde claro. No topo, há um ícone circular verde com um personagem branco saltando. Abaixo, o título "CRIE SUA CONTA" está em letras brancas maiúsculas. O formulário contém quatro campos de entrada brancos com rótulos em verde: "NOME*", "EMAIL*", "SENHA*" (acompanhada de um ícone de olho para alternar visibilidade) e "DATA DE NASCIMENTO*". Um botão verde com o texto "CADASTRAR" em branco está posicionado abaixo dos campos. Na base, há um link em azul: "Já está registrado? Faça login!".

Fonte: Elaborado pelos autores

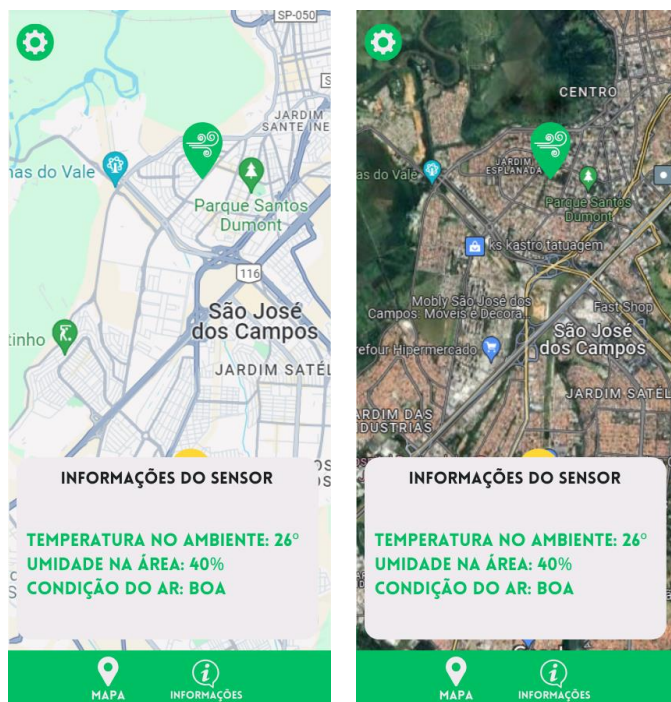
Figura 6: Tela de login

Fonte: Elaborado pelos autores

Figura 7: Tela principal com e sem imagens do satélite

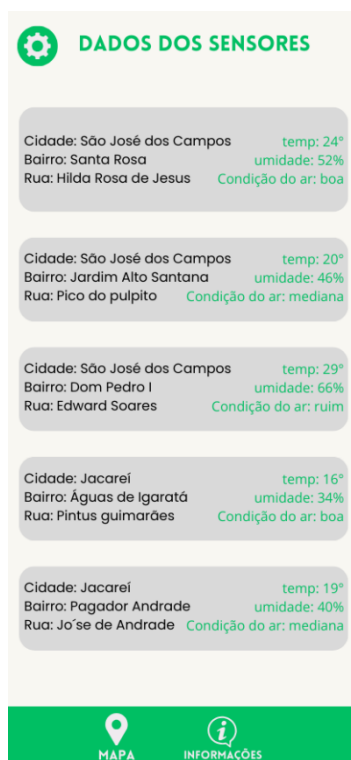
Fonte: Elaborado pelos autores

Figura 8: Tela com as informações dos sensores



Fonte: Elaborado pelos autores

Figura 9: Tela com os dados dos sensores



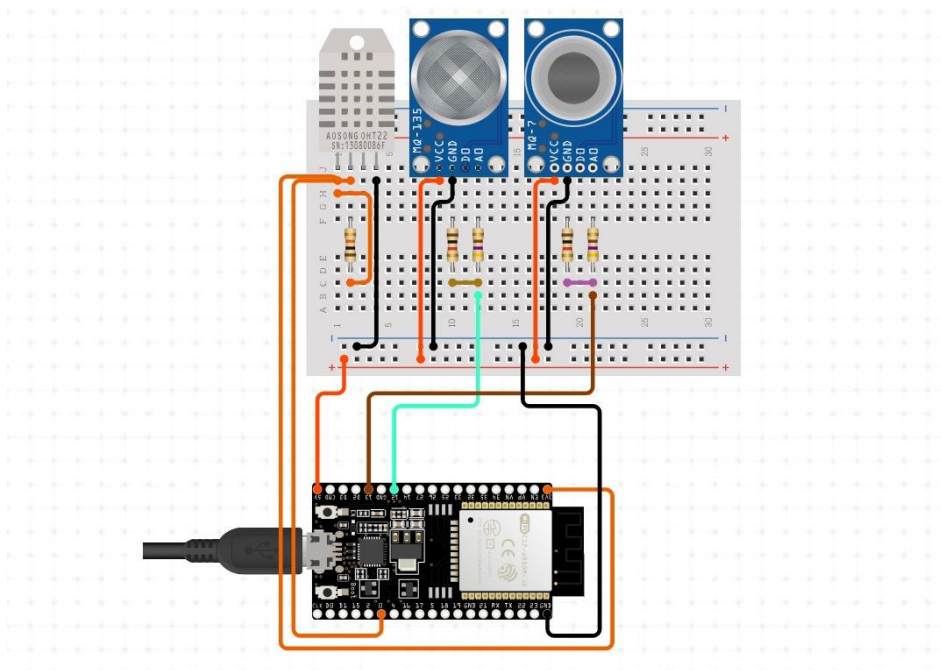
Fonte: Elaborado pelos autores

Figura 10: Tela de configurações



Fonte: Elaborado pelos autores

Figura 11: Protótipo do sensor



Fonte: Desenvolvido pelos autores

2.3.2.1 Estudo das cores, imagens, fontes e identidade visual a serem aplicadas ao sistema

Para a criação da identidade visual do projeto Qualivita, foram escolhidas formas e cores baseadas em seus significados dentro do projeto. O objetivo do aplicativo é passar pureza e uma conexão com a natureza ao usuário, por isso, as cores presentes na logo e em todo o sistema são; verde e branco, além do uso de figuras que remetem a natureza.

A Psicologia das Cores, estudo que examina como as cores afetam as emoções e comportamentos humanos, também desempenha um papel crucial na criação do design, assegurando uma experiência visual esteticamente agradável.

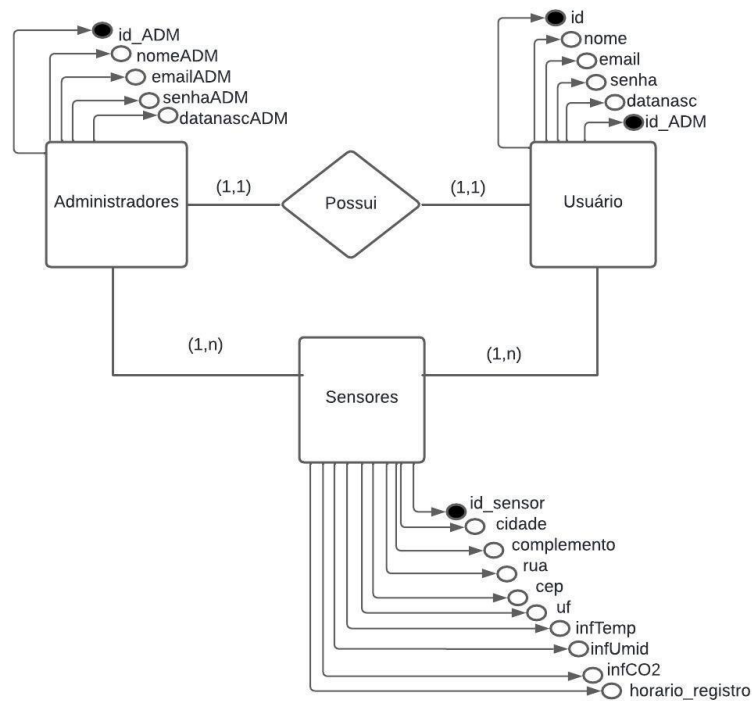
Portanto, o verde é utilizado como base, refletindo a conexão com a natureza e trazendo uma sensação de frescor e vitalidade ao design do aplicativo. Esta cor, associada à saúde e ao meio ambiente, é crucial para transmitir a ideia de um ar puro e de qualidade. O branco, por sua vez, complementa o verde, trazendo um toque de simplicidade e clareza. Ele ajuda a destacar os elementos visuais importantes e proporciona uma sensação de espaço e limpeza. A combinação dessas cores cria um ambiente visual harmonioso e agradável, essencial para um aplicativo que visa promover o bem-estar e a conscientização ambiental.

No logotipo foi usado figuras que remetem a natureza, como o vento e as folhas, também foi usado a figura de um boneco, que representa o usuário. As fontes escolhidas são simples e limpas, facilitando a leitura e a compreensão das informações apresentadas, sem causar distração ou desconforto visual.

2.4 Projeto de Banco de Dados

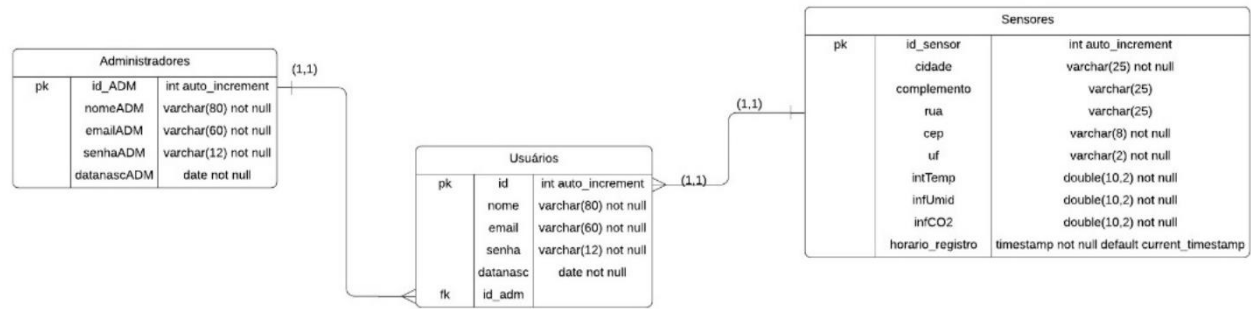
2.4.1 Modelo Conceitual e Lógico

Figura 12: Modelo Conceitual



Fonte: Desenvolvido pelos autores

Figura 13: Modelo Lógico



Fonte: Desenvolvido pelos autores

2.4.2 Modelo Físico (Não Relacional)

Figura 14: Tabela Usuários

```
1  {
2    "usuarios":{
3      "idDocumento":{
4        "dataNascimento": "2006-08-16",
5        "email": "joaovitordejesus855@gmail.com",
6        "nome": "jvpereira",
7        "userID": "CgoV0bRZYDcVQRWnofFMa232LKm1"
8      }
9    }
10 }
```

Fonte: Desenvolvido pelos autores

Figura 15: Tabela Sensores

```
1  {
2    "sensores": {
3      "idDocumento": {
4        "bairro": "Jardim Santa Rosa",
5        "cep": "12228894",
6        "cidade": "São José dos Campos",
7        "horarioRegistro": {
8          "seconds": "1699929600",
9          "nanoseconds": "0"
10       },
11       "infCO": 500,
12       "infCO2": 400,
13       "rua": "Rua Hilda Rosa de Jesus",
14       "temperatura": 27,
15       "uf": "SP",
16       "umidade": 60
17     }
18   }
19 }
```

Fonte: Desenvolvido pelos autores

Figura 16: Tabela Administradores

```

1  {
2      "administradores":{
3          "idDocumento":{
4              "adminID":"pdrYfV1lKsxyzMdzvOwXB",
5              "dataNasc":"2007-01-04",
6              "email":"lucasdonizeti@gmail.com",
7              "nome":"Lucas Donizeti dos Santos"
8          }
9      }
10 }

```

Fonte: Desenvolvido pelos autores

2.4.3 Dicionário de Dados

É essencial definir as entidades do banco de dados e suas inter-relações. O dicionário de dados fornece uma descrição detalhada de todos os objetos presentes em cada tabela, facilitando a compreensão e análise do sistema.

Tabela 19: Entidade Usuários

Entidade: Usuário			
Atributo	Domínio	Tamanho	Descrição
IdUsuario	Primary Key		Identificação de cada usuário
Nome	String	80	Nome do usuário
Email	String	60	E-mail do usuário
Datanascimento	String		Data de nascimento do usuário

Fonte: Desenvolvido pelos autores

Tabela 20: Entidade Sensores

Entidade: Sensores			
Atributo	Domínio	Tamanho	Descrição
idSensor	Primary Key		Identificação do sensor
Cidade	String	40	Cidade onde está localizado o sensor

Bairro	String	50	Bairro onde está localizado o sensor
Rua	String	35	Rua onde está localizado o sensor
CEP	String	8	CEP onde está localizado o sensor
uf	String	2	Estado onde está localizado o sensor
temperatura	Number		Informação coletada sobre a temperatura
umidade	Number		Informação coletada sobre a umidade
infCO2	Number		Informação coletada sobre o CO2
infCO	Number		Informação coletada sobre o CO
Horario_Registro	Number		Horário em que as informações foram coletadas

Fonte: Desenvolvido pelos autores

Tabela 21: Entidade Administradores

Entidade: Administradores			
Atributo	Domínio	Tamanho	Descrição
adminID	Primary Key		Identificação do administrador
Nome	String	80	Nome do administrador
Email	String	60	E-mail do administrador
DataNasc	String		Data de nascimento do administrador

Fonte: Desenvolvido pelos autores

2.5 Codificação do sistema

A seguir link para acessar o repositório do projeto no GitHub:
<https://github.com/jvpereira012/TCC>

Figura 17: Codificação do Login

```
1  useEffect(() => {
2    const unsubscribe = onAuthStateChanged(auth, (user) => {
3      if (user) {
4        console.log("Usuário já autenticado:", user);
5        navigation.navigate('TabNavigator');
6      }
7    });
8    return unsubscribe;
9  }, []);
10
11 function handleLogin () {
12   setCarregamento(true);
13   signInWithEmailAndPassword(auth, email, senha)
14     .then((userCredential) => {
15
16     const user = userCredential.user;
17     setCarregamento(false);
18     console.log("Usuário logado", user);
19     setEmail('');
20     setSenha('');
21     navigation.navigate('TabNavigator');
22   })
23   .catch((error) => {
24     setCarregamento(false);
25     Alert.alert("Erro de login", error.message);
26   });
27 }
28
```

Fonte: Desenvolvido pelos autores

Figura 18: Codificação do Cadastro

```

1  const fazercadastro = () => {
2    setCarregamento(true);
3    if (!nomeUsuario || !userEmail || !userDataNasc || !userSenha || !confirmarSenha) {
4      setCarregamento(false);
5      Alert.alert("Erro", "Por favor, preencha todos os campos!");
6      return;
7    }
8
9    if (userSenha !== confirmarSenha) {
10     Alert.alert("Erro", "As senhas não coincidem!");
11     return;
12   }
13
14   const dataFormatada = formatarDataParaBanco(userDataNasc);
15   setCarregamento(true);
16   createUserWithEmailAndPassword(auth, userEmail, userSenha)
17     .then((userCredential) => {
18       const user = userCredential.user;
19       setCarregamento(false);
20       console.log("Usuário cadastrado", user);
21
22       updateProfile(user, {
23         displayName: nomeUsuario,
24       })
25         .then(() => {
26           console.log("Perfil cadastrado com sucesso!");
27           setCarregamento(false);
28           salvarDadosAdicionais(user.uid, dataFormatada);
29         })
30         .catch((error) => {
31           setCarregamento(false);
32           console.error("Erro ao atualizar o perfil:", error);
33           Alert.alert("Erro", "Falha ao atualizar o perfil do usuário.");
34         });
35     })
36     .catch((error) => {
37       setCarregamento(false);
38       console.error("Erro ao criar usuário:", error.message);
39       Alert.alert("Erro", error.message);
40     });
41   });
42 };
43
44
45 const salvarDadosAdicionais = async (userId: string, dataNascimento: string) => {
46   try {
47     setCarregamento(true);
48     await setDoc(doc(db, "usuarios", userId), {
49       nome: nomeUsuario,
50       dataNascimento: dataNascimento,
51       email: userEmail,
52       userID: userId,
53     });
54     setCarregamento(false);
55     setEmail('');
56     setSenha('');
57     setDataNasc('');
58     setNomeUsuario('');
59     setConfirmarSenha('');
60     console.log("Dados adicionais salvos no Firestore!");
61     Alert.alert("Sucesso", "Cadastro realizado com sucesso!");
62     navigation.navigate('Login');
63   } catch (error) {
64     setCarregamento(false);
65     console.error("Erro ao salvar dados adicionais no Firestore:", error);
66     Alert.alert("Erro", "Falha ao salvar dados adicionais.");
67   }
68 };

```

Fonte: Desenvolvido pelos autores

Figura 19: Codificação da Redefinição de Senha

```

1  function redefinirsenha() {
2      setCarregamento(true);
3      if (email !== '') {
4          setCarregamento(true);
5          sendPasswordResetEmail(auth, email)
6              .then(() => {
7              Alert.alert("Mensagem", `Foi enviada uma mensagem para ${email}, verifique seu email!`);
8              setCarregamento(false);
9              navigation.navigate("Login");
10             setEmail('');
11             })
12             .catch((error) => {
13                 setCarregamento(false);
14                 Alert.alert("OPS!", error.message + " Tente novamente!");
15             });
16         } else {
17             setCarregamento(false);
18             Alert.alert("Mensagem", "É necessário informar o email para redefinir sua senha.");
19         }
20     }

```

Fonte: Desenvolvido pelos autores

Figura 20: Codificação do Mapa 1

```

1  async function requestLocationPermissions() {
2      const { granted } = await requestForegroundPermissionsAsync();
3      if (granted) {
4          const currentPosition = await getCurrentPositionAsync();
5          setLocalizacao(currentPosition);
6      }
7  }
8
9  useEffect(() => {
10     requestLocationPermissions();
11 }, []);
12
13  useEffect(() => {
14     watchPositionAsync(
15         {
16             accuracy: LocationAccuracy.Highest,
17             timeInterval: 1300,
18             distanceInterval: 1,
19         },
20         (resposta) => {
21             setLocalizacao(resposta);
22         }
23     );
24 }, []);
25
26  const getInf = async () => {
27     try {
28         const q = query(
29             collection(db, 'sensores'),
30             orderBy('horarioRegistro', 'desc'),
31             limit(1)
32         );
33
34         const querySnapshot = await getDocs(q);

```

Fonte: Desenvolvido pelos autores

Figura 21: Codificação do Mapa 2

```

35
36     if (!querySnapshot.empty) {
37         const doc = querySnapshot.docs[0];
38         const data = doc.data();
39         console.log(data);
40         setTemp(data.temperatura);
41         setUmidade(data.umidade);
42         let co2 = data.infcO2;
43         if (co2 >= 400 && co2 <= 750) {
44             setCorTexto('#00bf63');
45             setEstado('Boa');
46         }
47         else if (co2 > 750 && co2 <= 1200) {
48             setCorTexto('#e8b501');
49             setEstado('Média');
50         }
51         else if (co2 > 1200) {
52             setCorTexto('#e61913');
53             setEstado('Ruim')
54         }
55     } else {
56         console.log('Nenhum documento encontrado.');

```

Fonte: Desenvolvido pelos autores

Figura 22: Codificação das Informações

```

1  const getInf = async () => {
2    setcarregamento(true);
3    try {
4      const q = query(
5        collection(db, 'sensores'),
6        orderBy('horarioRegistro', 'desc'),
7        limit(1)
8      );
9
10     const querySnapshot = await getDocs(q);
11
12     if (!querySnapshot.empty) {
13       const doc = querySnapshot.docs[0];
14       const data = doc.data();
15       console.log(data);
16       setCidade(data.cidade);
17       setRua(data.rua);
18       setBairro(data.bairro);
19       setTemp(data.temperatura);
20       setUmidade(data.umidade);
21
22       let co2 = data.infCO2;
23       if (co2 >= 400 && co2 <= 750) {
24         setCor('#4CAF50');
25         setCorTexto('#4CAF50');
26         setEstado('Boa');
27       }
28       else if (co2 > 750 && co2 <= 1200) {
29         setCor('#e8b501');
30         setCorTexto('#e8b501');
31         setEstado('Média');
32       }
33       else if (co2 > 1200) {
34         setCor('#e61913');
35         setCorTexto('#e61913');
36         setEstado('Ruim')
37       }
38
39     } else {
40       console.log('Nenhum documento encontrado.');

```

Fonte: Desenvolvido pelos autores

Figura 23: Codificação do Histórico

```

1  const getInf = async () => {
2    setCarregamento(true);
3    try {
4      const q = query(
5        collection(db, 'sensores'),
6        orderBy('horarioRegistro', 'desc'),
7      );
8
9      const querySnapshot = await getDocs(q);
10     const docs: {
11       idSensor: string;
12       bairro: string;
13       rua: string;
14       dia: string;
15       temperatura: string | number;
16       umidade: string | number;
17       infCO: string | number;
18       infCO2: string | number;
19     }[] = [];
20
21     querySnapshot.forEach(doc => {
22       const data = doc.data();
23       const timestamp = data.horarioRegistro?.seconds;
24       const formattedDate = timestamp
25         ? new Date(timestamp * 1000).toLocaleDateString('pt-BR')
26         : "Data não disponível";
27
28       docs.push({
29         idSensor: doc.id,
30         bairro: data.bairro || "N/A",
31         rua: data.rua || "N/A",
32         dia: formattedDate,
33         temperatura: data.temperatura ?? "N/A",
34         umidade: data.umidade ?? "N/A",
35         infCO: data.infCO ?? "N/A",
36         infCO2: data.infCO2 ?? "N/A",
37       });
38     });
39
40     setRegistros(docs);
41   } catch (error) {
42     console.error("Erro ao buscar dados:", error);
43   } finally {
44     setCarregamento(false);
45   }
46 };
47
48 useEffect(() => {
49   getInf();
50 }, []);

```

Fonte: Desenvolvido pelos autores

Figura 24: Codificação das Rotas

```

1  export function TabNavigator() {
2    return (
3      <Tab.Navigator screenOptions={{
4        tabBarStyle: {
5          backgroundColor: '#00bf63',
6          borderTopColor: 'transparent',
7          paddingBottom: 2,
8          paddingTop: 5,
9          height: 50
10       },
11       tabBarActiveTintColor: '#efebeF',
12     }}>
13
14     <Tab.Screen
15       name="MAPA"
16       component={Home}
17       options={{
18         headerShown: false,
19         tabBarIcon: ({ color, size }) => (
20           <Feather name="map-pin" size={size} color={'#efebeF'} />
21         ),
22         tabBarLabelStyle: { fontFamily: 'Lovelo', fontSize: 10, color: '#efebeF' },
23       }}
24     />
25     <Tab.Screen
26       name="INFORMAÇÕES"
27       component={Informacoes}
28       options={{
29         headerShown: false,
30         tabBarIcon: ({ color, size }) => (
31           <MaterialCommunityIcons name="information-variant" size={size} color={'#efebeF'} />
32         ),
33         tabBarLabelStyle: { fontFamily: 'Lovelo', fontSize: 10, color: '#efebeF' },
34       }}
35     />
36     <Tab.Screen
37       name = "CONFIGURAÇÕES"
38       component={Configuracoes}
39       options={{
40         headerShown: false,
41         tabBarIcon: ({ color, size }) => (
42           <MaterialCommunityIcons name="cog" size={size} color={'#efebeF'} />
43         ),
44         tabBarLabelStyle: { fontFamily: 'Lovelo', fontSize: 10, color: '#efebeF' },
45       }}
46     />
47   </Tab.Navigator>
48 );
49 }

```

Fonte: Desenvolvido pelos autores

Figura 25: Codificação das Informações do Usuário

```

1  useEffect(() => {
2    const unsubscribe = onAuthStateChanged(auth, async (currentUser) => {
3      if (currentUser) {
4        try {
5          const docRef = doc(db, 'usuarios', currentUser.uid);
6          const docSnap = await getDoc(docRef);
7          if (docSnap.exists()) {
8            setUserProfile(docSnap.data() as UserProfile);
9          } else {
10             Alert.alert('Erro', 'Usuário não encontrado no banco de dados.');

```

Fonte: Desenvolvido pelos autores

Figura 26: Codificação do Sensor 1

```

60  // Leituras do DHT11
61  int h = dht.readHumidity();
62  int t = dht.readTemperature();
63
64  // Leituras dos sensores MQ
65  int sensorValueMQ7 = analogRead(MQ7_PIN);
66  float ppmCO = calculatePPMCO(sensorValueMQ7);
67
68  int sensorValueMQ135 = analogRead(MQ135_PIN);
69  float ppmCO2 = calculatePPMCO2(sensorValueMQ135);
70
71  // Verificação das leituras
72  if (isnan(h) || isnan(t)) {
73    Serial.println("Falha na leitura do sensor DHT!");
74    return;
75  }
76
77  if (ppmCO < 0 || ppmCO > 1000) ppmCO = 0; // Validar faixa de CO
78  if (ppmCO2 < 0 || ppmCO2 > 5000) ppmCO2 = 0; // Validar faixa de CO2
79

```

Fonte: Desenvolvido pelos autores

Figura 27: Codificação do Sensor 2

```

91 // Formatar dados JSON
92 String jsonData = "{";
93 jsonData += "\"fields\": {";
94 jsonData += "\"cidade\": {\"stringValue\": \"" + cidade + "\"},";
95 jsonData += "\"bairro\": {\"stringValue\": \"" + bairro + "\"},";
96 jsonData += "\"rua\": {\"stringValue\": \"" + rua + "\"},";
97 jsonData += "\"cep\": {\"stringValue\": \"" + cep + "\"},";
98 jsonData += "\"uf\": {\"stringValue\": \"" + uf + "\"},";
99 jsonData += "\"temperatura\": {\"integerValue\": \"" + String(t) + "\"},";
100 jsonData += "\"umidade\": {\"integerValue\": \"" + String(h) + "\"},";
101 jsonData += "\"infCO2\": {\"doubleValue\": \"" + String(ppmCO2) + "\"},";
102 jsonData += "\"infCO\": {\"doubleValue\": \"" + String(ppmCO) + "\"},";
103 jsonData += "\"horarioRegistro\": {\"timestampValue\": \"" + timestampFirestore + "\"},";
104 jsonData += "}}";
105
106 // Envio para o Firebase
107 String path = "https://firestore.googleapis.com/v1/projects/" + projectID + "/databases/(default)/documents/" + collectionID + "?key=" + firebaseAuth;

```

Fonte: Desenvolvido pelos autores

3 RESULTADOS

A tela Splash conforme é ilustrado na figura 28, é exibida no lançamento do aplicativo, enquanto o carregamento está em andamento:

Figura 28: Tela de Splash



Fonte: Desenvolvido pelos autores

A seguir, apresentam-se as telas de introdução nas figuras 29, 30 e 31, com uma breve explicação dos conceitos e funcionalidades do aplicativo:

Figura 29: Tela de Introdução 1



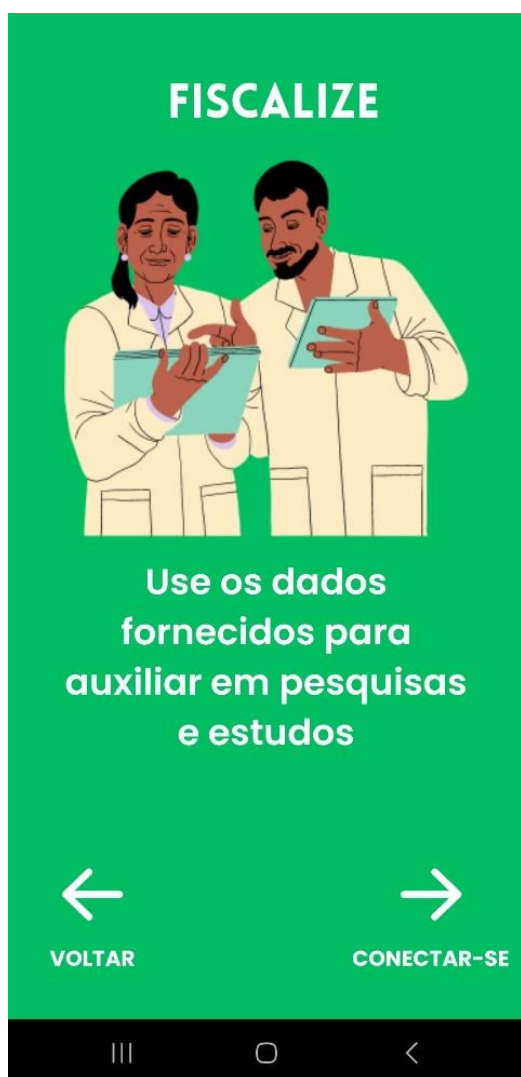
Fonte: Desenvolvido pelos autores

Figura 30: Tela de Introdução 2



Fonte: Desenvolvido pelos autores

Figura 31: Tela de Introdução 3



Fonte: Desenvolvido pelos autores

A tela de login, conforme a figura 32, permite que o usuário se conecte ao aplicativo após verificar seus dados. Caso o cadastro ainda não tenha sido realizado, será necessário efetuá-lo na tela de cadastro, apresentada na figura 33. Se o usuário esquecer sua senha, poderá recuperá-la por meio da tela de recuperação de senha, conforme ilustrado na figura 34:

Figura 32: Tela de Login

A imagem mostra a interface de login de um aplicativo em um dispositivo móvel. O fundo é verde. No topo, há um ícone circular verde com um personagem branco saltando e folhas amarelas. Abaixo do ícone, o texto "LOGIN" aparece em letras grandes e negritadas. Segue-se a frase "ENTRE NA PLATAFORMA PARA CONTINUAR" em letras menores. Há dois campos de entrada: "EMAIL*" com o placeholder "Insira seu email" e "SENHA*" com o placeholder "Insira sua senha" e um ícone de olho para alternar a visibilidade. Abaixo dos campos está um botão verde com o texto "ENTRAR" em branco. Na base da tela, há o texto "Esqueceu a sua senha?" e "Não tem uma conta? CADASTRE-SE". O rodapé do dispositivo mostra os ícones de menu, home e voltar.

Fonte: Desenvolvido pelos autores

A figura 33 exibe a tela de cadastro, na qual os dados do usuário são inseridos no banco de dados, permitindo a realização do login:

Figura 33: Tela de Cadastro

Left Screen:

Nome de usuário*

Insira seu nome

Email*

Insira seu email

Crie uma senha*

Crie uma senha

Confirme sua senha*

Confirme sua senha

Data de nascimento*

Insira sua data de nascimento (Dia/Mês/Ano)

Right Screen:

Nome de usuário*

Insira seu nome

Email*

Insira seu email

Crie uma senha*

Crie uma senha

Confirme sua senha*

Confirme sua senha

Data de nascimento*

Insira sua data de nascimento (Dia/Mês/Ano)

CADASTRAR

Já está registrado? Faça Login

Fonte: Desenvolvido pelos autores

A tela de recuperação de senha permite alterar os dados cadastrados. Ao inserir o e-mail, uma mensagem será enviada para a alteração da senha, possibilitando o login novamente:

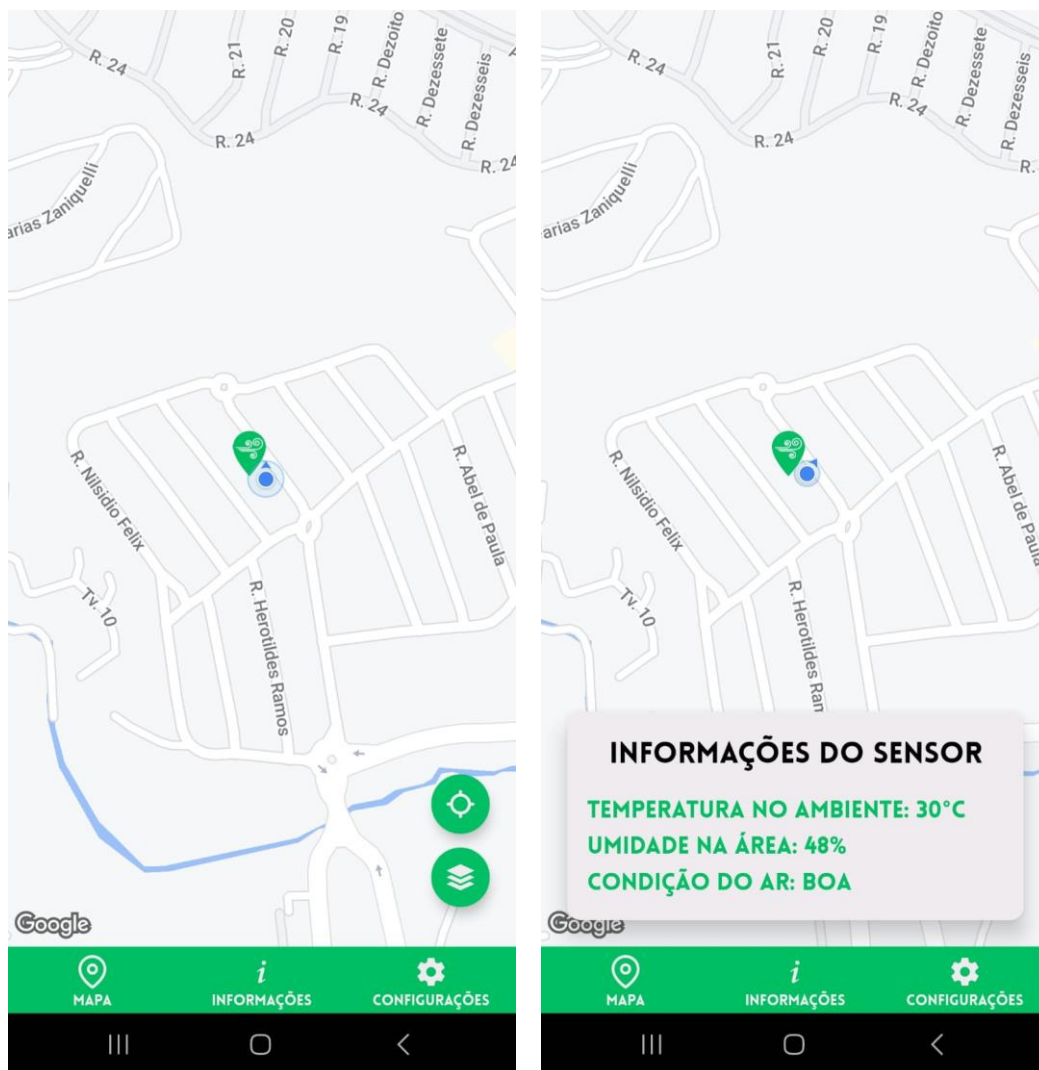
Figura 34: Tela de Recuperação de Senha

A imagem mostra a interface de uma tela de recuperação de senha em um aplicativo móvel. O fundo é verde. No topo, há um ícone circular verde com um personagem branco saltando e folhas douradas. Abaixo do ícone, o título "RECUPERAÇÃO DE SENHA" está em letras maiúsculas e negrito. Segue o texto "INSIRA SEU EMAIL PARA REDEFINIR SUA SENHA" em letras maiúsculas. Abaixo disso, o rótulo "EMAIL*" precede um campo de entrada branco com o placeholder "Insira seu email". Um botão verde com o texto "ENVIAR" em branco está abaixo do campo. Na base da tela, há um link "Vá para a tela de login". O rodapé do aplicativo, com ícones de menu, home e voltar, é visível na parte inferior.

Fonte: Desenvolvido pelos autores

A tela do mapa funcional, apresentada na figura 35, exibe a localização exata do sensor. Ao pressionar a localização do sensor, é possível visualizar as informações sobre umidade, temperatura e qualidade do ar:

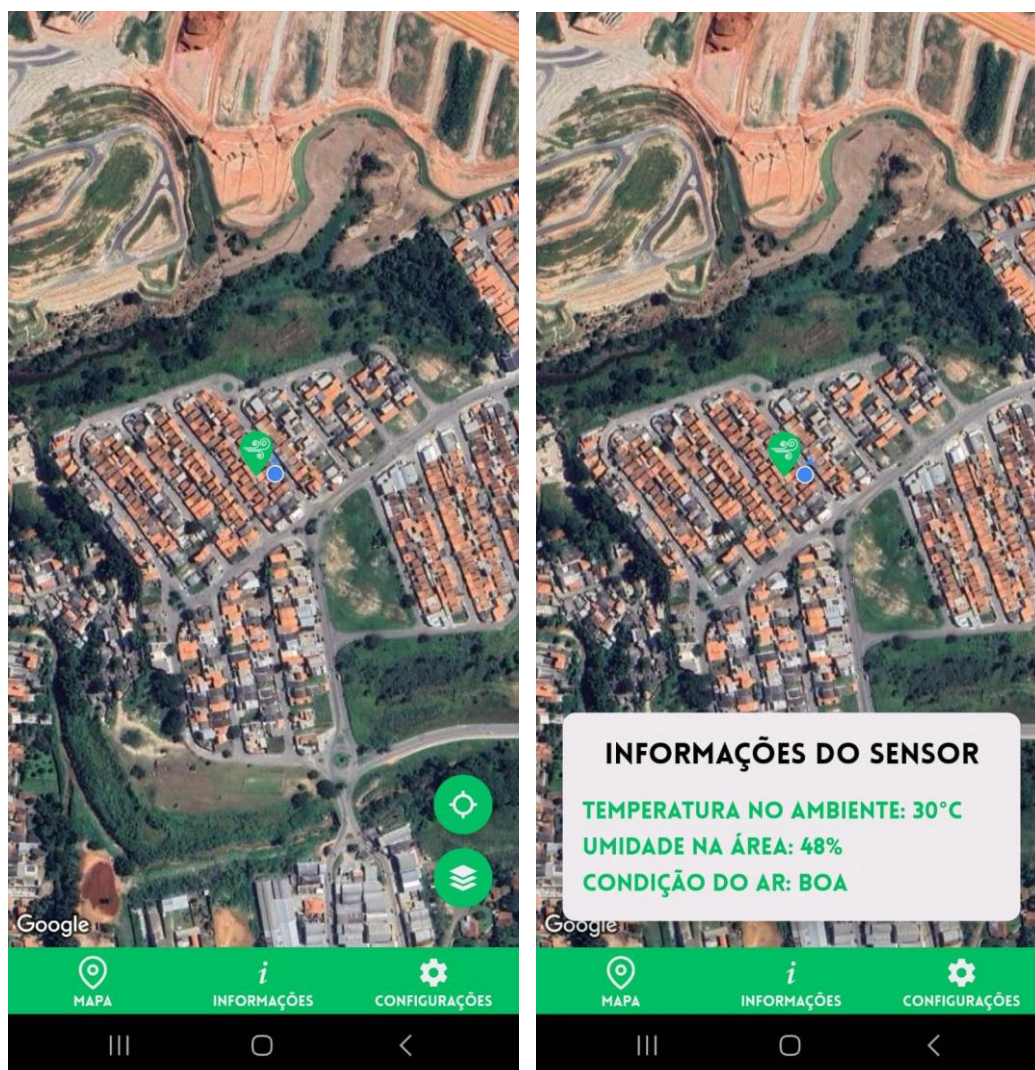
Figura 35: Tela do Mapa



Fonte: Desenvolvido pelos autores

Tela do mapa no tipo satélite exibida na figura 36:

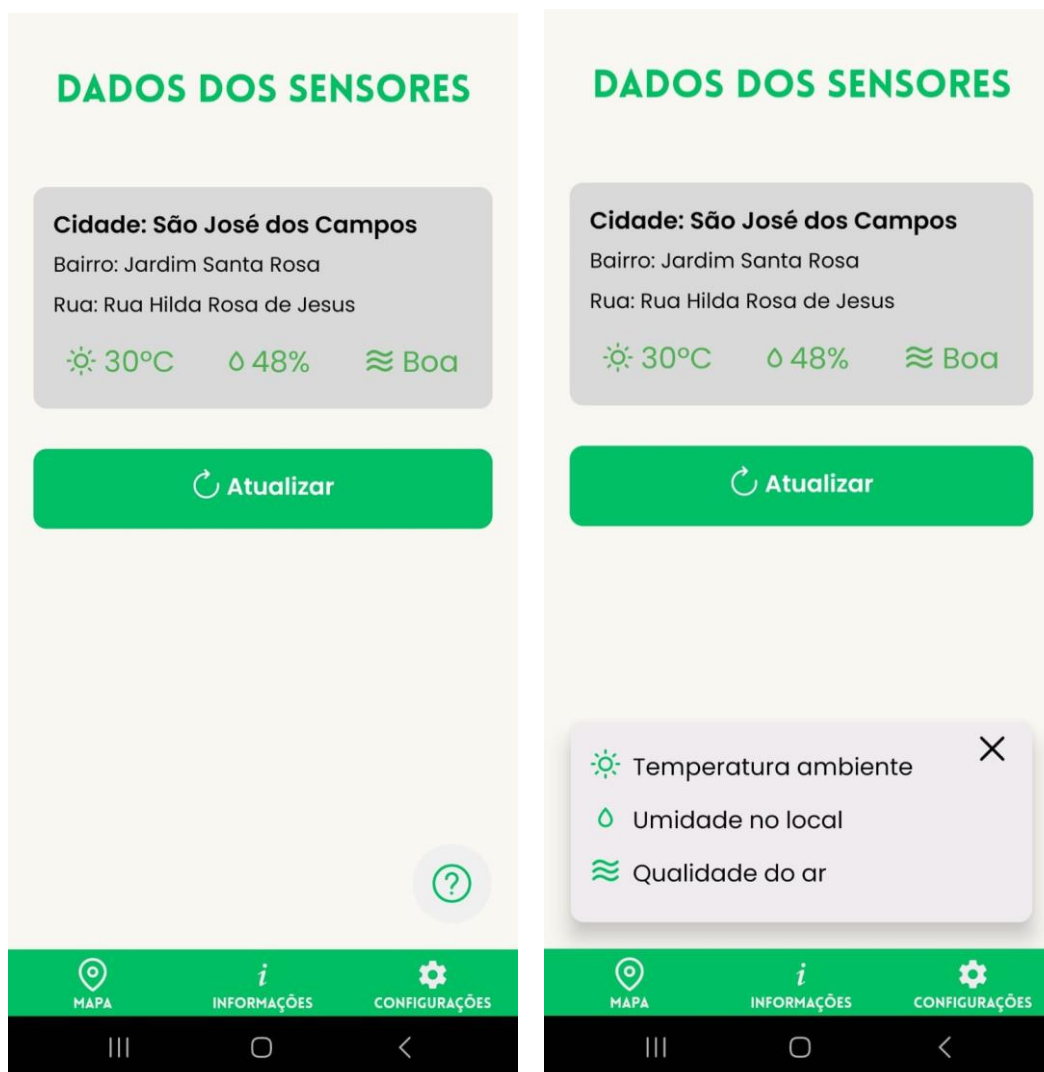
Figura 36: Tela do Mapa Satélite



Fonte: Desenvolvido pelos autores

A tela de informações, conforme a figura 37, tem a função de exibir os dados sobre temperatura, umidade e qualidade do ar em formato de lista:

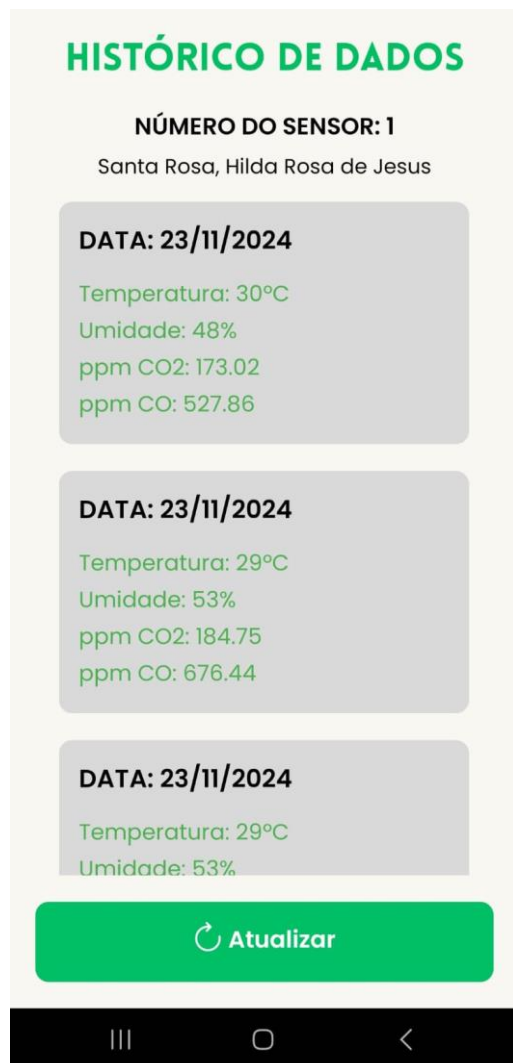
Figura 37: Tela de Informações



Fonte: Desenvolvido pelos autores

A figura 38 apresenta a tela que exibe o histórico dos dados coletados de cada sensor específico:

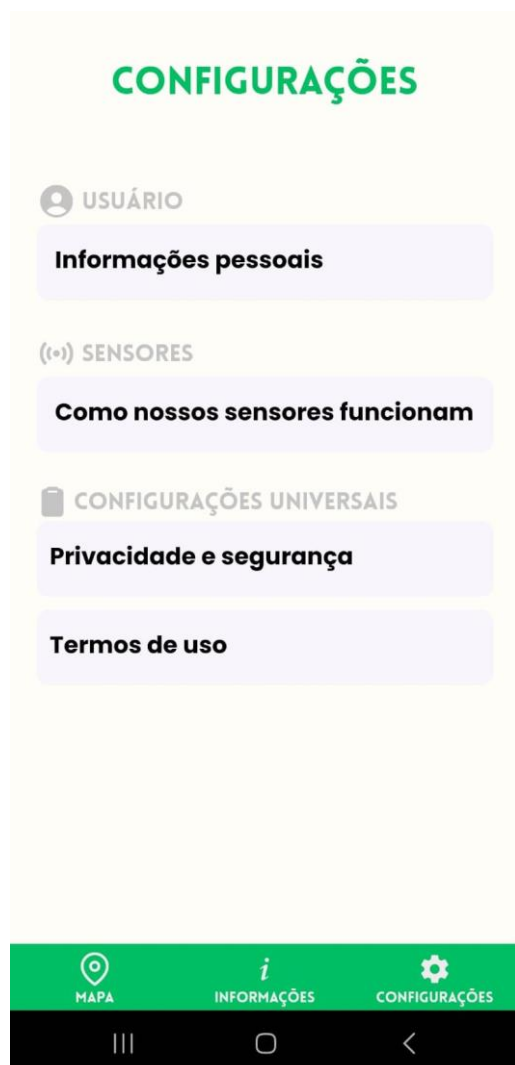
Figura 38: Tela de Histórico



Fonte: Desenvolvido pelos autores

A tela de configurações, conforme a figura 39 demonstra, possui várias opções, desde visualizar as informações anteriormente cadastradas, até como funcionam nossos sensores:

Figura 39: Tela de Configurações



Fonte: Desenvolvido pelos autores

Tela que apresenta os dados previamente cadastrados pelo usuário e oferece a opção de logout, como indicado na figura 40:

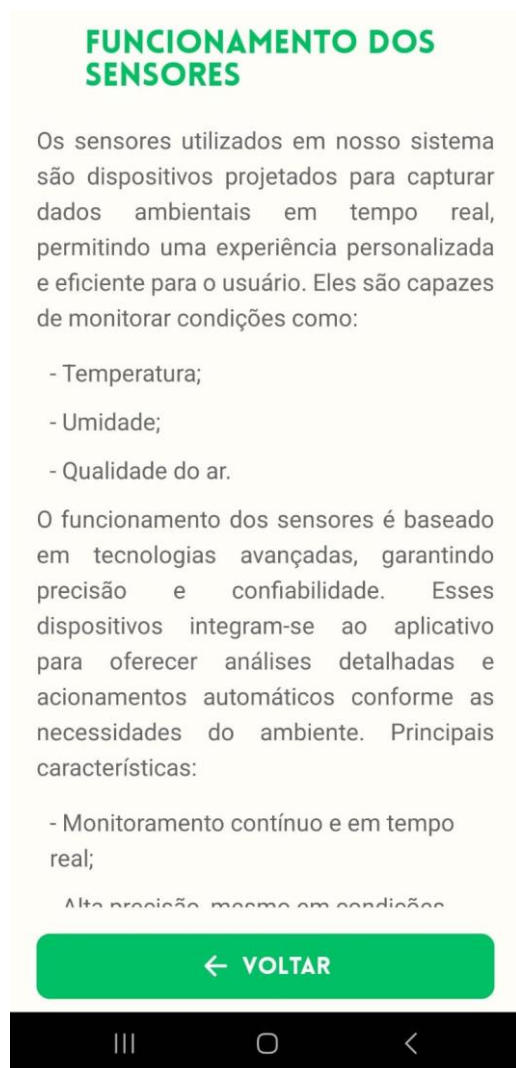
Figura 40: Tela de Informações do Usuário



Fonte: Desenvolvido pelos autores

Tela que explica como os sensores funcionam, ilustrada na figura 41:

Figura 41: Tela de Como Nossos Sensores Funcionam



Fonte: Desenvolvido pelos autores

Telas ilustradas na Figura 42, que apresentam os termos de uso do aplicativo, além de informações sobre privacidade e segurança do usuário:

Figura 42: Telas de Privacidade e Termos de Uso



Fonte: Desenvolvido pelos autores

4 CONCLUSÃO

O trabalho de conclusão de curso teve como foco desenvolver um aplicativo mobile e um sensor voltados para monitorar a qualidade do ar de forma simples e informativa. Esse projeto foi idealizado para melhora da qualidade de vida dos usuários, oferecendo dados sobre o ar em suas localidades.

Após a conclusão do sistema, o Qualivita passou a ser utilizado por usuários, permitindo à equipe confirmar que o aplicativo atende aos requisitos estabelecidos. Ele facilita a visualização de informações essenciais sobre a qualidade do ar e contribui para a conscientização e estudos sobre a poluição em diferentes regiões.

O Qualivita tem o potencial de impactar positivamente a saúde e a qualidade de vida de seus usuários, ao oferecer uma visão clara da situação da poluição no ambiente em que vivem, promovendo um estilo de vida mais saudável e informado.

5 REFERÊNCIAS

Gov.br. **Poluição é risco para saúde: saiba as principais doenças relacionadas ao ambiente poluído.** 2024. Disponível em: <<https://www.gov.br/ebserh/pt-br/comunicacao/noticias/poluicao-e-risco-para-saude-saiba-as-principais-doencas-relacionadas-ao-ambiente-poluido>>. Acesso em: 12 nov. 2024.

INSTITUTO DE ENERGIA E MEIO AMBIENTE (IEMA). **Cidade de São Paulo tem poluição do ar acima do recomendado pela OMS nos últimos 22 anos.** 2022. Disponível em: <<https://energiaeambiente.org.br/cidade-de-sao-paulo-tem-poluicao-do-ar-acima-do-recomendado-pela-oms-nos-ultimos-22-anos-20220526>>. Acesso em: 12 nov. 2024.

ecycle. **Sensor caseiro e simples de fazer pode ajudar no combate à poluição do ar.** 2020. Disponível em: <<https://www.ecycle.com.br/sensor-caseiro-de-poluicao-do-ar-diy/>>. Acesso em: 12 nov. 2024.

AGÊNCIA AMBIENTAL DO ESTADO DE SÃO PAULO (CETESB). **CETESB lança nova versão de aplicativo sobre qualidade do ar.** 13 mar. 2020. Disponível em: <<https://cetesb.sp.gov.br/blog/2020/03/13/cetesb-lanca-nova-versao-de-aplicativo-sobre-qualidade-do-ar/>>. Acesso em: 13 mar. 2024.

PROGRAMA DAS NAÇÕES UNIDAS PARA O MEIO AMBIENTE (PNUMA). **Maior plataforma de dados sobre qualidade do ar é lançada pelo PNUMA.** Disponível em: <<https://www.unep.org/pt-br/noticias-e-reportagens/comunicado-de-imprensa/maior-plataforma-de-dados-sobre-qualidade-do-ar-e>>. Acesso em: 13 mar. 2024.

ALURA. **React Native: tudo que você precisa saber.** Disponível em: <<https://www.alura.com.br/artigos/react-native>>. Acesso em: 20 mar. 2024.

MOZILLA DEVELOPER NETWORK. **JavaScript.** Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 20 mar. 2024.

ALURA. **Node.js: tudo que você precisa saber.** Disponível em: <<https://www.alura.com.br/artigos/node-js>>. Acesso em: 21 mar. 2024.

ESCOLA DNC. **Introdução ao MySQL Workbench: uma ferramenta poderosa para desenvolvedores e administradores de banco de dados.** Disponível em: <<https://www.escoladnc.com.br/blog/introducao-ao-mysql-workbench-uma-ferramenta-poderosa-para-desenvolvedores-e-administradores-de-banco-de-dados/>>. Acesso em: 23 mar. 2024.

VICTOR VISION. **O que é Arduino?** Disponível em: <<https://victorvision.com.br/blog/o-que-e-arduino/>>. Acesso em: 23 mar. 2024.

MINATTI, Douglas. **Conhecendo o ESP32: Guia Prático para Projetos IoT.** Disponível em: <<https://www.filipeflop.com/blog/esp32-o-guia-definitivo/>>. Acesso em: 23 mar. 2024.

TECH-MAKER. **Como Configurar e Utilizar o ESP32 em Projetos IoT.** Disponível em: <<https://www.techmaker.com.br/blog/o-que-e-esp32/>>. Acesso em: 27 mar. 2024.

MASTERWALKER. **Como Usar com Arduino - Sensor de Umidade e Temperatura DHT11.** Disponível em: <<https://blogmasterwalkershop.com.br>>. Acesso em: 30 mar. 2024.

PROCEDIMENTO.COM.BR. **Como Utilizar o Sensor DHT11 com Arduino para Medir Temperatura e Umidade.** Disponível em: <<https://www.procedimento.com.br>>. Acesso em: 30 mar. 2024.

ARDUINO PROJECTS. **Como Usar o Sensor de Gás MQ-7 com Arduino.** Disponível em: <<https://www.arduino-projects.com>>. Acesso em: 5 abr. 2024.

ELECTROFUN. **Monitorando Monóxido de Carbono com o Sensor MQ-7.** Disponível em: <<https://electrofun.com.br>>. Acesso em: 5 abr. 2024.

TREINAWEB. **VS Code: o que é e por que você deve usar.** Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em: 7 mai. 2024.

PHP. **O que é PHP.** Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 7 mai. 2024.

PHP. **Strings em PHP.** Disponível em: <<https://www.php.net/manual/en/language.types.string.php>>. Acesso em: 7 mai. 2024.

DIEGO WILLIAM. **O que é o pacote Office e qual a sua importância no mercado de trabalho.** Disponível em: <<https://diegowilliam.com.br/artigo/o-que-e-o-pacote-office-e-qual-a-sua-importancia-no-mercado-de-trabalho>>. Acesso em: 7 mai. 2024.

EDUCAMUNDO. **O que é pacote Office e quais seus programas mais utilizados.** Disponível em: <<https://www.educamundo.com.br/blog/o-que-e-pacote-office-e-quais-seus-programas-mais-utilizados-2/>>. Acesso em: 7 mai. 2024.

GITHUB. **phpMyAdmin.** Disponível em: <<https://github.com/phpmyadmin/phpmyadmin.git>>. Acesso em: jun nov. 2024.

REACT NATIVE MAPS. **Documentação oficial da biblioteca react-native-maps.** Disponível em: <<https://github.com/react-native-maps/react-native-maps>>. Acesso em: 07 jul. 2024.

GOOGLE. **Introdução ao Firebase.** Disponível em: <<https://firebase.google.com/docs>>. Acesso em: 20 nov. 2024.

FIREBASE. **Autenticação no Firebase.** Disponível em: <<https://firebase.google.com/docs/auth?hl=pt-br>>. Acesso em: 20 nov. 2024.

FIREBASE. **Banco de dados em tempo real no Firebase.** Disponível em: <<https://firebase.google.com/docs/database?hl=pt-br>>. Acesso em: 19 nov. 2024.

FIREBASE. **Modelo de dados no Firestore.** Disponível em: <<https://cloud.google.com/firestore/docs/data-model?hl=pt-br>>. Acesso em: 29 nov. 2024.

FERRARI, A. R. **Requisitos dos Clientes e a Qualidade no Monitoramento Ambiental - CENED Cursos.** Disponível em: <<https://cenedcursos.com.br/meio-ambiente/requisitos-dos-clientes-qualidade/>>. Acesso em: 20 nov. 2024.

SISTEMA DE MONITORAMENTO DE QUALIDADE DO AR UTILIZANDO SENSORES DE BAIXO CUSTO. Disponível em: <https://www.researchgate.net/publication/320010558_Sistema_de_monitoramento_de_qualidade_do_ar_com_sensores_de_baixo_custo>. Acesso em: 17 out. 2024.

PLANO DE TESTES PARA SISTEMAS DE MONITORAMENTO AMBIENTAL. Disponível em: <<https://www.cin.ufpe.br/~gamr/FAFICA/Desenvolvimento%20de%20sistemas/Exemplo%20de%20Plano%20de%20Testes.doc>>. Acesso em: 22 nov. 2024

APÊNDICE – Plano de Teste

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC ILZA NASCIMENTO PINTUS

QUALIVITA
Plano de Teste

SÃO JOSÉ DOS CAMPOS
2024

CEETEPS
CENTRO ESTADUAL EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC ILZA NASCIMENTO PINTUS

LUCAS DONIZETI DOS SANTOS
JOÃO VITOR PEREIRA DE JESUS
GUSTAVO RAMOS DA SILVA SANTOS

QUALIVITA

SÃO JOSÉ DOS CAMPOS

2024

Sumário

.....	
1 INTRODUÇÃO.....	77
2 SOBRE O PROJETO.....	77
3 OBJETIVOS.....	77
3.1 Sobre o sistema.....	78
3.2 Escopo.....	78
4 REQUISITOS A SEREM TESTADOS – REQUISITOS DE QUALIDADE.....	79
4.1 Teste de Banco de Dados.....	79
4.2 Teste Funcional.....	79
4.3 Teste da Interface de Usuário.....	80
4.4 Teste de Desempenho.....	80
4.5 Teste de Estresse.....	81
4.6 Teste de Segurança.....	81
4.7 Teste de Instalação e Compatibilidade.....	81
4.8 Teste de Usabilidade.....	82
5 REQUISITOS DO CLIENTE.....	82
6 REQUISITOS DO SOFTWARE.....	83
6.1 Requisitos Funcionais.....	83
6.2 Requisitos Não Funcionais.....	86
7 ESTRATÉGIAS E FERRAMENTAS DE TESTE.....	89
7.1 Teste do Banco de Dados.....	89
7.2 Teste Funcional.....	89
7.3 Teste da Interface do Usuário.....	90
7.4 Teste de Performance.....	90
7.5 Teste de Carga.....	90

7.6 Teste de Segurança e Controle de Acesso.....	91
7.7 Teste de Instalação.....	91
8 EQUIPE E INFRAESTRUTUTRA.....	91
9 CRONOGRAMA.....	92
10 CONCLUSÃO.....	93

1 INTRODUÇÃO

Os testes de software serão aplicados no projeto Qualivita para garantir o correto funcionamento de todas as funcionalidades e maior controle do desenvolvimento do sistema. O objetivo é assegurar a confiança em relação às informações sobre a qualidade do ar disponibilizadas aos usuários e o desempenho eficiente do aplicativo.

2 SOBRE O PROJETO

O projeto Qualivita foi desenvolvido para monitorar a qualidade do ar em diferentes pontos da cidade, disponibilizando informações importantes sobre os níveis de poluição. Ele foca em conscientizar os usuários sobre os impactos da poluição em sua qualidade de vida e ajudá-los a tomar decisões mais precisas sobre locais com melhores condições ambientais.

O aplicativo fornece dados em tempo real coletados por sensores integrados ao ESP32, que são exibidos em mapas interativos e relatórios visuais. As informações coletadas são armazenadas em um banco de dados conectado ao sistema, permitindo que os usuários visualizem a qualidade do ar em diferentes áreas de forma rápida e informativa.

A aplicativo é compatível com dispositivos Android, e foi desenvolvido utilizando tecnologias como React Native, C++ e o firestore do firebase para envio de dados.

3 OBJETIVOS

Os objetivos dos testes se concentram na identificação e correção de possíveis falhas no sistema, garantindo uma experiência de uso estável e segura para os usuários. Além disso, os testes servem para validar que o aplicativo atende a todos os requisitos funcionais e não funcionais especificados no projeto.

Cada teste avaliará uma funcionalidade específica do aplicativo, como mostrar dados em tempo real, conectar com o banco de dados e facilitar o uso da interface. Após cada teste, serão feitos relatórios detalhados para facilitar a análise e a implementação de melhorias no sistema.

3.1 Sobre o sistema

O Qualivita é um aplicativo desenvolvido em React Native, feito para ser simples e fácil de usar. A integração entre a base de dados e a interface é feita por meio de funções construídas a partir dos comandos do Firebase SDK.

O armazenamento de informações é feito em um servidor Firebase, tendo como banco de dados principal o Firestore. Os dados são gerenciados pelos autores do projeto através da interface do Firebase, garantindo o envio e armazenamento correto das informações.

3.2 Escopo

O projeto Qualivita passará pelos testes unitários, de integração e de sistema. Esses testes têm o objetivo de validar a funcionalidade do código, a integridade do banco de dados, a navegabilidade da interface e o controle de acesso. Além disso, serão realizados testes de segurança para proteger os dados dos usuários e do sistema, bem como testes de desempenho para garantir a eficiência do aplicativo. Durante a execução, serão utilizados emuladores de celular e dispositivos reais para simular diferentes cenários de uso, com o apoio do aplicativo Expo Go. Os testes mais importantes envolvem verificar a conexão entre os sensores e o aplicativo, garantir que o banco de dados funcione corretamente e avaliar como o usuário interage com a interface.

4 REQUISITOS A SEREM TESTADOS – REQUISITOS DE QUALIDADE

Os requisitos a seguir devem ser verificados durante os testes do Qualivita, cobrindo tanto os aspectos funcionais quanto os não funcionais do sistema funcionais quanto não funcionais do sistema. Esses testes têm como objetivo garantir a eficiência, segurança e a melhor experiência de uso para o público-alvo.

4.1 Teste de Banco de Dados

O objetivo do teste de banco de dados é garantir que todas as informações sejam corretamente processadas e que o sistema seja capaz de lidar com os dados de forma eficiente. Os testes devem verificar:

- Se os dados dos usuários (nome, email, senha) são cadastrados corretamente e podem ser consultados ou excluídos sem erros.
- Se os dados coletados pelos sensores (qualidade do ar, localização dos sensores, etc.) são armazenados no banco de dados de maneira confiável.
- Se as atualizações dos dados de sensores (como medição de poluição) são sincronizadas corretamente entre o aplicativo e o banco de dados.
- Se os dados podem ser acessados de forma rápida e eficiente, mesmo com grandes volumes de informações.

4.2 Teste Funcional

Os testes funcionais têm como objetivo garantir que todas as funcionalidades do aplicativo estejam funcionando corretamente. Os testes incluirão:

- Verificar se o usuário pode acessar o aplicativo corretamente após realizar o login com suas informações.
- Testar se o aplicativo mostra corretamente os dados sobre a qualidade do ar coletados em diferentes pontos da cidade.
- Validar se o aplicativo permite a visualização e interação com os mapas de monitoramento de qualidade do ar.

- Avaliar se o sistema de atualização de dados (como a exibição de dados de sensores em tempo real) está funcionando sem falhas.
- Testar se os usuários podem realizar mudanças nas suas configurações ou dados de forma intuitiva.

4.3 Teste da Interface de Usuário

Este teste foca em garantir que a experiência do usuário seja simples, intuitiva e eficaz. Para isso, será verificado:

- Se a disposição dos ícones, botões e menus facilita a navegação do usuário.
- Se o layout das telas é claro e todas as funcionalidades estão bem identificadas.
- Se a navegação entre as telas do aplicativo é simples, sem travamentos ou dificuldade.
- Se os botões estão funcionando conforme o esperado, acionando as ações corretas (exibição de dados, configuração do perfil, etc.).
- Se o texto exibido está bem estruturado e segue as normas da língua portuguesa, evitando interpretações ou erros gramaticais.

4.4 Teste de Desempenho

O desempenho do aplicativo será testado para garantir que ele funcione de forma rápida e eficiente em diferentes cenários. Isso inclui:

- Testar o tempo necessário para gerar informações referentes à qualidade do ar, a fim de garantir que não tenha atraso significativo.
- Avaliar a performance ao processar novos dados, como adicionar informações dos sensores ou a atualização de dados no banco de dados.
- Verificar se o aplicativo responde de forma rápida e sem falhas quando o usuário realiza ações como pesquisar, filtrar ou acessar informações.

4.5 Teste de Estresse

Este teste tem como objetivo avaliar como o aplicativo se comporta em situações de alta carga. Será necessário verificar:

- Como o sistema reage ao aumento do número de usuários simultâneos realizando consultas e visualizações no mapa de qualidade do ar.
- Testar a capacidade do sistema de lidar com grandes volumes de dados (como vários sensores sendo monitorados ao mesmo tempo) sem prejudicar a performance ou estabilidade.
- Verificar a escalabilidade do banco de dados e do sistema como um todo quando tem uma alta escala de informações de usuários e sensores.

4.6 Teste de Segurança

A segurança do sistema é essencial para proteger os dados dos usuários e garantir o acesso controlado às informações. Os testes de segurança incluirão:

- Verificar se a criptografia das senhas dos usuários é feita de forma adequada para garantir a proteção dos dados.
- Avaliar se os dados sensíveis dos usuários, como localização e histórico de qualidade do ar, são armazenados de forma segura.
- Verificar se usuários não autorizados, como os que não estão cadastrados, são barrados ao tentarem acessar o sistema.
- Testar se apenas os administradores podem adicionar ou alterar dados sensíveis, como configurações de sensores e dados coletados.

4.7 Teste de Instalação e Compatibilidade

Esse teste foca em garantir que o aplicativo funcione corretamente em diferentes dispositivos e sistemas operacionais. Será verificado:

- Se o aplicativo pode ser instalado e configurado corretamente em dispositivos Android, sem erros durante o processo de instalação.

- Verificar a compatibilidade do aplicativo com versões diferentes dos sistemas operacionais Android.

4.8 Teste de Usabilidade

A usabilidade do aplicativo será verificada para garantir que os usuários tenham uma experiência satisfatória ao interagir com o sistema. Serão avaliados:

- Se a interface é intuitiva e a navegação entre as telas é simples e fluída, mesmo para usuários iniciantes.
- Se o sistema fornece informações suficientes para que o usuário entenda como utilizar as funcionalidades principais do aplicativo.
- A acessibilidade das informações, garantindo que dados importantes como a qualidade do ar e a localização dos sensores estejam facilmente acessíveis para os usuários.

5 REQUISITOS DO CLIENTE

Tabela 1: Requisitos do Cliente

Requisitos do cliente	
Requisito Básico	Requisito Fascinante
Verificar a qualidade do ar em diferentes pontos da cidade	Visualizar suas informações pessoais
Acessar histórico de dados do sensor	Visualizar localização do usuário
Visualizar os sensores no mapa	
Acessar lista dos Sensores	

6 REQUISITOS DO SOFTWARE

6.1 Requisitos Funcionais

Relacionados as atividades que o próprio sistema realizará, definindo as funções que cada tipo de usuário terá dentro do sistema ao ser acessado, alguns requisitos podem ser essenciais ou desejáveis, dependendo da necessidade deles no sistema.

Tabela 2: Coletar os Dados

Identificador	Nome	Tipo	
RF01	Coletar os dados pelo Esp32	✓ Essencial	Desejável
Descrição			
Nosso aplicativo irá coletar as informações sobre a qualidade de ar fornecida pelo Esp32 para passar ao usuário.			

Fonte: Elaborado pelos autores

Tabela 3: Mapa Funcional

Identificador	Nome	Tipo	
RF02	Mapa funcional	✓ Essencial	Desejável
Descrição			
O sistema deve conter um mapa funcional com os dados dos sensores			

Fonte: Elaborado pelos autores

Tabela 4: Integrar os Módulos

Identificador	Nome	Tipo	
RF03	Integrar o Esp32 e seus módulos	✓ Essencial	Desejável
Descrição			
O Esp32 será necessário como forma de integração dos sensores mq-135, mq-7 e DHT11 ao aplicativo, os módulos vão coletar os dados e o esp32 será responsável pela exibição e exportação para o MySQL			

Fonte: Elaborado pelos autores

Tabela 5: Cadastrar Usuário

Identificador	Nome	Tipo	
RF04	Cadastrar usuário	✓ Essencial	Desejável
Descrição			
Deverá ser fornecido uma tela de cadastro para o usuário, o cadastro no nosso aplicativo será apenas para identificação, pois não terá utilidade prática.			

Fonte: Elaborado pelos autores

Tabela 6: Tela de Login

Identificador	Nome	Tipo	
RF05	Tela de login	✓ Essencial	Desejável
Descrição			
A tela de login do aplicativo será necessária para identificar um usuário já existente, permitindo a entrada no aplicativo, nesta tela o usuário irá inserir seus dados e depois o login será possível, caso ele ainda não exista no banco de dados basta voltar a tela de cadastro			

Fonte: Elaborado pelos autores

Tabela 7: Exibir informações dos Sensores

Identificador	Nome	Tipo	
RF6	Exibir dados dos sensores em geral	✓ Essencial	Desejável
Descrição			
A aplicação também vai exibir as informações sobre o ar coletadas em uma tela com a informação de vários sensores juntas, como uma lista.			

Fonte: Elaborado pelos autores

Tabela 8: Tela de Histórico

Identificador	Nome	Tipo	
RF7	Tela para visualização de outros dados coletados anteriormente	✓ Essencial	Desejável
Descrição			
Tela onde os usuários poderão visualizar as informações antigas do mesmo sensor.			

Fonte: Elaborado pelos autores

Tabela 9: Notificar Usuário

Identificador	Nome	Tipo	
RF8	Notificar	Essencial	✓ Desejável
Descrição			
O usuário deve receber uma notificação caso a qualidade do ar mais próxima a ele estiver em nível crítico			

Fonte: Elaborado pelos autores

Tabela 10: Redefinir Senha

Identificador	Nome	Tipo	
RF9	Redefinir Senha	✓ Essencial	Desejável
Descrição			
Tela para o usuário redefinir sua senha em caso de perda.			

Fonte: Elaborado pelos autores

Tabela 11: Configurações

Identificador	Nome	Tipo	
RF10	Configurações	Essencial	✓ Desejável
Descrição			
Tela de configurações onde usuário poderá ver informações de seu cadastro e o funcionamento dos sensores.			

Fonte: Elaborado pelos autores

6.2 Requisitos Não Funcionais

Os requisitos Não Funcionais determinam o comportamento do sistema. Eles podem decidir qual o alcance de eficácia do sistema e é dividido em requisitos Essencial e Desejável como dito anteriormente nos requisitos funcionais.

Tabela 12: Permissão para acesso da localização

Identificador	Nome	Tipo	
RNF01	Permissão para acessar a localização do usuário	Essencial	✓ Desejável
Descrição			
Permissão do usuário para acessar sua localização por meio de seu dispositivo.			

Fonte: Elaborado pelos autores

Tabela 13: Conexão à internet

Identificador	Nome	Tipo	
RNF02	Conexão à internet	✓ Essencial	Desejável
Descrição			
Será necessário o usuário estar conectado à internet.			

Fonte: Elaborado pelos autores

Tabela 14: O aplicativo rodará no sistema operacional Android

Identificador	Nome	Tipo	
RNF03	O aplicativo rodará apenas no Android	✓ Essencial	Desejável
Descrição			
O usuário deverá ter um dispositivo com sistema operacional Android para usar o aplicativo e monitorar a qualidade de ar de sua região.			

Fonte: Elaborado pelos autores

Tabela 15: Conexão ao banco de dados

Identificador	Nome	Tipo	
RNF04	Conexão ao banco de dados	✓ Essencial	Desejável
Descrição			
O usuário e o sensor deverão estar conectados ao banco de dados do sistema para acessar e usufruir das funcionalidades do aplicativo.			

Fonte: Elaborado pelos autores

Tabela 16: Uso de API

Identificador	Nome	Tipo	
RNF05	Uso de API	✓ Essencial	Desejável
Descrição			
Uso de API (Application Programming Interface) para fornecer o mapa do sistema, e para integrar os dados dos sensores ao sistema.			

Fonte: Elaborado pelos autores

Tabela 17: Suporte ao volume de dados

Identificador	Nome	Tipo	
RNF06	Suporte ao volume de dados	✓ Essencial	Desejável
Descrição			
O dispositivo e o aplicativo deverão ser capazes de lidar com o volume de dados sem comprometer o desempenho ou a precisão dos dados coletados.			

Fonte: Elaborado pelos autores

Tabela 18: Consumo de energia

Identificador	Nome	Tipo	
RNF07	Consumo de energia	✓ Essencial	Desejável
Descrição			
O aplicativo deve ser otimizado para minimizar o consumo de energia do dispositivo móvel e do microprocessador Esp32, prolongando a vida útil da bateria e garantindo o funcionamento contínuo do sistema de monitoramento.			

Fonte: Elaborado pelos autores

7 ESTRATÉGIAS E FERRAMENTAS DE TESTE

7.1 Teste do Banco de Dados

Objetivo do Teste:	Avaliar a eficácia, confiabilidade, desempenho e segurança do sistema de gerenciamento de banco de dados, garantindo que ele armazene e recupere os dados dos sensores corretamente, atendendo aos requisitos funcionais e não funcionais.
Técnica:	Garantir que as relações entre as tabelas estejam corretas; verificar se não tem cópias nas chaves primárias e únicas; garantir que os dados são transferidos corretamente e mantêm sua integridade.
Critério de Finalização:	O sistema foi capaz de armazenar e recuperar os dados dos sensores corretamente, mantendo a integridade e eficiência.
Considerações Especiais:	Teste concluído com êxito, sem falhas na recuperação e armazenamento de dados.

Fonte: Elaborado pelos autores

7.2 Teste Funcional

Objetivo do Teste:	Garantir que o aplicativo atenda aos requisitos funcionais, verificando se o sistema executa as tarefas essenciais de monitoramento de qualidade do ar e visualização dos dados de maneira eficiente.
Técnica:	Verificação das entradas e saídas dos dados, garantindo que as informações de qualidade do ar sejam exibidas corretamente; avaliação da interação do usuário com a interface do aplicativo.
Critério de Finalização:	O sistema deve funcionar sem falhas, exibindo dados de qualidade do ar corretamente e permitindo uma interação satisfatória com o usuário.
Considerações Especiais:	Teste concluído com êxito, sem falhas nas funcionalidades principais.

Fonte: Elaborado pelos autores

7.3 Teste da Interface do Usuário

Objetivo do Teste:	Garantir que a interface do aplicativo Qualivita seja intuitiva e fácil de usar, oferecendo uma experiência de usuário clara e eficiente.
Técnica:	Avaliação da integração entre a interface e os dados, garantindo que as informações sobre a qualidade do ar sejam exibidas corretamente; avaliação da navegação entre as telas.
Critério de Finalização:	A interface deve ser didática, com rápida navegação e elementos visuais consistentes e funcionais.
Considerações Especiais:	Teste concluído com êxito, com navegação intuitiva e rápida resposta dos elementos gráficos.

Fonte: Elaborado pelos autores

7.4 Teste de Performance

Objetivo do Teste:	Avaliar o desempenho do aplicativo em cenários de uso real, garantindo que o sistema funcione de maneira eficiente, mesmo com múltiplos sensores e usuários.
Técnica:	Teste de inserção e consulta de grandes volumes de dados de sensores para avaliar o impacto no desempenho; avaliação do tempo de resposta do sistema em diferentes condições de carga.
Critério de Finalização:	O sistema deve manter um desempenho adequado, com tempo de resposta satisfatório e sem falhas, mesmo sob carga.
Considerações Especiais:	Teste concluído com êxito, com o sistema funcionando sem falhas críticas.

Fonte: Elaborado pelos autores

7.5 Teste de Carga

Objetivo do Teste:	Avaliar a capacidade do sistema em lidar com múltiplos usuários e sensores simultaneamente, sem comprometer o desempenho.
Técnica:	Simulação de múltiplos usuários acessando os dados de sensores ao mesmo tempo; teste de inserção de vários dados em curto intervalo.
Critério de Finalização:	O sistema deve operar de forma estável sob carga máxima, com boa performance.
Considerações Especiais:	Teste concluído com êxito, com o sistema operando de forma estável sob carga intensa.

Fonte: Elaborado pelos autores

7.6 Teste de Segurança e Controle de Acesso

Objetivo do Teste:	Garantir que os mecanismos de controle de acesso estejam funcionando adequadamente, protegendo os dados dos usuários e impedindo acessos não autorizados.
Técnica:	Teste de segurança com tentativas de acesso não autorizado; verificação da robustez dos mecanismos de autenticação e controle de sessões.
Critério de Finalização:	O sistema deve resistir a tentativas de acesso indevido, com os dados dos usuários protegidos e os controles de acesso funcionando corretamente.
Considerações Especiais:	Teste concluído com êxito, com todos os controles de segurança funcionando sem falhas.

Fonte: Elaborado pelos autores

7.7 Teste de Instalação

Objetivo do Teste:	Garantir que o aplicativo Qualivita seja instalado corretamente em diferentes dispositivos e sistemas operacionais.
Técnica:	Teste de instalação em diferentes dispositivos Android, com verificação de compatibilidade e configuração correta do sistema.
Critério de Finalização:	O sistema deve ser instalado corretamente em dispositivos diversos, com todos os arquivos necessários e componentes instalados.
Considerações Especiais:	Teste concluído com êxito.

Fonte: Elaborado pelos autores

8 EQUIPE E INFRAESTRUTURA

NOME COMPLETO DOS INTEGRANTES	E-MAIL
Gustavo Ramos da Silva Santos	gustavo.santos1122@etec.sp.gov.br
Lucas Donizeti dos Santos	lucas.santos2346@etec.sp.gov.br
João Vitor Pereira de Jesus	joao.jesus99@etec.sp.gov.br

Recursos do Sistema
Servidor de Banco de Dados Google Firebase Google Firebase Firestore Google Firebase Authentication
Terminais Clientes Smartphones (com acesso à internet, compatíveis com Android)
Repositório de Testes <ul style="list-style-type: none"> 3 Notebooks de Desenvolvimento para Teste (para simulação de uso em diferentes dispositivos)

9 CRONOGRAMA

MILESTONE	DATA INICIO	DATA FIM
Planejar Teste	11/11/2024	12/11/2024
Projetar Teste	12/11/2024	13/11/2024
Implementar Teste	15/11/2024	19/11/2024
Executar Teste	19/11/2024	23/11/2024
Avaliar Teste	24/11/2024	24/11/2024

10 CONCLUSÃO

Após as diversas fases de testes e validações das funcionalidades do Qualivita, foi possível constatar que o projeto cumpriu com sucesso os critérios de desempenho e aplicabilidade definidos. Dessa forma é claro que a solução proposta pode ser implementada de maneira eficiente, garantindo que todos os requisitos exigidos pelos usuários sejam atendidos, promovendo assim um impacto positivo no monitoramento da qualidade do ar e no uso consciente dessas informações.