

Um aplicativo para coleta de dados de sensoriamento em sistemas de dessalinização em comunidades carentes

Natan Ventura Menezes¹

Orientador: Thiago Ferreira de Noronha¹

Coorientadora: Anolan Yamilé Milanés Barrientos²

¹ Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

² Departamento de Computação - Centro Federal de Educação Tecnológica de Minas Gerais

Resumo. *Este projeto desenvolve um sistema de monitoramento para plantas de dessalinização utilizando o ESP32 e um aplicativo móvel. O sistema coleta dados de sensores de pH, temperatura, sensor que mede sólidos dissolvidos totais e os transforma através de uma fórmula em condutividade elétrica, transmitindo-os via Bluetooth para o aplicativo, desenvolvido com Flutter. O aplicativo possui três telas principais: conexão com o ESP32, recepção e compartilhamento de dados, e visualização de relatórios. Até agora, o sistema de compartilhamento de dados e o aplicativo básico foram desenvolvidos com sucesso. A próxima etapa incluirá detecção de falhas, padronização dos dados e envio ao servidor, garantindo um monitoramento robusto e eficiente.*

1. Introdução

1.1. Escopo

Há um projeto em andamento envolvendo o Departamento de Ciência da Computação (DCC) da UFMG para criar um sistema de sensoriamento do sistema atual de dessalinização. Este projeto envolve o desenvolvimento de um sistema utilizando o microcontrolador ESP32, além de sensores de pH, temperatura, sensor que mede sólidos dissolvidos totais e os transforma através de uma fórmula em condutividade elétrica. Com esses sensores, pretende-se obter dados sobre os sistemas de dessalinização e detectar possíveis problemas na qualidade da água. No futuro, essas informações podem ser utilizadas em técnicas de machine learning para analisar os dados coletados, permitindo a otimização e a manutenção preditiva das plantas de dessalinização.

Este trabalho se propõe a criar um sistema de transmissão dos dados entre o ESP32 e o aplicativo Android dos operadores. O escopo específico inclui a implementação da comunicação via Bluetooth entre o microcontrolador ESP32 e dispositivos Android, configurando o ESP32 para coletar e transmitir dados dos sensores de pH, temperatura, oxigênio dissolvido e condutividade elétrica. A escolha do Bluetooth como meio de comunicação se deve à falta de garantia de internet móvel nos pontos de dessalinização. Pesquisas feitas pelo projeto mostraram que, embora a internet móvel seja limitada nesses locais, os operadores possuem acesso à internet em seus lares, onde poderão transmitir os dados coletados.

Além disso, será desenvolvido um aplicativo utilizando Flutter para receber, processar e exibir os dados transmitidos pelo ESP32, com uma interface de usuário intuitiva

que permita aos operadores visualizar os dados de forma clara e eficiente. O aplicativo também contará com a funcionalidade de enviar os dados coletados inicialmente via e-mail para as secretarias responsáveis, com uma proposta futura para a criação de um servidor web onde os dados poderão ser armazenados e analisados em maior escala. Além disso, toda a configuração dos sensores, programação do ESP32 e implementação do aplicativo serão detalhadamente documentadas. Com este escopo específico, o trabalho busca fornecer uma solução tecnológica eficaz para melhorar o monitoramento e a qualidade dos sistemas de dessalinização, contribuindo para a sustentabilidade e a eficiência operacional das plantas de dessalinização no âmbito do Projeto Água Doce.

A motivação para esse Projeto Orientado em Computação é melhorar o monitoramento e a qualidade dos sistemas de dessalinização nas comunidades rurais do Semiárido brasileiro, onde a coleta manual de dados seria propensa a erros e inconsistências, comprometendo a qualidade da água. Durante uma visita, verificou-se que água esverdeada estava sendo consumida sem que o operador entendesse que aquilo era um problema, algo que teria sido detectado com um monitoramento efetivo. Dessa forma, poderemos diminuir o deslocamento das equipes de campo e melhorar o feedback dos operadores para a gestão estadual.

A falta de internet móvel confiável nos pontos de dessalinização dificulta a transmissão de dados em tempo real. Para resolver esses problemas, implementaremos um sistema automatizado de sensoriamento utilizando o microcontrolador ESP32 e sensores para medir pH, temperatura, oxigênio dissolvido e condutividade elétrica. Usaremos Bluetooth para comunicação com dispositivos Android, permitindo que os operadores transmitam os dados de suas casas, onde possuem acesso à internet. O desenvolvimento de um aplicativo Android intuitivo facilitará a visualização e o compartilhamento desses dados, aumentando a precisão da coleta e capacitando os operadores locais. A longo prazo, os dados poderão ser usados em técnicas de machine learning para análises preditivas e otimização dos sistemas de dessalinização, garantindo um fornecimento de água potável mais eficiente e sustentável.

1.2. Objetivos

Os objetivos deste projeto orientado em computação 1 são amplos e detalhados, visando um aprofundamento em várias áreas cruciais para o desenvolvimento de um sistema de monitoramento de dessalinização. Inicialmente, o projeto busca entender mais sobre o desenvolvimento de Internet das Coisas (IoT) utilizando o microcontrolador ESP32, programado em C++. Simultaneamente, é essencial compreender o desenvolvimento mobile com o Flutter, uma ferramenta poderosa para a criação de aplicativos móveis. Definir claramente o escopo do projeto é um passo fundamental, garantindo que todas as mudanças necessárias sejam realizadas em conformidade com a metodologia ágil, que permite ajustes rápidos e eficientes durante o desenvolvimento. Além disso, é crucial entender a tecnologia Bluetooth Low Energy (BLE), incluindo seu funcionamento e as bibliotecas disponíveis para ambos os sistemas, ESP32 e Flutter.

Outro objetivo importante é definir um protocolo que detecte e evite perdas na transmissão de dados sem fio, assegurando a integridade das informações coletadas. Também é necessário definir o padrão de arquivo JSON que será utilizado na comunicação entre os dispositivos, garantindo uma troca de dados consistente e estruturada

O projeto também inclui o desenvolvimento do design das telas do aplicativo, proporcionando uma interface de usuário intuitiva e eficaz. A modelagem do sistema será feita utilizando cartões CRC (Classes, Responsabilidades e Colocações), uma técnica que facilita a visualização e organização das partes do sistema. Por fim, o desenvolvimento do sistema será iniciado, estabelecendo uma base sólida que será continuada na próxima fase do projeto (POC 2). Estes objetivos combinam aspectos teóricos e práticos, proporcionando uma compreensão abrangente e aplicada das tecnologias envolvidas.

A justificativa para os objetivos deste projeto é desenvolver um sistema robusto e eficiente para o monitoramento de plantas de dessalinização, integrando tecnologias modernas de IoT com o ESP32 programado em C++ e desenvolvimento mobile com Flutter. Entender essas tecnologias é essencial para criar uma solução de sensoriamento precisa e um aplicativo intuitivo. A definição do escopo e o uso da metodologia ágil garantem flexibilidade e adaptação, enquanto a compreensão da tecnologia BLE assegura uma comunicação eficiente entre dispositivos.

Além disso, a definição de um protocolo de transmissão de dados robusto é necessária para evitar perdas de informação, e o uso de um padrão de arquivo JSON facilita a integração dos componentes do sistema. O design das telas do aplicativo e a modelagem do sistema com cartões CRC proporcionam uma interface amigável e uma organização clara das responsabilidades. Iniciar o desenvolvimento do sistema estabelece uma base sólida que será continuada na próxima fase do projeto (POC 2), garantindo melhorias contínuas e a sustentabilidade dos sistemas de dessalinização.

2. Metodologia

2.1. Metodologia Ágil

A metodologia ágil é uma abordagem de gerenciamento de projetos que promove flexibilidade, colaboração e entrega contínua de valor. Ela divide o trabalho em ciclos curtos chamados iterações ou sprints, geralmente de uma a quatro semanas. Durante cada sprint, a equipe planeja o trabalho, desenvolve funcionalidades, realiza testes e coleta feedback. O processo começa com o planejamento da iteração, onde a equipe revisa o backlog do produto e define as tarefas para o sprint. [Valente 2020] Para este projeto, adotaremos uma abordagem ágil adaptada às nossas necessidades específicas, garantindo flexibilidade e foco na entrega contínua de valor. Ao invés das reuniões diárias tradicionais, realizaremos uma reunião semanal para verificar o progresso do projeto, permitindo discussões sobre as atividades realizadas e ajustes no plano de ação. Nossos sprints serão mais flexíveis em termos de duração e escopo, possibilitando adaptações rápidas às mudanças de requisitos e prioridades. Coletaremos feedback continuamente durante as reuniões semanais.

No desenvolvimento de projetos ágeis, a clareza e a organização dos requisitos são fundamentais. Histórias de usuário e épicos são ferramentas essenciais para alcançar essa clareza, garantindo que todos os membros da equipe e stakeholders compreendam as necessidades e objetivos do projeto. Utilizar histórias de usuário e épicos facilita a decomposição de grandes funcionalidades em tarefas menores e manejáveis, promovendo uma entrega incremental de valor.

Épicos são grandes corpos de trabalho que representam funcionalidades amplas e complexas dentro de um projeto. Eles não podem ser completados em um único sprint

e, portanto, são divididos em histórias de usuário menores e mais específicas. Os épicos fornecem uma visão geral do que precisa ser realizado, permitindo que a equipe mantenha o foco nos objetivos de longo prazo enquanto trabalha nas tarefas diárias. Histórias de usuário são descrições curtas e simples de uma funcionalidade ou recurso desejado do ponto de vista do usuário final. Elas são escritas em uma linguagem não técnica e têm como objetivo capturar os requisitos de maneira clara e compreensível, ajudando a equipe a entender exatamente o que é necessário e a priorizar o trabalho de forma eficiente.

No contexto do nosso projeto, as histórias de usuário seriam e épicos e podem ser descritos da seguinte forma:

Épico: Desenvolvimento de um Aplicativo Mobile para receber dados do ESP32

- **História de Usuário:** "Como um operador, eu quero receber os dados coletados pelos sensores desde a última visita."
- **História de Usuário:** "Como um operador, eu quero uma interface intuitiva no aplicativo para que eu possa acessar e interpretar facilmente os dados coletados pelos sensores."
- **História de Usuário:** "Como um operador, eu quero enviar os dados coletados para as secretarias responsáveis para garantir que as informações sejam compartilhadas de forma eficiente."

Épico: Implementação da Comunicação BLE

- **História de Usuário:** "Como um operador, eu quero transmitir os dados coletados pelo ESP32 para o aplicativo via Bluetooth para garantir a continuidade do monitoramento sem depender de internet móvel."
- **História de Usuário:** "Como um operador, eu quero que a transmissão de dados via Bluetooth seja confiável e sem perdas, para que todas as informações coletadas sejam precisas e completas."
- **História de Usuário:** "Como gerente do sistema, quero que somente usuários autenticados consigam receber os dados do ESP32."

Épico: Armazenamento de dados no servidor para utilização futura em pesquisas de ciência de dados

- **História de Usuário:** "Como cientista de dados quero que os dados sejam enviados do celular ao servidor para que eu possa utilizá-los em pesquisas."
- **História de Usuário:** "Como cientista de dados quero que os dados estejam padronizados para utilização em pesquisa."
- **História de Usuário:** "Como gerente do sistema, quero que somente usuários autenticados consigam enviar os dados ao servidor."

2.2. Modelagem das telas

Realizamos a modelagem de telas especificamente para a validação das funcionalidades com os engenheiros envolvidos. Esses protótipos visuais foram apresentados aos engenheiros para garantir que todas as funcionalidades necessárias estivessem corretamente representadas e operacionais. As telas foram aprovadas pelos engenheiros, confirmando que o design atende aos requisitos técnicos e funcionais do sistema.



Figura 1. Telas modeladas

A primeira tela exibe dois botões principais: "Receber Dados" e "Visualizar Dados". O botão "Receber Dados" permite ao usuário iniciar o processo de conexão com o ESP32 via Bluetooth, enquanto o botão "Visualizar Dados" direciona o usuário para a tela onde pode acessar os relatórios de dados já recebidos.

Ao clicar em "Receber Dados", o usuário é levado à segunda tela, que exibe uma lista de dispositivos Bluetooth encontrados, incluindo o ESP32 e outros dispositivos disponíveis. O usuário pode selecionar o ESP32 da lista para iniciar a conexão. Um botão "Buscar" está disponível para atualizar a lista de dispositivos detectados. Após a conexão bem-sucedida, a terceira tela confirma que os dados foram recebidos com sucesso, exibindo uma mensagem visual de confirmação e um botão "Visualizar Dados" para acessar os dados recebidos.

Na quarta tela, são listados os relatórios de dados de sensoriamento organizados por data. Cada relatório pode ser visualizado clicando no ícone de olho, ou compartilhado clicando no ícone de compartilhamento. Finalmente, a quinta tela exibe os detalhes do relatório selecionado, mostrando a data do relatório e uma avaliação geral da qualidade dos dados (Ruim, Regular, Bom) através de ícones expressivos. Os dados específicos como TDS, Vazão, Temperatura e Pressão são apresentados com indicadores de status, permitindo uma rápida interpretação dos parâmetros monitorados. Essas telas foram criadas para assegurar que os operadores possam facilmente conectar-se ao ESP32, receber, visualizar e compartilhar dados de maneira eficiente, melhorando a gestão da qualidade da água nas plantas de dessalinização. Vale ressaltar que esse é um protótipo inicial de telas podendo sofrer alterações conforme as necessidades percebidas no decorrer do projeto.

2.3. Tecnologias

A escolha do ESP32 foi baseada em sua versatilidade, eficiência energética e robustez para operar em ambientes adversos. O ESP32 é um microcontrolador de baixo custo e alta performance, equipado com capacidades de Wi-Fi e Bluetooth, ideal para aplicações de Internet das Coisas (IoT). No nosso projeto, o ESP32 é responsável por coletar dados de vários sensores (pH, temperatura, oxigênio dissolvido, condutividade elétrica) e transmitir esses dados para dispositivos móveis via Bluetooth. [Mohammed et al. 2023]

Utilizamos Flutter para desenvolver o aplicativo móvel do projeto, aproveitando sua capacidade de criar interfaces de usuário responsivas e intuitivas. Flutter é um framework de desenvolvimento de aplicativos móveis criado pelo Google, que permite a criação de interfaces de usuário nativas para iOS e Android a partir de uma única base de código. A escolha de Flutter permite um desenvolvimento rápido e eficiente, com uma

performance nativa em ambas as plataformas. [Yu et al. 2019]

A tecnologia Bluetooth Low Energy (BLE) é utilizada para a comunicação entre o ESP32 e os dispositivos móveis. BLE é uma tecnologia sem fio de baixa potência, ideal para aplicações de IoT que requerem uma conexão eficiente e de baixo consumo energético. Utilizamos BLE para garantir uma transmissão de dados confiável e contínua entre os sensores e o aplicativo móvel, mesmo em locais com limitações de conectividade de internet.

2.4. Arquitetura

Os cartões CRC (Classe-Responsabilidade-Colaborador) são ferramentas importantes na engenharia de software por várias razões. Eles ajudam a definir claramente o que cada classe deve fazer, evitando a sobrecarga de responsabilidades em uma única classe. Promovem uma melhor compreensão de como diferentes classes colaboram entre si, essencial para um design coeso e bem estruturado. Durante o design orientado a objetos, os cartões CRC são úteis para identificar e organizar classes de forma sistemática. Com responsabilidades bem definidas, o código se torna mais modular e, portanto, mais fácil de manter, testar e modificar. Além disso, servem como uma ferramenta de comunicação eficaz entre membros da equipe, facilitando a revisão e a discussão do design do sistema.

A arquitetura MVC (Model-View-Controller) é um padrão de design que separa a aplicação em três componentes principais, cada um com responsabilidades distintas. O Modelo (Model) representa a lógica de negócios e os dados da aplicação, sendo responsável por acessar e gerenciar os dados e assegurar que as regras de negócios são aplicadas. A Visualização (View) é responsável pela apresentação dos dados ao usuário e atualiza a interface do usuário com base nas mudanças nos dados do modelo. O Controlador (Controller) interpreta as entradas do usuário e as converte em comandos para o modelo ou a visualização. Esta separação de responsabilidades facilita a manutenção, teste e escalabilidade do sistema, além de permitir que desenvolvedores trabalhem em paralelo em diferentes componentes da aplicação sem interferência direta, promovendo uma maior modularidade e flexibilidade no desenvolvimento de software. Utilizaremos de ambos conceitos para propor nosso sistema conforme pode ser visto abaixo:

Modelos

Report

Classe: Report

- **Responsabilidades:**
 - Armazenar informações de um relatório
 - Gerenciar dados e formato do relatório
 - Validar conteúdo do relatório
- **Colaboradores:**
 - File (para salvar e carregar relatórios)
 - Device (para enviar relatórios)

File

Classe: File

- **Responsabilidades:**
 - Gerenciar operações de leitura e escrita em arquivos
 - Armazenar metadados de arquivos
- **Colaboradores:**
 - Report (para salvar e carregar relatórios)
 - FileController (para operações de arquivos)

Device

Classe: Device

- **Responsabilidades:**
 - Gerenciar informações do dispositivo
 - Conectar e desconectar de redes
 - Enviar e receber dados via BLE
- **Colaboradores:**
 - BLEController (para gerenciar conexões BLE)
 - Report (para enviar relatórios)

Controladores

BLEController

Classe: BLEController

- **Responsabilidades:**
 - Gerenciar conexões BLE
 - Iniciar e encerrar sessões BLE
 - Transmitir dados entre dispositivos
 - Assegurar a segurança nas transmissões
- **Colaboradores:**
 - Device (para gerenciar informações e conexões do dispositivo)
 - Report (para enviar relatórios via BLE)

FileController

Classe: FileController

- **Responsabilidades:**
 - Gerenciar operações de arquivos
 - Criar, abrir, salvar e deletar arquivos
 - Integrar com o modelo File
- **Colaboradores:**
 - File (para operações de arquivos)
 - FilesScreen (para exibir status de operações de arquivos)

Visualizações

MainScreen

Classe: MainScreen

- **Responsabilidades:**
 - Pedir permissões do aparelho
 - Scannear dispositivos Bluetooth
- **Colaboradores:**
 - BLEController (para leitura de dispositivos)

FilesScreen

Classe: FilesScreen

- **Responsabilidades:**
 - Exibir lista de arquivos
 - Permitir operações de abrir, salvar e deletar arquivos
 - Mostrar detalhes e metadados dos arquivos
- **Colaboradores:**
 - FileController (para operações de arquivos)
 - File (para exibir detalhes dos arquivos)

ReportScreen

Classe: ReportScreen

- **Responsabilidades:**
 - Exibir relatório detalhado
 - Salvar alterações do relatório
- **Colaboradores:**
 - Report (para exibir e editar relatórios)
 - FileController (para salvar relatórios editados)

3. progresso atual

Até o momento, alcançamos avanços significativos no desenvolvimento do nosso projeto de monitoramento de plantas de dessalinização. Desenvolvemos um sistema no microcontrolador ESP32 capaz de compartilhar arquivos de dados de sensoriamento com dispositivos móveis. Além disso, implementamos um aplicativo móvel simples com três telas funcionais.

A primeira tela do aplicativo permite ao usuário conectar-se ao ESP32 via Bluetooth. Esta interface facilita a detecção e o emparelhamento com o dispositivo de sensoriamento. Após a conexão, o usuário é direcionado para a segunda tela, onde os arquivos de dados podem ser recebidos do ESP32. Nesta tela, os usuários também têm a opção de compartilhar os arquivos recebidos por e-mail ou outros meios, garantindo que os dados possam ser facilmente distribuídos às autoridades competentes.

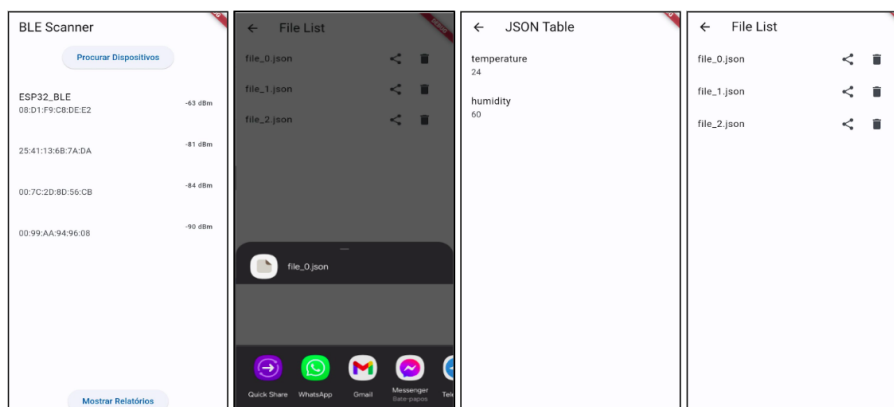


Figura 2. Capturas de tela do aplicativo no estado atual

A terceira tela do aplicativo exibe os relatórios recebidos, permitindo aos operadores visualizar os dados de sensoriamento de maneira organizada e intuitiva. Esta funcionalidade é essencial para o monitoramento contínuo e a avaliação da qualidade da água nas plantas de dessalinização. Na terceira tela podemos selecionar um relatório e visualizá-lo na quarta tela. Na quarta tela exibimos o relatório JSON em uma versão mais amigável ao usuário, mostrando cada atributo e seu respectivo valor medido. Vamos apresentar prints do aplicativo em execução para ilustrar essas funcionalidades e demonstrar o progresso realizado até agora.

4. Conclusão Parcial

Até o momento, avançamos significativamente no desenvolvimento do nosso sistema de monitoramento para plantas de dessalinização. Desenvolvemos com sucesso um sistema no microcontrolador ESP32 para compartilhar arquivos de dados de sensoriamento e implementamos um aplicativo móvel funcional com três telas principais. Fornecendo uma interface inicial amigável e intuitiva ao usuário.

Cada uma dessas etapas foi essencial para criar uma base sólida para o nosso sistema. A conexão Bluetooth confiável e a transferência de arquivos asseguram que os dados de sensoriamento possam ser capturados e transmitidos de maneira eficiente. A capacidade de visualizar e compartilhar relatórios diretamente pelo aplicativo móvel facilita a disseminação das informações para as autoridades competentes, garantindo a transparência e a continuidade do monitoramento da qualidade da água.

Na próxima etapa do projeto, planejamos expandir as funcionalidades do sistema com foco em autenticação, detecção de falhas, padronização dos arquivos de dados e envio de dados ao servidor. A autenticação garantirá que apenas usuários autorizados possam acessar e manipular os dados sensíveis. A detecção de falhas será crucial para identificar e corrigir problemas no sistema de sensoriamento em tempo real, melhorando a confiabilidade e a precisão dos dados. A padronização dos arquivos assegurará a consistência e a integridade dos dados transmitidos, facilitando a análise e o processamento subsequentes. Finalmente, o envio dos dados ao servidor permitirá o armazenamento centralizado e o acesso remoto às informações, aprimorando ainda mais a capacidade de monitoramento e análise.

Esses desenvolvimentos são vitais para a conclusão bem-sucedida do projeto e

garantirão que nosso sistema de monitoramento seja robusto, seguro e eficaz, atendendo plenamente às necessidades das plantas de dessalinização e das comunidades que dependem delas.

Referências

Mohammed, S., Khan, A., Ahmad, A., and Ullah, A. (2023). Design and implementation of esp32-based iot devices. *Sensors*, 23(2):345.

Valente, M. T. (2020). *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. Editora: Independente.

Yu, M.-C., Wang, C.-L., and Hsu, L.-W. (2019). A framework for cross-platform mobile application development using flutter. In *Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pages 468–477. IEEE.