

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências exatas e de tecnologia

Departamento de Computação

Trabalho de conclusão de curso - TCC

Fernando Ogata Marubayashi

**Glycos – Sistema IoT de monitoramento de
glicemia e auxílio à prevenção e ao tratamento
da Diabetes Mellitus**

São Carlos - São Paulo

2023

Fernando Ogata Marubayashi

Glycos – Sistema IoT de monitoramento de glicemia e auxílio à prevenção e ao tratamento da Diabetes Mellitus

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação como parte dos requisitos para a conclusão da graduação em Engenharia de Computação.

Orientação Prof. Dr. Fredy J. Valente

São Carlos - São Paulo

2023

Agradecimentos

Agradeço aos meus pais, Ines Komoguchi Ogata e Nilton Nobuo Marubayashi, pelo constante apoio durante toda minha trajetória de vida, todos os conselhos e ensinamentos que levo comigo para sempre. Obrigado por me ensinarem as coisas mais importantes da vida e me permitirem crescer ao lado de vocês. Agradeço também à minha avó (ba-chan) Mitico Komoguchi Ogata por estar sempre na torcida para que eu nunca desistisse dos meus objetivos e sempre ser a primeira pessoa a ficar feliz quando os alcançava. Por fim, agradeço aos amigos que fiz ao longo da caminhada, estes que foram responsáveis por tirar o peso e pressão da faculdade em inúmeras ocasiões. Tenho um orgulho infinito de poder dedicar este trabalho a vocês.

*“Eu gostaria que houvesse uma maneira de saber que você está nos bons e velhos tempos
antes de realmente deixá-los.”
(Andy Bernard)*

Resumo

Em 2020 o número de pessoas no Brasil que possuem diabetes, ou *Diabetes Mellitus (DM)*, está por volta de 16.8 milhões de acordo com a pesquisa feita pela *International Diabetes Federation*. Com esse número significativo em mente e, utilizando conceitos de Internet das Coisas (Internet of Things, IoT) foi criado um aparelho capaz de medir a glicose capilar a fim de facilitar o tratamento e autocuidado de pessoas com o DM ou pré-diabéticas. Além de medir a glicose, o aparelho conhecido como glicosímetro possui conectividade Wi-Fi integrada, ou seja, todas as medições realizadas podem ser enviadas diretamente para um servidor na nuvem onde os dados podem ser guardados num banco de dados. Esses dados de medidas consistem não só da glicose sanguínea, mas também de diversas perguntas como “Está praticando exercício físico?” ou “Está utilizando medicação corretamente” que permitem com que uma técnica de inteligência artificial, conhecida como lógica fuzzy, seja aplicada. Essa técnica é capaz de prever a tendência da glicemia com base nos dados de medidas anteriores para identificar os padrões de resposta do usuário, assim apontando se existem chances da glicemia aumentar, diminuir ou estabilizar. Após o tratamento de dados, o usuário, assim como seu médico e familiares, terão acesso a esses por meio de uma plataforma *Progressive Web App (PWA)* que roda nativamente pelo navegador de qualquer dispositivo *smart-phone, tablet* ou *desktop*. Com esses dados de fácil acesso e uma tendência calculada a partir deles, o atendimento e tratamento para pacientes com o DM será mais ágil e preciso nas medidas preventivas, além de proporcionar diferentes visões ao médico. Ao longo do trabalho diversos testes foram conduzidos a fim de testar a lógica e a qualidade do sistema numa espécie de homologação do mesmo. Os resultados obtidos comprovam o funcionamento tanto da previsão de tendência quanto do sistema como um todo, mostrando que testes com pacientes ou dados reais anônimos poderiam ser conduzidos num futuro próximo.

Palavras-chave: Diabetes Mellitus. Glicosímetro. Internet das Coisas. IoT. Progressive Web App. PWA. Nuvem. Inteligência Artificial. Lógica Fuzzy. MQTT. ESP32.

Abstract

In 2020 the number of people in Brazil that had diabetes, or Diabetes Mellitus (DM), was around 16.8 million according to the research done by the International Diabetes Federation. With this meaningful number in mind and utilizing concepts of Internet of Things (IoT) a device capable of measuring capillary blood glucose was created to assist with the treatment and self-care of people with DM or in states of pre-diabetes. The device also known as a glucometer also has the capability to connect itself to the internet through an integrated Wi-Fi module, which means that all measures are directly sent to a cloud server where the data can be kept in a database. The data consists not only of the blood sugar but also questions like "Have you been practicing physical exercise?" or "Are you taking the medicine correctly?" which allow an Artificial Intelligence technique to take place known as fuzzy logic. This technique is capable of predicting the glycemic levels based on past measures and the answers to the other questions, thus being able to say if there are chances of it raising, lowering or stabilizing. After the data treatment, the user along with their doctor and family members will have access to all the information through a Progressive Web App (PWA) that runs natively in the browser of any smartphone. With this easy-to-access data and a prediction made based on said data, the treatment and general service to patients with DM will be more agile, precise on the preventive measures and supply the medical staff with other point of views. Along the project, a number of tests were conducted in order to test the logic and quality of the system in a homologation process. The results prove that both the trend prediction and the system work well, and that it'll be possible to move into tests with patients or real anonymous data in the future.

Keywords: Diabetes Mellitus. Glucometer. Internet of Things. IoT. Progressive Web App. PWA. Cloud. Artificial Intelligence. Fuzzy Logic. MQTT. ESP32

Lista de ilustrações

Figura 1 – Fluxo do sistema. Fonte: Autor	16
Figura 2 – Conjunto fuzzy com variável linguística “Jovem”. Fonte: Autor	20
Figura 3 – Sistema fuzzy. Fonte: Autor	21
Figura 4 – Sistema <i>Publish & Subscriber</i> com <i>broker</i> . Fonte: Autor	22
Figura 5 – Tabela de usuários. Fonte: Autor	23
Figura 6 – Tabela de medidas. Fonte: Autor	24
Figura 7 – Tabela de observações. Fonte: Autor	24
Figura 8 – Tela de tabela de medidas. Fonte: Autor	26
Figura 9 – Tela de tabela de medidas em dispositivo móvel. Fonte: Autor	27
Figura 10 – Formulário manual. Fonte: Autor	28
Figura 11 – Tela de tabela de medidas da PoC. Fonte: (SOUZA, 2021)	28
Figura 12 – Página de gráficos. Fonte: Autor	29
Figura 13 – Página de gráficos em dispositivo móvel. Fonte: Autor	30
Figura 14 – Página de gráficos em dispositivo móvel com detalhes. Fonte: Autor	31
Figura 15 – Página de gráficos da PoC. Fonte: (SOUZA, 2021)	32
Figura 16 – Página de observações médicas. Fonte: Autor	33
Figura 17 – Página de observações médicas em dispositivo móvel. Fonte: Autor	34
Figura 18 – Página de observações médicas da PoC. Fonte: (SOUZA, 2021)	35
Figura 19 – Página de alarmes. Fonte: Autor	36
Figura 20 – Página de alarmes em dispositivo móvel. Fonte: Autor	37
Figura 21 – Página de alarmes da PoC. Fonte: (SOUZA, 2021)	38
Figura 22 – Página de perfil de usuário. Fonte: Autor	39
Figura 23 – Página de perfil do usuário em dispositivo móvel. Fonte: Autor	40
Figura 24 – Página de perfil de usuário da PoC. Fonte: (SOUZA, 2021)	41
Figura 25 – Gráfico de medidas para valores bons. Fonte: Autor	43
Figura 26 – Gráfico de medidas para valores quase ideais. Fonte: Autor	43
Figura 27 – Gráfico de medidas para valores não ideais. Fonte: Autor	44
Figura 28 – Gráfico de medidas para valores ruins. Fonte: Autor	45

Lista de tabelas

Tabela 1 – Perguntas e respostas do glicosímetro. Fonte: (SOUZA, 2021)	15
Tabela 2 – Níveis de glicemia para lógica fuzzy.	21

Sumário

1	INTRODUÇÃO	9
2	OBJETIVOS	11
2.1	Objetivos Gerais	11
2.2	Resultados Esperados	11
3	DESENVOLVIMENTO	12
3.1	Revisão	12
3.1.1	Estado da Arte	12
3.1.2	Hardware	14
3.1.3	Software	15
3.2	Análise de requisitos	16
3.3	Desenvolvimento do MVP	18
3.3.1	Back-end	18
3.3.1.1	Lógica Fuzzy	19
3.3.2	Broker e Publish & Subscriber	21
3.3.3	Banco de dados	22
3.3.4	Front-end	25
4	HOMOLOGAÇÃO	42
4.1	Medidas boas	42
4.2	Medidas quase ideais	43
4.3	Medidas não ideais	44
4.4	Medidas ruins	44
5	CONCLUSÕES FINAIS	46
5.1	Trabalhos Futuros	46
	REFERÊNCIAS	48

1 Introdução

O Diabetes Mellitus (DM) é uma doença crônica ocasionada pela falência das células beta pancreática ou pela tolerância aumentada das células do corpo à insulina, levando à hiperglicemia, característica principal de portadores da doença. Existem três tipos principais de DM, o tipo I (DM1), o tipo II (DM2) e o DM gestacional. Em 2020, de acordo com a pesquisa feita pela *International Diabetes Federation* ([INTERNATIONAL . . . , 2021](#)), o número de pessoas portadoras da doença era de 16,8 milhões. A forma mais comum é o DM2, sendo este aproximadamente 90% dos casos registrados de DM, principalmente em adultos acima dos quarenta anos. Este está fisiopatologicamente associado a hábitos alimentares e falta de exercícios físicos no dia a dia, além da obesidade. A hiperglicemia crônica é a principal contribuinte para o desenvolvimento de complicações crônicas como a neuropatia, retinopatia, nefropatia e macro e micro angiopatias.

Visto que o principal agravador do DM é a hiperglicemia, é de extrema importância ter o monitoramento frequente dos níveis glicêmicos pelo paciente diabético ou pré-diabético. Esse monitoramento é feito geralmente por aparelhos chamados de glicosímetros, capazes de medir a glicemia capilar. Um fator importante dos glicosímetros é que sejam de fácil manuseio, pois a medida será feita diversas vezes no dia, além do fato de que o DM pode levar o paciente à perda de visão (retinopatia).

Atualmente a maioria dos glicosímetros comerciais possuem apenas um *display* que exibe o valor medido, fornecendo apenas de forma visual os resultados, dificultando a medição a pessoas com deficiências visuais. Logo, a solução desenvolvida levou em consideração tais pacientes e incorporou fatores sonoros no aparelho. Este que emite sons para dar instruções ao paciente para saber em qual etapa da medição está e qual pergunta deve responder em seguida. Além disso, possui conectividade Wi-Fi, facilitando o envio dos dados para dispositivos como computadores e celulares. ([BORGES; KATO; LEAL, 2019](#))

Com a conexão à rede, o glicosímetro é capaz de enviar os dados medidos para um banco de dados e, com isso, os valores passam a ser exibidos num sistema web de fácil acesso ao paciente. A partir daí o paciente com deficiência visual pode se dispor de várias técnicas e ferramentas para consumir a informação, como um assistente de voz que lê os dados exibidos na tela.

Tendo esses dados na nuvem, é possível utilizá-los para identificar tendências, realizar levantamentos estatísticos e estabelecer relações de dependência entre os níveis glicêmicos e outros fatores relacionados à saúde do usuário. Com isso, é possível também sugerir instruções para que o paciente possa melhorar o seu nível glicêmico conforme os

resultados obtidos. Além disso, como os dados ficam dispostos no sistema web, a família do paciente e seus médicos podem acessar os valores facilmente e com segurança de que não foram manipulados pelo paciente ou por qualquer outro imprevisto.

Após serem coletados, os dados passam por uma análise utilizando ferramentas de inteligência artificial. Neste caso, a ferramenta utilizada é a lógica fuzzy, uma espécie modificada da lógica booleana tradicional, explicada no trabalho *Introduction to Fuzzy Logic* realizado pela *Facta universitatis - series: Electronics and Energetics* em 2005 (MORAGA, 2005), que proporcionará resultados com base no nível glicêmico e outras perguntas binárias as quais o paciente deve responder junto com a medição, como se está em jejum, se pratica exercícios físicos, se passou por estresse mental ou físico e se está utilizando medicação corretamente. Com esses valores em mãos, o sistema consegue prever se os índices glicêmicos tendem a aumentar, estabilizar ou diminuir. Além disso, futuramente será possível sugerir medidas preventivas ao paciente para ajudar no tratamento com base nas tendências calculadas pelo sistema.

2 Objetivos

2.1 Objetivos Gerais

Este trabalho de conclusão de curso tem como objetivo a construção de um Mínimo Produto Viável (MVP — *Minimum Viable Product*) a partir da Prova de Conceito (PoC - *Proof of Concept*), (SOUZA, 2021), trabalho de conclusão de curso feito em 2021. Para isso, será necessário primeiro revisar os requisitos do projeto, como os algoritmos utilizados, a lógica implementada para as medições, os designs de telas, banco de dados, entre outros. Com a ajuda de um profissional da área da saúde serão feitas evoluções na PoC, fazendo as alterações necessárias caso hajam para modernizar o sistema utilizando tecnologias mais recentes e testes de bancada com dados fictícios. A validação da lógica fuzzy será feita a partir de testes com dados fictícios, de uma simulação do glicosímetro feita em código e também com dados inseridos a partir do formulário do sistema. Será necessário mostrar que o mesmo mantém os resultados coerentes com a PoC, mostrando-se ser uma evolução tecnológica daquele sistema. Além de produzir resultados que façam sentido com base nos valores apresentados. Também será importante traçar passos futuros de melhoria para o projeto.

2.2 Resultados Esperados

Abaixo se encontram os resultados esperados do desenvolvimento do projeto:

1. Revisão de requisitos do projeto;
2. Revisão do design da solução, incluindo módulos de hardware e software;
3. Testes de bancada usando dados fictícios;
4. Produção de um relatório em formato de TCC com os resultados obtidos para demonstração da evolução do projeto;

Para atingir estes resultados serão empregadas técnicas de inteligência artificial para a análise dos dados coletados. O objetivo final é demonstrar que o glicosímetro e a lógica de análise de dados se suporta quando comparados a um glicosímetro de mercado e à PoC para que seja possível prever tendências e recomendar ações preventivas ao paciente de acordo com a análise.

3 Desenvolvimento

Para o desenvolvimento do projeto foram criadas etapas a serem seguidas durante um cronograma específico a fim de manter todas as partes interessadas no trabalho informadas do que estaria acontecendo em cada período. Abaixo estão as etapas e o tempo de desenvolvimento de cada uma delas:

1. Revisão [3.1](#)
2. Análise de requisitos [3.2](#)
3. Desenvolvimento do MVP [3.3](#)
4. Homologação [4](#)
5. Correções (etapa realizada em paralelo com as outras)

3.1 Revisão

Para a etapa de revisão foi necessário fazer uma revisão da versão atual e bibliografia da PoC ([SOUZA, 2021](#)) a fim de entender em qual etapa do desenvolvimento o projeto está, quais foram os artigos utilizados como referência no trabalho anterior e ver quais teriam que ser aprofundados para o desenvolvimento do MVP.

3.1.1 Estado da Arte

Um dos primeiros passos a serem realizados foi a pesquisa por trabalhos relacionados para entender o que já foi ou está sendo feito nos últimos anos. Existem diversos artigos científicos que falam sobre o tratamento de pacientes aliados à nuvem e IoT e, mais recentemente, técnicas de inteligência artificial.

Como citado anteriormente, o primeiro trabalho a ser revisado foi a própria PoC. Foi necessário entender quais eram as tecnologias utilizadas, os motivos por trás das escolhas e o que poderia ser mudado contando com a tecnologia atual que avançou exponencialmente desde a realização do trabalho. Como esse projeto é uma continuação e evolução do trabalho ([SOUZA, 2021](#)), muitas coisas podem se manter até certo ponto. A arquitetura proposta pela PoC funciona e faz sentido, o modelo de *broker*, *publish* e *subscriber* também será mantido, e algumas outras funcionalidades que serão discutidas mais adiante.

O trabalho apresentou bons resultados como PoC e apontou evoluções a serem seguidas. Na conclusão, o autor cita algumas delas como:

1. Considerar mais fatores além dos propostos no trabalho;
2. Adicionar mais ferramentas na plataforma web;

Esses que foram levados em consideração para este projeto para criar uma versão evoluída da PoC, trazendo para uma realidade mais atual de tecnologias e de análises.

Outro trabalho que foi importante ser revisado foi o projeto do glicosímetro IoT (BORGES; KATO; LEAL, 2019). Nessa papente é explicado como o glicosímetro foi projetado a partir das especificações de professores dos departamentos da computação e medicina da UFSCar. O trabalho discute muito sobre a questão do DM ter correlação com a retinopatia diabética, essa que pode levar à cegueira. Logo, um dos focos maiores do trabalho é a parte auditiva do glicosímetro, que permite com que pacientes afetados pela retinopatia diabética ou qualquer deficiência visual possam realizar as medidas e utilizar o equipamento com todas suas funções.

O trabalho também entra em detalhes sobre como o Arduino realiza as medidas e como ele foi construído, desde os testes iniciais com diversos modelos diferentes de Arduinos até a construção de um protótipo final. O mesmo se dispõe de diversos gráficos e projetos elétricos para explicar o funcionamento completo do glicosímetro. O que não havia sido feito ainda foi o sistema capaz de armazenar os dados e exibi-los aos usuários, tirando proveito do módulo de conectividade Wi-Fi integrado no Arduino para enviar esses dados à um servidor e fazer análises em cima deles.

O trabalho “An IoT-Based Glucose Monitoring Algorithm to Prevent Diabetes Complications” (VALENZUELA et al., 2020) descreve uma espécie de sistema muito parecida com o que este trabalho planeja fazer. Lá, os autores falam sobre utilizar um dispositivo IoT como glicosímetro, aplicar um algoritmo capaz de prever as tendências e analisar como a glicose do paciente se comportará no futuro, assim como também ter um sistema capaz de exibir todos esses dados. A maior diferença entre os trabalhos é a tecnologia utilizada e o fato de que o glicosímetro ainda era teórico no caso da pesquisa feita pelos autores do artigo. Além disso, o algoritmo desenvolvido não se tratava da mesma lógica fuzzy, apenas uma série com base nas médias de valores anteriores. Também não possui um *back-end* dedicado à lógica do sistema, tendo apenas uma *raspberry pi* como servidor, provendo os dados para uma tela web exibi-los. Porém, existem diversas semelhanças entre os dois trabalhos e existem características que podem ser aproveitadas como a ideia de ter notificações para o paciente e médico ou familiares.

Em paralelo também houve a necessidade de fazer a revisão do estado da arte em glicosímetros IoT. Nessa revisão foi pesquisado sobre glicosímetros com a medição através da extração de sangue do paciente que funcionam a partir da reação da glicose do sangue capilar com a fita presente no dispositivo assim como descrito na apresentação "TECNOVIGILÂNCIA – Estudo de Caso - GLICOSÍMETRO" publicado pela Secretaria

de Saúde do Paraná (SANTOS, 2013). Quanto a esse tipo de glicosímetro, não foram encontrados modelos com as mesmas capacidades sonoras e de conexão Wi-Fi. O que foi encontrado foram dispositivos conhecidos como CGM (Continuous Glucose Monitoring — Monitoração Contínua da Glicose). Conforme o artigo “*Fundamentals of glucose monitoring sensors.*” de Chaithanya Ratakonda, M.D. (RATAKONDA, 2021) e a matéria publicada na revista virtual *Cleveland Clinic* (CONTINUOUS...), estes dispositivos ficam acoplados ao paciente e medem a glicose com base no tecido subcutâneo num período de 7 a 14 dias sem ter que ser removidos, o funcionamento dos. Um exemplo de CGM seria o Freestyle Libre da empresa Abbot. Este que possui um artigo no qual cita as diferenças entre o monitoramento contínuo e a medição no sangue através da picada de agulha. O funcionamento deste CGM é explicitado na publicação “Qual a diferença entre monitoramento contínuo de glicose e ponta de dedo?”, no blog da empresa “Blog — Freestyle Libre” (QUAL..., 2023). Existem benefícios de se utilizar o CGM ao invés do glicosímetro comum, é mais prático por não ter que colher sangue a cada medida, os dados são mais confiáveis por ser uma medida contínua ao invés de pontual, e a maioria dos CGM’s do mercado possuem uma UI (User Interface — Interface de Usuário) específica que faz a coleta das medidas e upload de dados através de uma conexão bluetooth. Porém, o custo de um CGM é extremamente elevado, chegando a custar até USD 1.000,00 por mês, assim como descrito no artigo “*Continuous Glucose Monitoring (CGMs) vs. Blood Glucose Monitoring (Finger Stick): What’s the Difference?*” publicado no blog *Keto-Mojo* especializado em hábitos e alimentação saudável (NELSON, 2021), o que é fora dos padrões para a grande maioria da população, lembrando que o salário mínimo no Brasil em 2023 é de R\$1.320,00 segundo a publicação mais recente sobre o salário mínimo no portal da câmara legislativa (dois de maio de 2023) (NOTÍCIAS, 2023). Portanto, o projeto em questão proporcionaria a parte de UI e conexão com a rede que os glicosímetros comuns não possuem hoje, por um custo mais acessível.

3.1.2 Hardware

Para a revisão do *hardware* não há a necessidade de mudar nada. O glicosímetro é um microcontrolador de módulo ESP32 que fará a coleta dos dados e o envio para o sistema. É possível realizar essa tarefa, pois esse modelo de microcontrolador possui em sua composição uma placa *wireless* capaz de se conectar a uma rede Wi-Fi. A partir dessa conexão é possível programar instruções na ESP32 para que, assim que uma medida seja realizada, o Arduino envie os dados para um *broker* MQTT¹ através de um *publish* de mensagem num tópico especificado. A programação da placa foi realizada no trabalho anterior a este, (SOUZA, 2021), em C++ e com o uso de bibliotecas já prontas. Além disso, todo o sistema para medição da glicose e a construção do hardware foi realizado em 2019 e está evidenciado na patente do dispositivo (BORGES; KATO; LEAL, 2019).

¹ Sistema que lê, filtra, e envia mensagens conforme o tópico e clientes registrados.

No momento da medição da glicose, o glicosímetro fará uma série de perguntas que devem ser respondidas com sim ou não pelo paciente. Cada pergunta será feita em voz alta pelo glicosímetro para que pacientes com deficiências visuais realizem a medição sem problemas. As perguntas se encontram na tabela 1). Estas que são as mesmas perguntas realizadas no trabalho de PoC de 2021.

N.º	Pergunta	Resposta
1	Está em Jejum?	Sim ou Não
2	Pratica exercícios físicos?	Sim ou Não
3	Passou por estresse físico ou mental recentemente?	Sim ou Não
4	Está utilizando a medicação corretamente?	Sim ou Não

Tabela 1 – Perguntas e respostas do glicosímetro. Fonte: (SOUZA, 2021)

O paciente deve responder essas perguntas com o pressionar de um botão no glicosímetro para a opção “Sim” e não pressionar o botão para a opção “Não”.

Após a coleta de dados, o glicosímetro faz um pré-processamento destes, agrupando-os num modelo JSON e enviá-os como uma mensagem a um *broker* MQTT. O *back-end* do sistema estará conectado a esse mesmo *broker* como um Subscriber e assim que uma mensagem é publicada, o *back-end* também irá recebê-la. A partir disso iniciá-se a criação de uma medida no banco de dados. Essa etapa é explicada com mais detalhes na subseção de Lógica Fuzzy 3.3.1.1.

3.1.3 Software

Ao realizar a revisão da frente de software foi concluído que o mesmo teria que ser re-escrito, pois a tecnologia utilizada foi a linguagem de programação Python, com o *framework Flask*, uma linguagem que não é muito comum no desenvolvimento de sistemas web atualmente. No artigo “*Python VS JavaScript — What are the Key Differences Between The Two Popular Programming Languages?*” a autora cita as principais diferenças entre o python e o javascript para a criação de aplicações web e conclui que o javascript é mais utilizado quando se trata de aplicações web e mobile. Com isso em mente, para o MVP, será desenvolvido um sistema PWA² (Progressive Web App). O maior motivo pelo direcionamento do projeto para o desenvolvimento de um PWA foi a permissão de execução de um sistema web como um aplicativo em aparelhos celulares *smart-phones*. Dessa forma os usuários terão acesso aos dados através do celular, tablet ou computador. Além disso, pelo fato do PWA ser executado nativamente em navegadores, o versionamento do aplicativo é trivial, uma vez que, ao fazer o *deploy* do sistema, todos os aplicativos irão

² Sistema que simula um aplicativo em dispositivos inteligentes rodando-o nativamente no navegador do aparelho.

automaticamente utilizar a versão mais recente sem ter que baixar e instalar uma nova atualização.

Quanto aos serviços do *back-end*, a grande maioria terá de ser re-escrita também devido ao fato de que uma mudança para um *back-end* em javascript faz sentido ao utilizar o *front-end* com a mesma linguagem. Logo, para o *back-end* será utilizado NodeJS (NODEJS, 2023) que é um framework que possibilita a execução de *scripts*, e o Typescript (TYPESCRIPT, 2023) para auxiliar no desenvolvimento.

O trabalho anterior também não tinha um banco de dados persistente. Estava sendo utilizado um banco, que era inicializado a cada execução do programa, incapaz de guardar informações depois do sistema ser finalizado. Logo, uma solução encontrada foi a criação de um banco de dados relacional. Este que fará a função de guardar todas as medidas dos usuários assim como seus dados e observações médicas. Segundo o artigo “Qual é a diferença entre bancos de dados relacionais e não relacionais?” publicado no blog da *Amazon Web Services* esse modelo de banco de dados é mais organizado e recomendado para projetos nos quais os dados são previsíveis em termos de tamanho, estrutura e frequência de acesso. Além disso, é fácil de criar tabelas e administrar as colunas e tipos de dados. Também existem diversas bibliotecas para fazer a comunicação dos aplicativos com o banco dada as tecnologias que serão utilizadas.

3.2 Análise de requisitos

Na etapa de análise de requisitos foi necessário entender o que o sistema deveria possuir de funcionalidades e serviços para que fosse considerando um MVP. Para isso, os seguintes diagramas foram desenhados, demonstrando o relacionamento entre as entidades mais importantes do sistema, o glicosímetro, banco de dados, *back-end* e *front-end*.

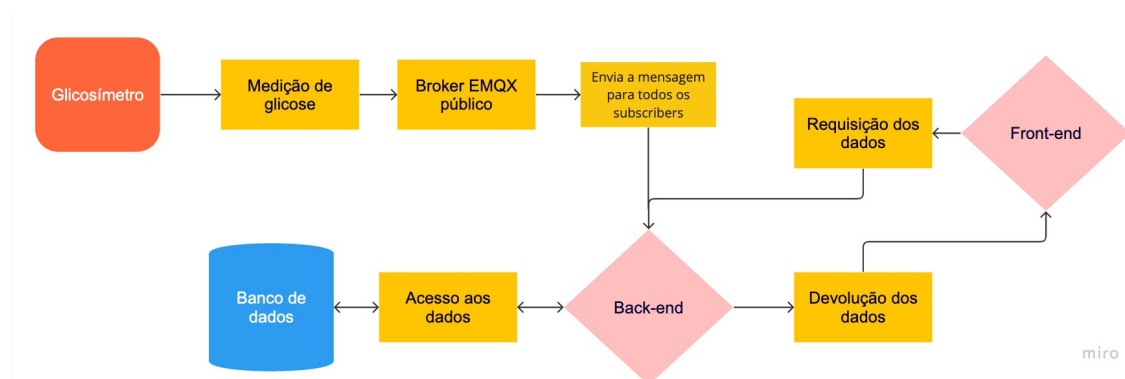


Figura 1 – Fluxo do sistema. Fonte: Autor

No fluxograma da figura 1 existem dois pontos de partida. O primeiro sendo o glicosímetro e a medição de glicose através dele. Trabalhou-se com um modelo de *Publish-Subscriber* utilizando o *broker* EMQX público, cujas informações podem ser encontradas

na documentação do mesmo (EMQX, 2023), para fazer o intermédio entre os dois. A partir da mensagem publicada pelo *broker*, o *back-end* recebe os dados por ser um *subscriber* no tópico criado (*userGlucose*) e faz a criação da medida no banco dados. Esses dados são utilizados pelo segundo ponto de partida, trabalhando num padrão de código MVC (Model View Controller), o qual está descrito em detalhes na publicação de 2020 “O que é MVC?” do blog Treinaweb (O..., 2020) no qual o *front-end* (View) se comunica com o *back-end* (Controller) através de requisições para uma API *restful* que é uma interface de programação de aplicações (API ou API web) em conformidade com as restrições do estilo de arquitetura REST assim como explicado no texto publicado em 2023 no blog da “Red Hat”, uma das maiores empresas de desenvolvimento de sistema operacional Linux (API..., 2023). O *back-end* acessa o banco de dados (Model) e retorna-os para o *front-end*. É importante frisar que o *front-end* não tem contato direto com o banco de dados e, sempre que precisa acessar algum dado, deve passar pelo *back-end*. Isso é feito para o sistema ter uma manutenção facilitada.

Pensando nas funcionalidades que o sistema deveria ter dentro do escopo de MVP, as soluções pensadas foram as seguintes:

1. Realizar medidas manuais;
2. Visualizar as medidas realizadas (tabela e gráfico);
3. Passar ao usuário quais sons o glicosímetro emite;
4. Comunicação entre o paciente e seu médico;

Para essas funcionalidades, decidiu-se utilizar as mesmas telas que o projeto de 2021 já tinha, com um design melhorado e a adição de um formulário manual para a inserção de medidas. Este que seria um ponto de destaque do sistema uma vez que a partir deste será possível realizar medidas com outros glicosímetros de mercado e inserir os valores na plataforma para que a inferência de glicemia seja calculada também.

As telas, portanto, são:

1. Tabela de medidas com formulário manual para inserção de medida nova;
2. Gráfico de medidas;
3. Observações médicas;
4. Alarmes e sons que o glicosímetro emite;
5. Perfil do usuário.

É importante citar que as telas de tabela de medidas, observações médicas, gráficos e alarmes ficarão disponíveis para acesso do doutor e família que estejam cadastrados com o mesmo usuário de paciente no banco de dados.

Com essas diretrizes acredita-se que o usuário poderia utilizar a plataforma sem nenhum impasse e, a partir do uso contínuo, será possível observar quais módulos ainda precisam ser inseridos e quais podem ser removidos, além do que está confuso para o usuário e precisaria ser modificado.

No *back-end* foram implementados serviços que também já estavam na PoC de 2021 como a lógica fuzzy para previsão da glicemia do paciente com base nos dados coletados pelo glicosímetro, e o *handler* para a requisição que o glicosímetro fará para inserção de dados no banco de dados. E há serviços novos que foram desenvolvidos como retornar os dados do usuário, medidas e observações medicas através de uma rota de API, inserção das medidas manuais no banco de dados, e a simulação do glicosímetro em código para validação do sistema e regras fuzzy.

Com isso é possível ter um entendimento melhor sobre o sistema, suas funcionalidades e requisitos, e saber exatamente o que será necessário para desenvolver o MVP.

3.3 Desenvolvimento do MVP

O desenvolvimento do MVP consistiu em 4 peças importantes, 2 repositórios de código, um para o *front-end* e um para o *back-end*, um banco de dados e o glicosímetro. Essas 3 primeiras peças foram desenvolvidas em paralelo, a fim de evitar problemas de *deadlock* quando uma das peças precisava de dados e vice-versa, enquanto o glicosímetro não precisou de alterações.

3.3.1 Back-end

A primeira peça a ser desenvolvida foi o *back-end* que será a parte central do projeto, encarregado de fazer a comunicação com o banco de dados, o *front-end*, o glicosímetro e também de todas as regras de negócio e lógicas de inteligência artificial. A linguagem escolhida foi o javascript pelo fato de ser possível desenvolver tanto o *back-end* quanto o *front-end* utilizando-o, facilitando a comunicação entre os dois. Para isso, utilizou-se o NodeJS que é um ambiente de runtime de javascript que possibilita a execução da aplicação localmente na máquina ao invés de na rede assim como explicado na própria documentação do NodeJS ([NODEJS, 2023](#)). Dessa forma não há necessidade de fazer um *deploy* da aplicação para testá-la. A outra tecnologia utilizada foi o Typescript que nada mais é que uma variação do javascript que possibilita a tipagem do código descrito também na documentação do Typescript ([TYPESCRIPT, 2023](#)), a fim de facilitar o desenvolvimento, minimizar a quantidade de defeitos no código, e testes. Além disso, também foi utilizada a

biblioteca ExpressJS que é altamente recomendada para o desenvolvimento de aplicações web de acordo com a documentação da mesma ([EXPRESSJS, 2023](#)), e ficará encarregada de gerenciar as rotas, requisições e respostas da API.

3.3.1.1 Lógica Fuzzy

Conforme o trabalho “Fuzzy Sets” ([ZADEH, 1965](#)), a lógica fuzzy se trata de uma espécie de lógica booleana tradicional com algumas diferenças. A mais importante é que os valores não são necessariamente verdadeiros ou falsos, eles podem assumir qualquer valor entre 0 e 1. O fato de não ser apenas um booleano permite que essa lógica tenha aplicações na área da inteligência artificial, podendo gerar tendências de acordo com os dados.

Uma das evoluções da lógica fuzzy para a lógica booleana é a função de pertinência fuzzy. Essas que vão calcular qual é o grau de verdade de uma declaração. Por exemplo, se na lógica tradicional tem-se a relação de que x é G logo y é H , na lógica fuzzy existe uma função de pertinência $F(G \rightarrow H)(x, y)$ que mede o grau de verdade da relação de implicação entre x e y .

Cada uma dessas funções de pertinência vão dar valores a variáveis, que juntos formam um conjunto fuzzy, este que é uma classe de objetos com graus de pertinência. Cada entrada num conjunto fuzzy passou por uma função de pertinência que lhe atribuiu um valor de 0 a 1 para estar dentro do conjunto ou não. Os graus de pertinência são referentes a uma variável linguística, ou seja, cada valor que passa pela função de pertinência será analisado com base nessa variável linguística a partir de regras pré-definidas. A figura 2 abaixo ilustra como funciona um conjunto fuzzy.

Nesse caso, a variável linguística foi “Jovem”, e pode-se ver que o pico de valor está na idade 27 e a partir disso os valores vão decrescendo, estando menos dentro da categoria “Jovem”. No eixo Y estão os graus de pertinência, demonstrando qual é o valor de cada idade para a variável linguística escolhida.

Existem diversas formas de definir as funções de pertinência para solucionar diferentes tipos de problemas. A mais comum é a mesma utilizada na figura 2, uma função triangular. Pode-se inclusive ter combinações de funções para atingir objetivos mais complexos.

Outra parte de extrema importância para a lógica fuzzy são as regras. Estas que são criadas de acordo com o que se quer extrair da análise e funcionam como regras lógicas comuns. Para o projeto realizado o problema proposto é o de analisar a tendência da glicose de um paciente a partir do valor medido, se o paciente pratica exercícios físicos, passou por estresse mental ou físico, está utilizando os medicamentos corretamente e se está de jejum. Com essas variáveis deve-se construir todas as regras para contemplar os

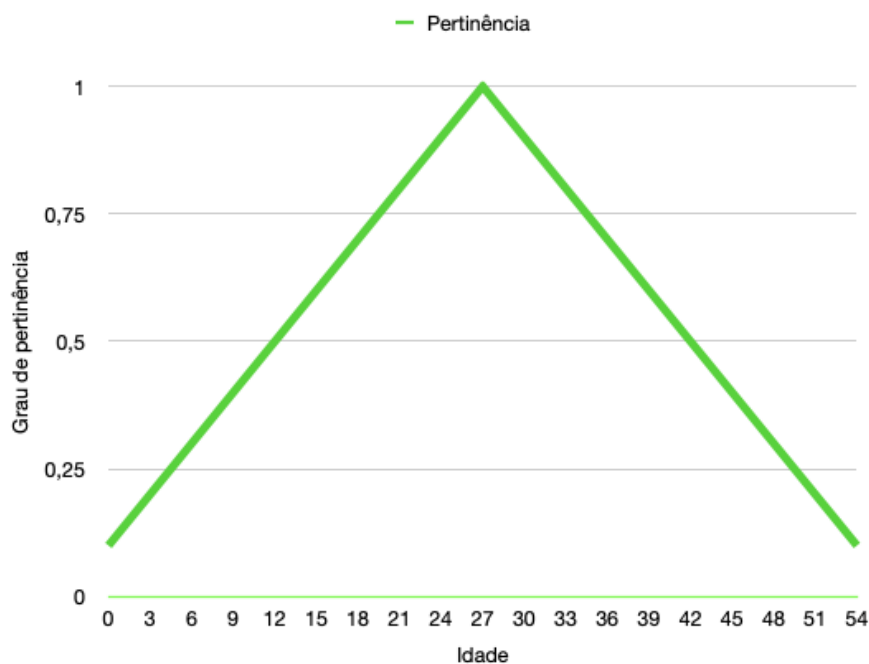


Figura 2 – Conjunto fuzzy com variável linguística “Jovem”. Fonte: Autor

casos. A estruturação da regra é:

SE glicose = alta E exercício = sempre ENTÃO tendencia = diminuir

Devido à quantidade de variáveis presentes no sistema, mais de 200 regras foram criadas já no trabalho da PoC, estas que, para este trabalho, foram revisadas e modificadas com o auxílio de duas médicas, Julia Diniz Fabbri, CRM — SP 239.219 e Marília Bitelli Dressler, CRM — SP 233.618. Com o *feedback* de ambas foi possível entender quais variáveis contribuem mais ou menos para a tendência da glicemia. Dentre elas pode-se citar o jejum que influencia no que é considerado um valor normal, pré-diabético ou diabético. Também deram destaque para a idade do paciente, exercício físico e medicação como sendo os fatores mais importantes para o controle e previsão da glicose.

Na lógica fuzzy as variáveis booleanas como exercício, medicação, estresse e jejum, recebem um grau de pertinência, sendo inseridos nos conjuntos que podem ter um valor de sempre, predominantemente sim, às vezes, predominantemente não, e nunca. Para calcular a tendência são sempre usadas as últimas 20 medidas registradas na conta do usuário, sejam elas realizadas manualmente pelo formulário ou pelo glicosímetro. Para a variável da glicose foi utilizada a tabela 2 para definir se a glicose está alta, baixa ou normal. Esses valores foram definidos a partir de artigos como o da Dra Myrna Perez Campagnoli “GLICOSE ALTA, BAIXA E NORMAL: ENTENDA O QUE SIGNIFICA” (CAMPAGNOLI, 2023), publicado no site do laboratório Delboni, uma das maiores redes de laboratório do país. Na publicação “Mean fasting blood glucose” da World Health Organization (Organização Mundial de Saúde — OMS), (MEAN... , 2023), e pela ajuda

de ambas médicas.

Valor (mg/dL)	Classificação
70 <	Baixa
70 à 125	Normal
> 125	Alta

Tabela 2 – Níveis de glicemia para lógica fuzzy.

É possível descrever o funcionamento da lógica fuzzy através de um sistema como na figura 3. Nesse sistema as entradas são valores comuns que alimentam o sistema. A partir disso, passam pelo processo de fuzzyficação no qual a função de pertinência atribuirá um valor para a entrada fuzzyficação conforme as regras pre-estabelecidas. Depois disso, os valores são defuzzyficados e por fim a saída volta como um valor comum para ser inserido no conjunto fuzzy.

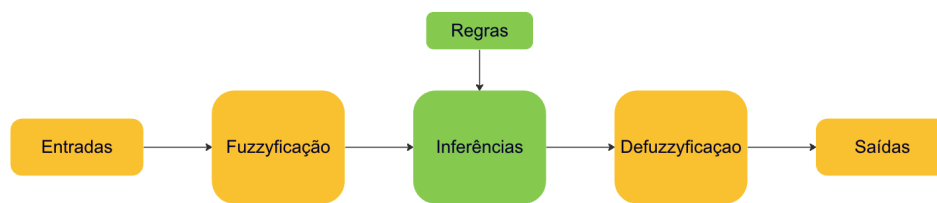


Figura 3 – Sistema fuzzy. Fonte: Autor

Logo, toda medida encaminhada para o *back-end* passa primeiro pelo tratamento da lógica fuzzy para que seja calculada a tendência no momento daquela medida. Então, o sistema guarda os valores e a tendência no banco de dados para serem visualizados no *front-end* imediatamente após o cálculo da mesma.

3.3.2 Broker e Publish & Subscriber

A comunicação com o glicosímetro também é feita no *back-end* e a metodologia escolhida para isso foi o de *Publish & Subscriber* com um *broker*. Como o glicosímetro utilizado possui conexão Wi-Fi, é possível fazê-lo enviar dados através da rede e uma das melhores formas de fazer isso é por uma publicação num *broker* MQTT por conta da simplicidade de se criar um canal de comunicação e a acessibilidade dos *brokers*. O código para fazer o glicosímetro enviar os dados no formato JSON foi realizado no trabalho anterior a este (SOUZA, 2021), em C++.

Na fase de homologação do MVP foi-se criada uma simulação do glicosímetro para a criação de dados para os testes de bancada a fim de validar o funcionamento da comunicação entre o glicosímetro e o banco de dados, assim como o funcionamento da lógica fuzzy. Utilizou-se um *broker*-público da EMQX, cujo funcionamento pode ser

encontrado na documentação do mesmo (EMQX, 2023), que fica contido num *container* Docker (VINCENZI, 2020), escutando quando algum *publisher* ou *subscriber* se conectam ao *broker* e publicam alguma mensagem. Para o código no *backend* foi utilizada a biblioteca *mqtt* (MQTT, 2023) para NodeJS que possui suporte para Typescript também. Com ela é possível estabelecer a conexão com o *broker*, enviar e receber mensagens. Além disso, a biblioteca foi utilizada para a criação do glicosímetro simulado, para que seja possível fazer a conexão entre a simulação e o *broker* EMQX.

O dispositivo simulado se conecta ao *broker* e publica medidas de glicose aleatórias no tópico *userGlucose*. Essas que são entre 70 e 130 para a glicose, e sim ou não para as outras perguntas realizadas pelo glicosímetro. Isso acontece de dois em dois minutos. Enquanto isso, o *back-end* se conecta como *subscriber* e espera a mensagem ser enviada que corresponda ao tópico *userGlucose*. Ao receber uma mensagem, o sistema aplicará a lógica fuzzy como descrita na subseção anterior 3.3.1.1 e salvará os dados e a tendência calculada no banco de dados, ficando prontos para serem exibidos pelo *front-end*

A figura 4 mostra o funcionamento do modelo de *Publish & Subscriber* com o *broker* utilizado no desenvolvimento do projeto.



Figura 4 – Sistema *Publish & Subscriber* com *broker*. Fonte: Autor

3.3.3 Banco de dados

Em seguida foi criado o banco de dados a partir, novamente, do uso da plataforma Docker e utilizando o trabalho “Prática de DevOps com Docker” (VINCENZI, 2020) para entender o funcionamento da mesma. Com isso foi possível criar uma imagem de um banco de dados PostgreSQL, rodada em um *container*. A partir dessa imagem, qualquer pessoa que for trabalhar no projeto poderá baixar a imagem e executá-la dentro de um *container* docker local, sem a necessidade de criar um banco de dados local do zero. A escolha do PostgreSQL para o banco foi feita por conta da facilidade de trabalhar com bancos

relacionais, uma vez que a organização fica mais clara e é possível utilizar *frameworks* no *back-end* para facilitar a comunicação com o banco.

Com o banco de dados criado, a próxima etapa seria popular esse banco com as tabelas que serão utilizadas no desenvolvimento do projeto. Para isso foi utilizada a biblioteca TypeORM cuja documentação da mesma (TYPEORM, 2023) fala que permite a comunicação entre o *back-end* e o banco de dados para a criação e edição de tabelas, assim como a busca e inserção de dados. Foram criadas então as tabelas de usuários e medidas presentes nas figuras 5, 6 e 7).












Column Name	#	Data type
 id	1	serial4
 firstName	2	varchar
 lastName	3	varchar
 email	4	varchar
 CPF	5	varchar
 password	6	varchar
 role	7	int4
 gender	8	int4
 birthday	9	varchar
 created_at	10	timestamp
 updated_at	11	timestamp

Figura 5 – Tabela de usuários. Fonte: Autor








Column Name	#	Data type
 id	1	serial4
 value	2	int4
<input checked="" type="checkbox"/> fasting	3	bool
<input checked="" type="checkbox"/> exercise	4	bool
<input checked="" type="checkbox"/> stress	5	bool
<input checked="" type="checkbox"/> medication	6	bool
 date	7	timestamp
 created_at	8	timestamp
 updated_at	9	timestamp
 userId	10	int4
 trend	13	int4

Figura 6 – Tabela de medidas. Fonte: Autor







Column Name	#	Data type
 id	1	serial4
 observation	2	varchar
 created_at	3	timestamp
 updated_at	4	timestamp
 doctorId	5	int4
 patientId	6	int4

Figura 7 – Tabela de observações. Fonte: Autor

É importante citar que na tabela de medidas presente na figura 6 a coluna **userId** faz referência à tabela de usuários numa relação vários para um, podendo ter várias medidas para um único usuário. Assim como na tabela de observações da figura 7, a mesma relação existe para o ID do médico e do paciente, uma vez que um médico, para um único usuário, pode ter várias observações de acordo com o período do tratamento do paciente.

As tabelas, por sua vez, serão populadas com dados em 3 instantes diferentes:

1. Criação de um novo usuário;
2. Medida feita pelo glicosímetro;
3. Medida manual a partir do formulário do *front-end*;
4. Criação de observação de médico para paciente;

Logo, o *back-end* possui métodos específicos para cada um desses instantes e também métodos para retornar os dados para o *front-end*, todos facilitados pela biblioteca TypeORM.

3.3.4 Front-end

Quanto ao *front-end*, foram criadas telas para a exibição dos dados recebidos através da API, assim como um formulário para o envio de medidas diretamente pelo usuário. O projeto em *front-end* foi desenvolvido em Typescript e Next.js, um framework criado a partir do React. A escolha do Next.js foi feita a partir de um de seus maiores diferenciais do React puro que é o controle de rotas através dos diretórios do projeto explicada na documentação do framework (NEXT..., 2023). Essa facilidade se dispõe na forma de quando um diretório é criado, uma rota com o mesmo nome desse diretório também é criada e acessível, exibindo o conteúdo do arquivo dentro do diretório. Essa funcionalidade ajuda no desenvolvimento, podendo não se preocupar com criar um arquivo de rotas e especificar cada uma para cada componente do projeto.

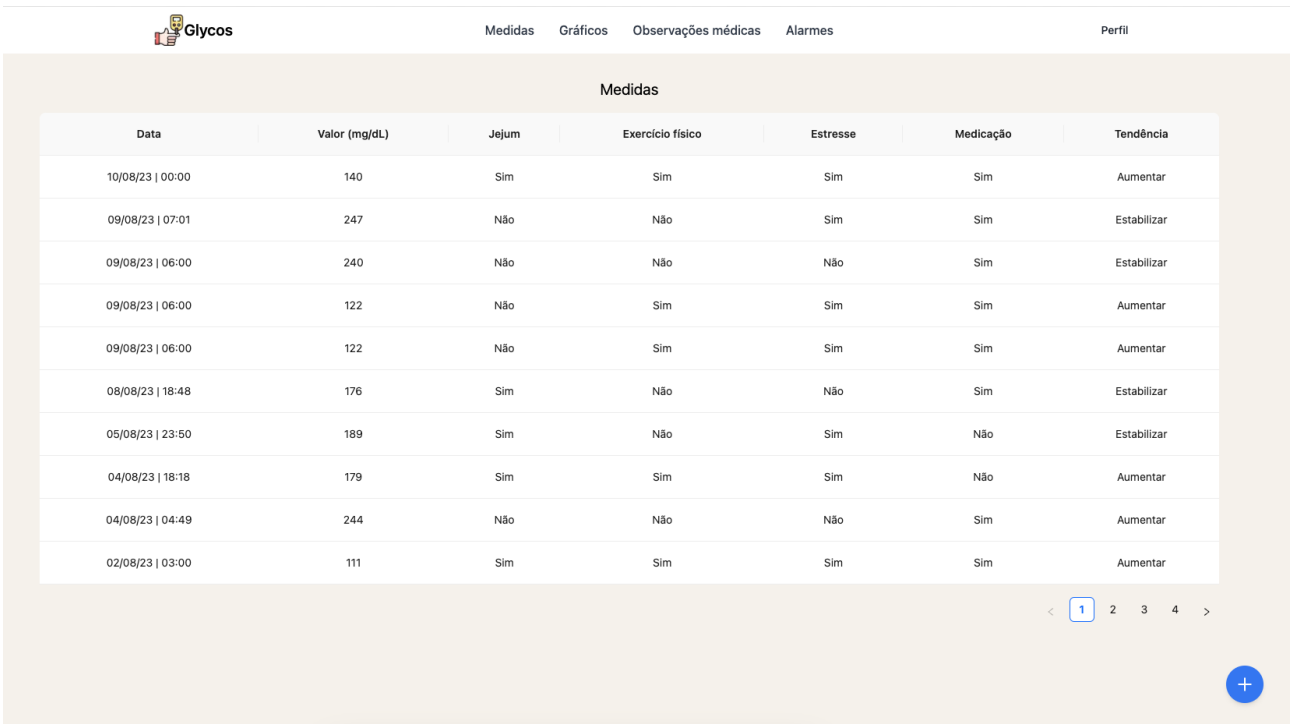
Além disso, o Next.js foi escolhido pela facilidade em se criar o PWA a partir de uma biblioteca chamada next-pwa (NEXT-PWA, 2023), na qual a criação do mesmo se resume a poucas linhas de código e a criação de um arquivo importante que é o “manifest.js”. Esse arquivo especifica como o aplicativo irá se comportar, qual imagem deve usar de capa e se pode ser instalado em dispositivos móveis. Segundo o texto de Donald Bickel, publicado no blog da Mercury Works, “How Does a PWA Work? (and Other Questions You Were Afraid to Ask)” (BICKEL, 2022), o PWA roda nativamente no navegador do dispositivo, essencialmente simulando um aplicativo mobile. Além disso, o PWA possui vantagens como a não atualização de versões, já que o sistema pode ser atualizado do lado do desenvolvedor e o usuário não terá que instalar uma versão nova. Outra ferramenta importante para o desenvolvimento do PWA foi a biblioteca Tailwindcss (TAILWINDCSS,

2023), que ajuda na estilização dos componentes de forma mais rápida por conter classes nativas como o *display flex*. Isso auxilia na criação de telas responsivas, capaz de se adaptar a diferentes tamanhos de *display* como de celulares e notebooks.

O último motivo pela escolha do Next.js é o fato de ser feito pela Vercel, uma empresa de serviços em nuvem, na qual é possível fazer o *deploy* de projetos *front-end* gratuitamente. Por ser um framework criado pela própria plataforma, a integração entre o projeto no Github e a Vercel é extremamente facilitado, fazendo o *build* e *deploy* do projeto toda vez que alguma entrada nova é feita na árvore principal do mesmo.

Com as escolhas de tecnologias feitas, o projeto foi iniciado com base nas telas propostas inicialmente na seção de análise de requisitos 3.2.

A primeira tela de tabela de medidas age também como a página principal do aplicativo, tendo em forma de tabela todas as medidas já realizadas pelo usuário em ordem decrescente. Pode-se ver essa tela na figura 8. Nessa tela também há um botão para abrir o formulário manual para que o usuário possa registrar medidas que fez em horários diferentes sem conexão à internet na hora da medida, ou com aparelhos glicosímetros diferentes. Esse formulário se encontra na figura 10. A tela de tabela da PoC se encontra na figura 11.

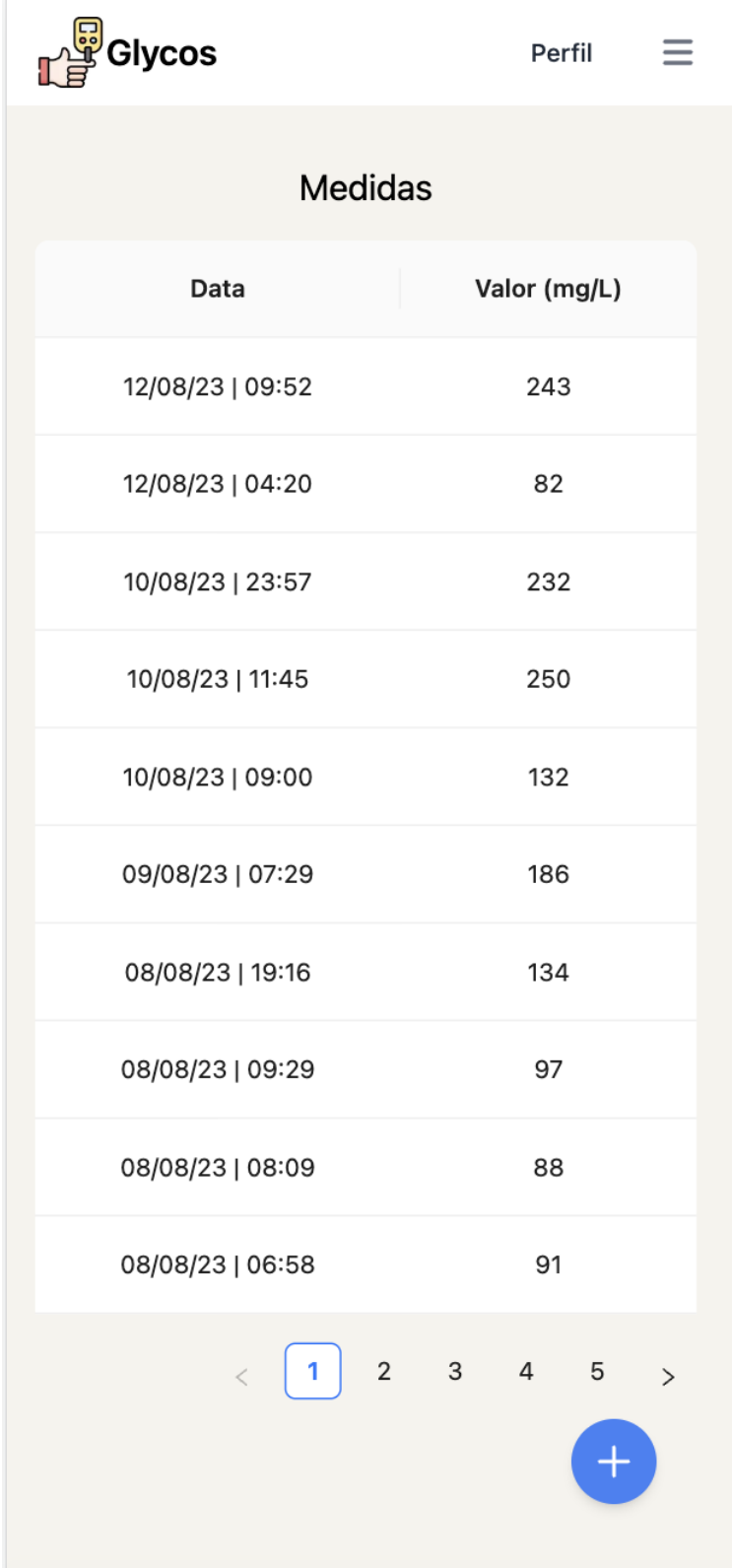


The screenshot shows the Glycos application interface. At the top, there is a navigation bar with the Glycos logo and menu items: Medidas, Gráficos, Observações médicas, Alarmes, and Perfil. The main content area is titled 'Medidas' and contains a table with the following data:

Data	Valor (mg/dL)	Jejum	Exercício físico	Estresse	Medicação	Tendência
10/08/23 00:00	140	Sim	Sim	Sim	Sim	Aumentar
09/08/23 07:01	247	Não	Não	Sim	Sim	Estabilizar
09/08/23 06:00	240	Não	Não	Não	Sim	Estabilizar
09/08/23 06:00	122	Não	Sim	Sim	Sim	Aumentar
09/08/23 06:00	122	Não	Sim	Sim	Sim	Aumentar
08/08/23 18:48	176	Sim	Não	Não	Sim	Estabilizar
05/08/23 23:50	189	Sim	Não	Sim	Não	Estabilizar
04/08/23 18:18	179	Sim	Sim	Sim	Não	Aumentar
04/08/23 04:49	244	Não	Não	Não	Sim	Aumentar
02/08/23 03:00	111	Sim	Sim	Sim	Sim	Aumentar

At the bottom right of the table, there is a pagination control showing '< 1 2 3 4 >' and a blue circular button with a white plus sign.

Figura 8 – Tela de tabela de medidas. Fonte: Autor



The screenshot shows the 'Glycos' mobile application interface. At the top left is the app logo, a thumbs-up icon holding a glucose meter, with the text 'Glycos'. To the right of the logo are the words 'Perfil' and a hamburger menu icon. The main content area is titled 'Medidas' and contains a table with two columns: 'Data' and 'Valor (mg/L)'. The table lists 11 measurements with their respective dates and times and values. At the bottom of the screen, there is a pagination indicator with numbers 1 through 5, where '1' is highlighted in a blue box. To the right of the pagination is a large blue circular button with a white plus sign, used for adding new measurements.

Data	Valor (mg/L)
12/08/23 09:52	243
12/08/23 04:20	82
10/08/23 23:57	232
10/08/23 11:45	250
10/08/23 09:00	132
09/08/23 07:29	186
08/08/23 19:16	134
08/08/23 09:29	97
08/08/23 08:09	88
08/08/23 06:58	91

Figura 9 – Tela de tabela de medidas em dispositivo móvel. Fonte: Autor

Figura 10 – Formulário manual. Fonte: Autor

#	Valor de Glicose	Data	Hora	Jejum	Exercícios	Estresse	Medicação
32	103.52	01/02/2020	11:03	Sim	Não	Sim	Sim
31	103.59	31/01/2020	11:03	Sim	Sim	Não	Sim
30	102.86	30/01/2020	11:03	Não	Não	Não	Não
29	102.24	29/01/2020	11:03	Não	Sim	Não	Sim
28	102.09	28/01/2020	11:03	Não	Não	Sim	Sim
27	101.81	27/01/2020	11:03	Sim	Não	Sim	Não
26	100.01	26/01/2020	11:03	Não	Não	Sim	Não
25	99.07	25/01/2020	11:03	Sim	Sim	Não	Não

Figura 11 – Tela de tabela de medidas da PoC. Fonte: (SOUZA, 2021)

A tela de gráficos exibe as medidas na forma de um gráfico de linha, podendo escolher o período de visualização com base nas datas das medidas. Ao passar o cursor ou clicar em cima de alguma medida no gráfico, é possível visualizar os detalhes da medida, com as informações de jejum, exercício, estresse, medicação e a tendência calculada a partir das últimas 20 medidas. A página se encontra na figura 12 A tela da PoC se encontra na figura 15.



Figura 12 – Página de gráficos. Fonte: Autor

Espera-se que, com a visualização do gráfico, os médicos e pacientes tenham uma visão mais clara do índice glicêmico atual e, com a tendência explícita, seja possível traçar medidas preventivas para o paciente. Além de manter um histórico bem retratado de todas as medidas num grande espaço de tempo. Na figura 12 os dados estão representados meramente como testes, sem ser dados fiéis à realidade. Para os experimentos de homologação, casos de testes mais perto da realidade foram criados.

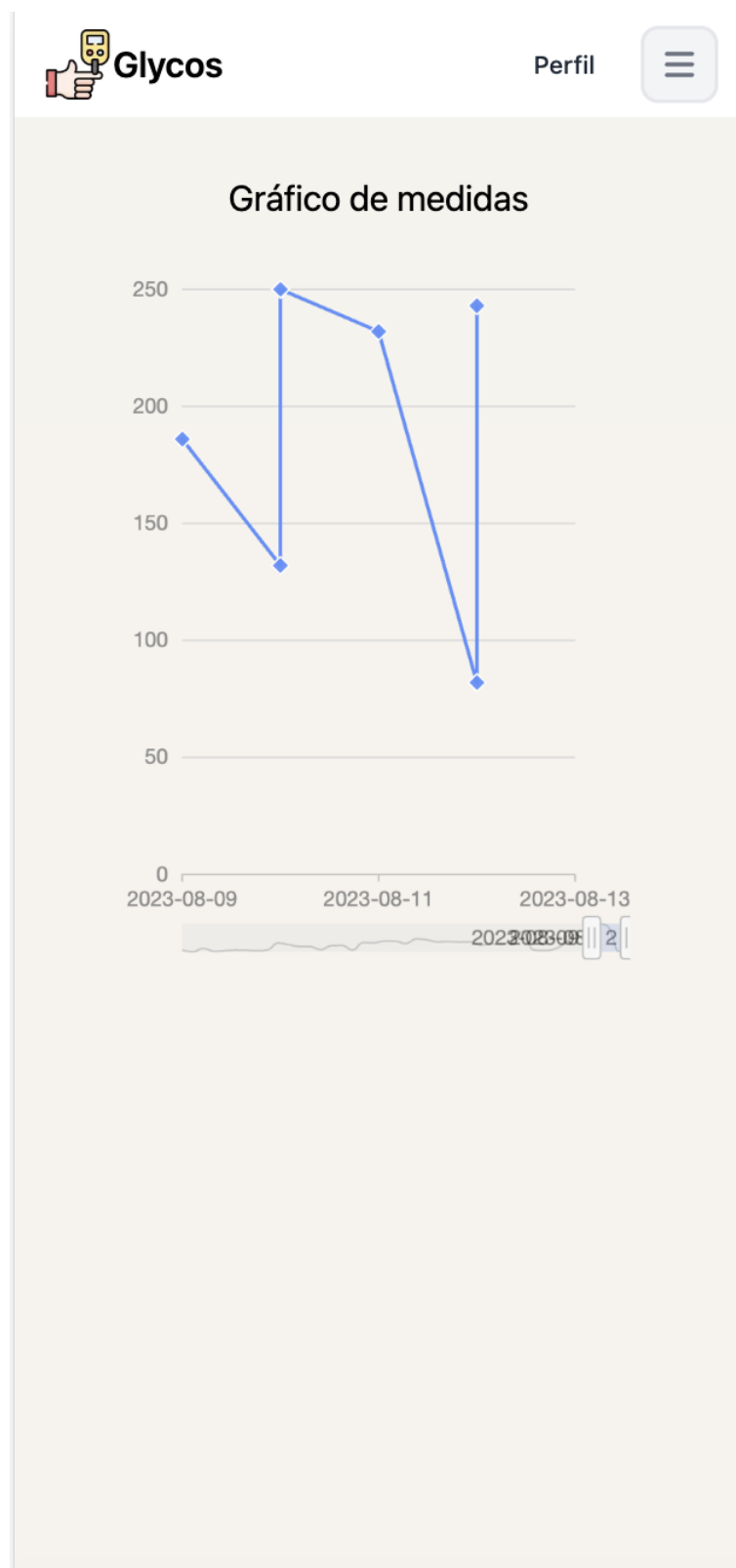


Figura 13 – Página de gráficos em dispositivo móvel. Fonte: Autor

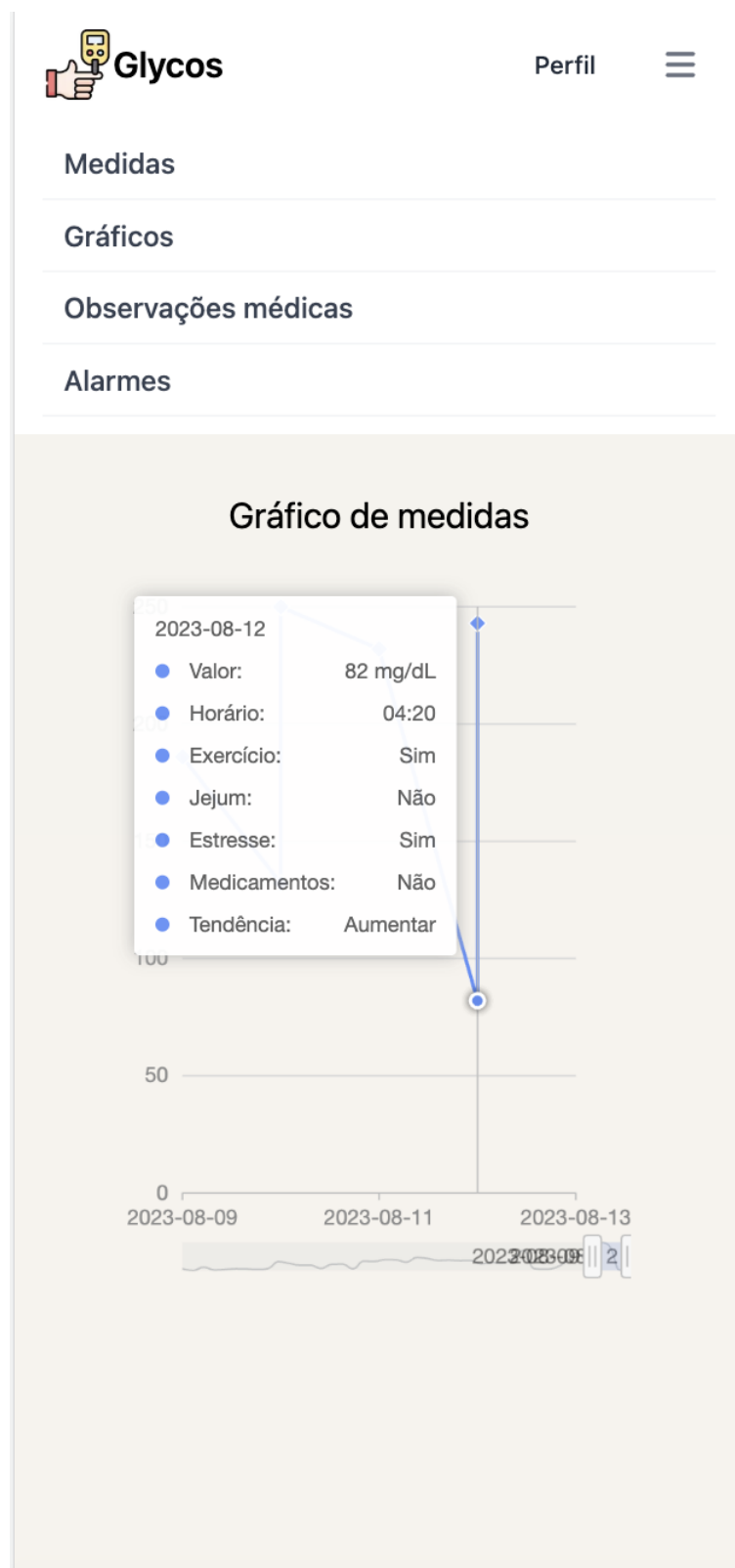


Figura 14 – Página de gráficos em dispositivo móvel com detalhes. Fonte: Autor



Figura 15 – Página de gráficos da PoC. Fonte: (SOUZA, 2021)

A figura 16 mostra a página de observações médicas, cuja finalidade é ser um espaço no qual o médico pode se comunicar indiretamente com o seu paciente ao analisar suas medidas na tabela e gráfico. Dessa forma, é possível que o médico deixe alguma mensagem para o paciente como algo sobre o tratamento ou alguma medida preventiva que deve tomar. A tela da PoC se encontra na figura 18

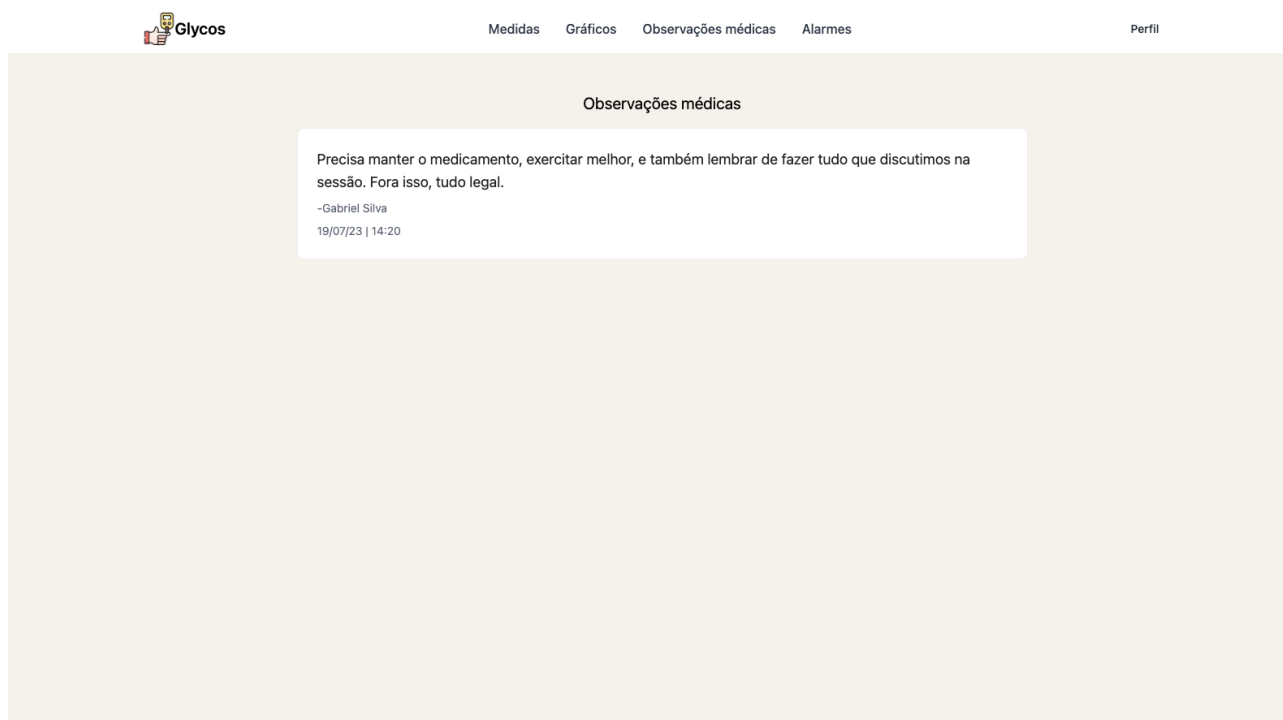


Figura 16 – Página de observações médicas. Fonte: Autor

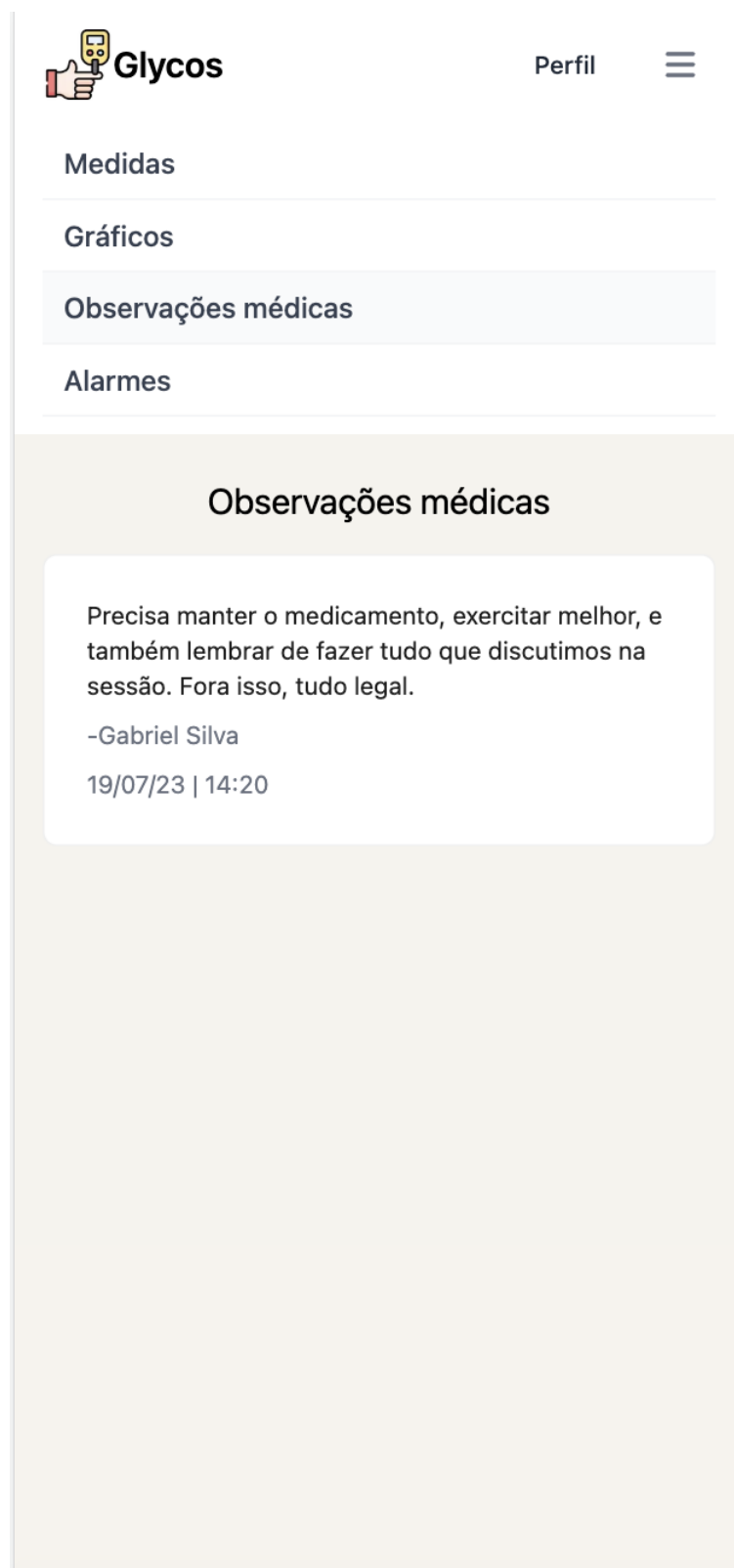


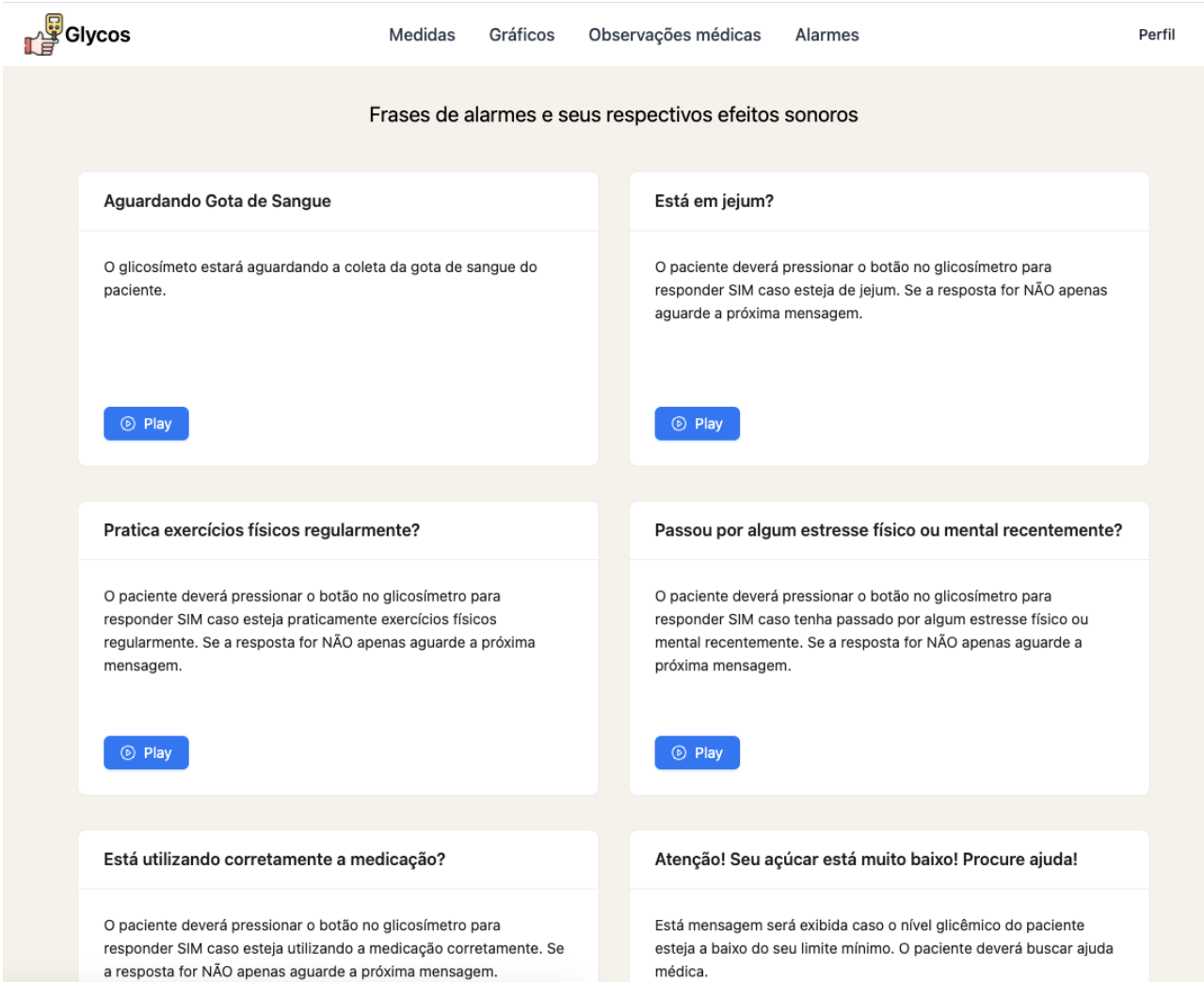
Figura 17 – Página de observações médicas em dispositivo móvel. Fonte: Autor



Figura 18 – Página de observações médicas da PoC. Fonte: (SOUZA, 2021)

A figura 19 mostra a página de alarmes. Esta é uma das partes mais importantes do glicosímetro, visto que é um dos diferenciais para os outros glicosímetros de mercado. A emissão de sons para auxiliar nas medidas é de extrema importância, dado que o DM pode causar problemas visuais, como apontado na pesquisa “Associação entre diabetes mellitus e doenças oculares em pessoas com deficiência visual”, realizada pela UERJ (Universidade Estadual do Rio de Janeiro) em 2020 (E.S. et al., 2020).

Logo, o glicosímetro em questão, possui a capacidade de ler em voz alta as perguntas para o paciente saber o que responder em cada momento da coleta de dados. Os alarmes que o glicosímetro emite são para o paciente conseguir responder às perguntas com “Sim” ou “Não”. Além disso, existem sons para avisar o paciente de que sua glicose está muito abaixo do comum ou muito acima, sugerindo-se que se encaminhe a um médico. Estes que são os mesmos sons presentes no trabalho da PoC. Os textos para os alarmes também são os mesmos utilizados anteriormente. A tela pode ser encontrada na figura 21



Glycos Medidas Gráficos Observações médicas Alarmes Perfil

Frases de alarmes e seus respectivos efeitos sonoros

Aguardando Gota de Sangue O glicosímetro estará aguardando a coleta da gota de sangue do paciente. Play	Está em jejum? O paciente deverá pressionar o botão no glicosímetro para responder SIM caso esteja de jejum. Se a resposta for NÃO apenas aguarde a próxima mensagem. Play
Pratica exercícios físicos regularmente? O paciente deverá pressionar o botão no glicosímetro para responder SIM caso esteja praticando exercícios físicos regularmente. Se a resposta for NÃO apenas aguarde a próxima mensagem. Play	Passou por algum estresse físico ou mental recentemente? O paciente deverá pressionar o botão no glicosímetro para responder SIM caso tenha passado por algum estresse físico ou mental recentemente. Se a resposta for NÃO apenas aguarde a próxima mensagem. Play
Está utilizando corretamente a medicação? O paciente deverá pressionar o botão no glicosímetro para responder SIM caso esteja utilizando a medicação corretamente. Se a resposta for NÃO apenas aguarde a próxima mensagem.	Atenção! Seu açúcar está muito baixo! Procure ajuda! Esta mensagem será exibida caso o nível glicêmico do paciente esteja a baixo do seu limite mínimo. O paciente deverá buscar ajuda médica.

Figura 19 – Página de alarmes. Fonte: Autor

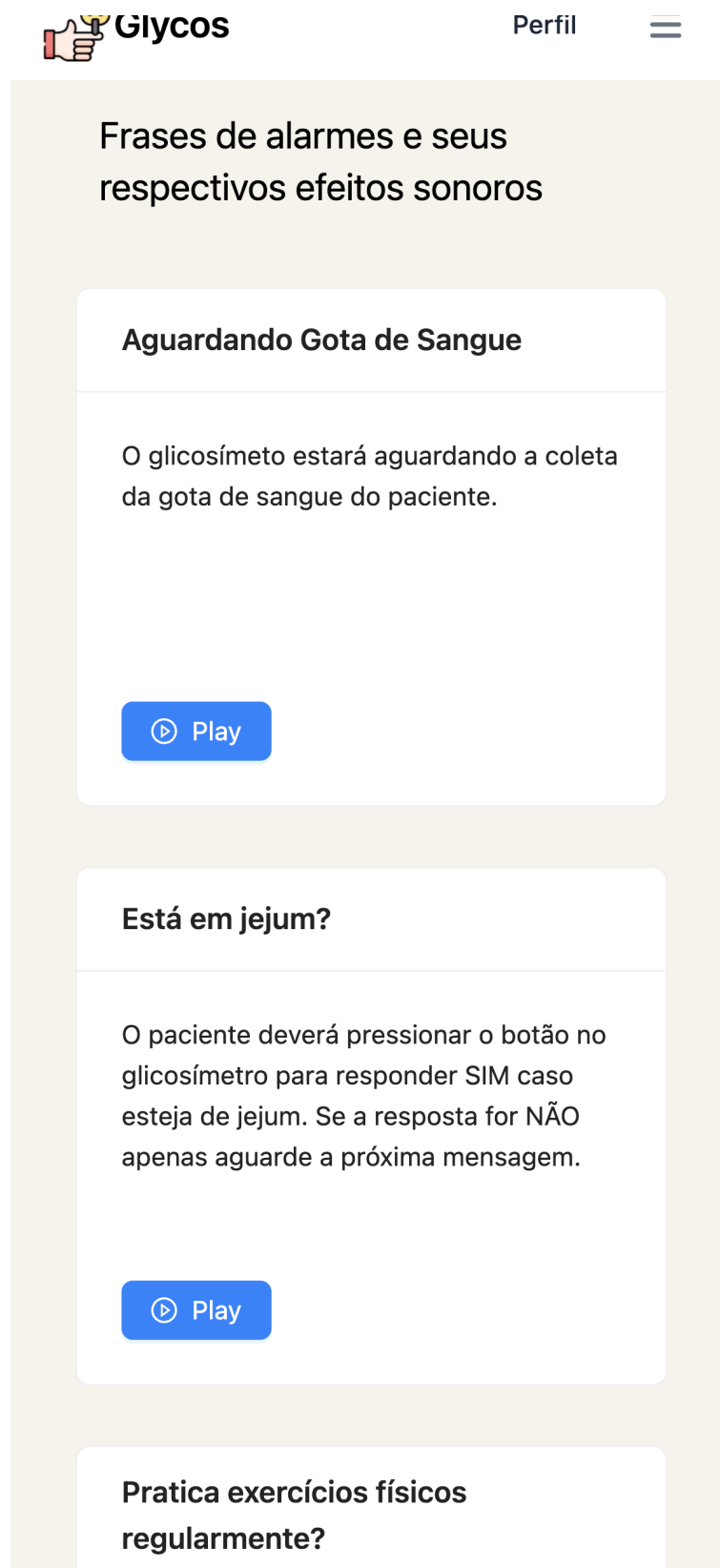


Figura 20 – Página de alarmes em dispositivo móvel. Fonte: Autor

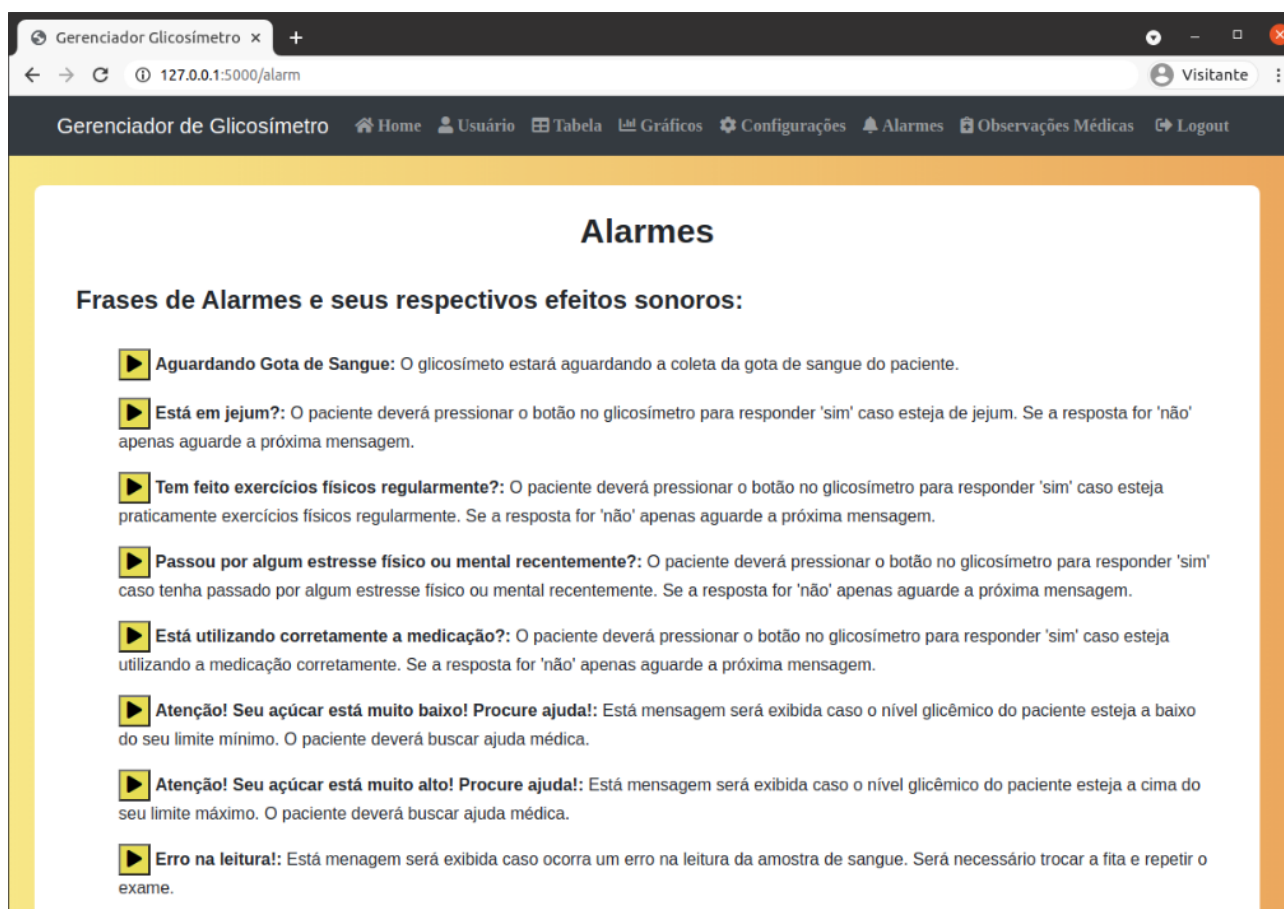


Figura 21 – Página de alarmes da PoC. Fonte: (SOUZA, 2021)

Espera-se que com isso seja mais fácil e prático para pacientes com deficiências visuais medirem a glicose e ter o controle da mesma. É possível também utilizar assistentes de leitura para áudio na grande maioria dos navegadores, podendo então ter um acesso completo ao sistema. Vale citar também que todas as imagens presentes no sistema possuem textos alternativos justamente para esses leitores de página.

Por fim, a última página do MVP é a página do perfil na figura 22, na qual o usuário terá acesso aos seus dados que foram cadastrados inicialmente antes de entrar na plataforma. Nessa tela o usuário pode conferir o nome, e-mail, CPF e o tipo de usuário (paciente, familiar ou médico). A tela de perfil da PoC pode ser encontrada na figura 24.

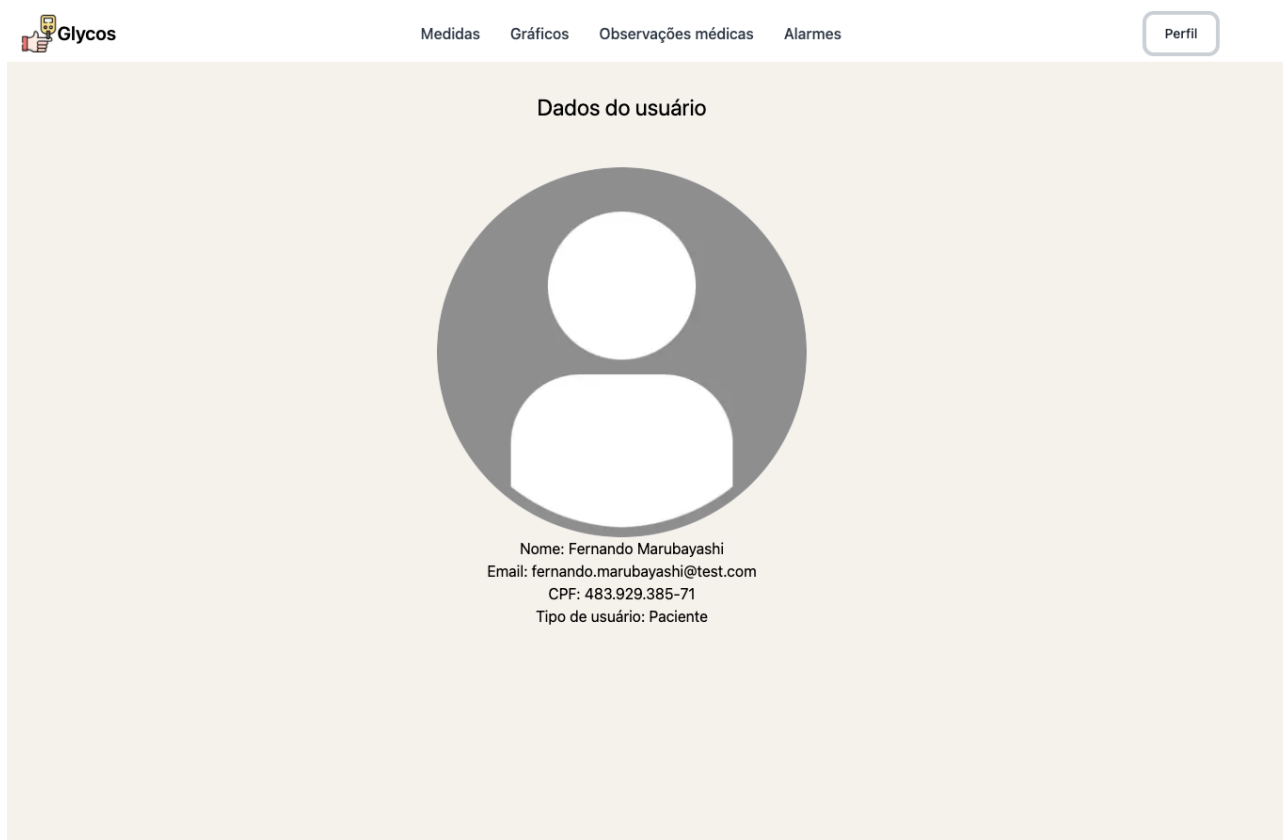


Figura 22 – Página de perfil de usuário. Fonte: Autor

É importante lembrar que as telas do *front-end* foram feitas com o MVP em mente, ou seja, são o mínimo necessário para que o sistema execute suas funções mais básicas e torne possível a homologação do projeto. Ao longo do tempo novas telas e módulos serão adicionados ao sistema, o que será discutido mais a fundo na seção de melhorias do projeto no capítulo de conclusão 5.

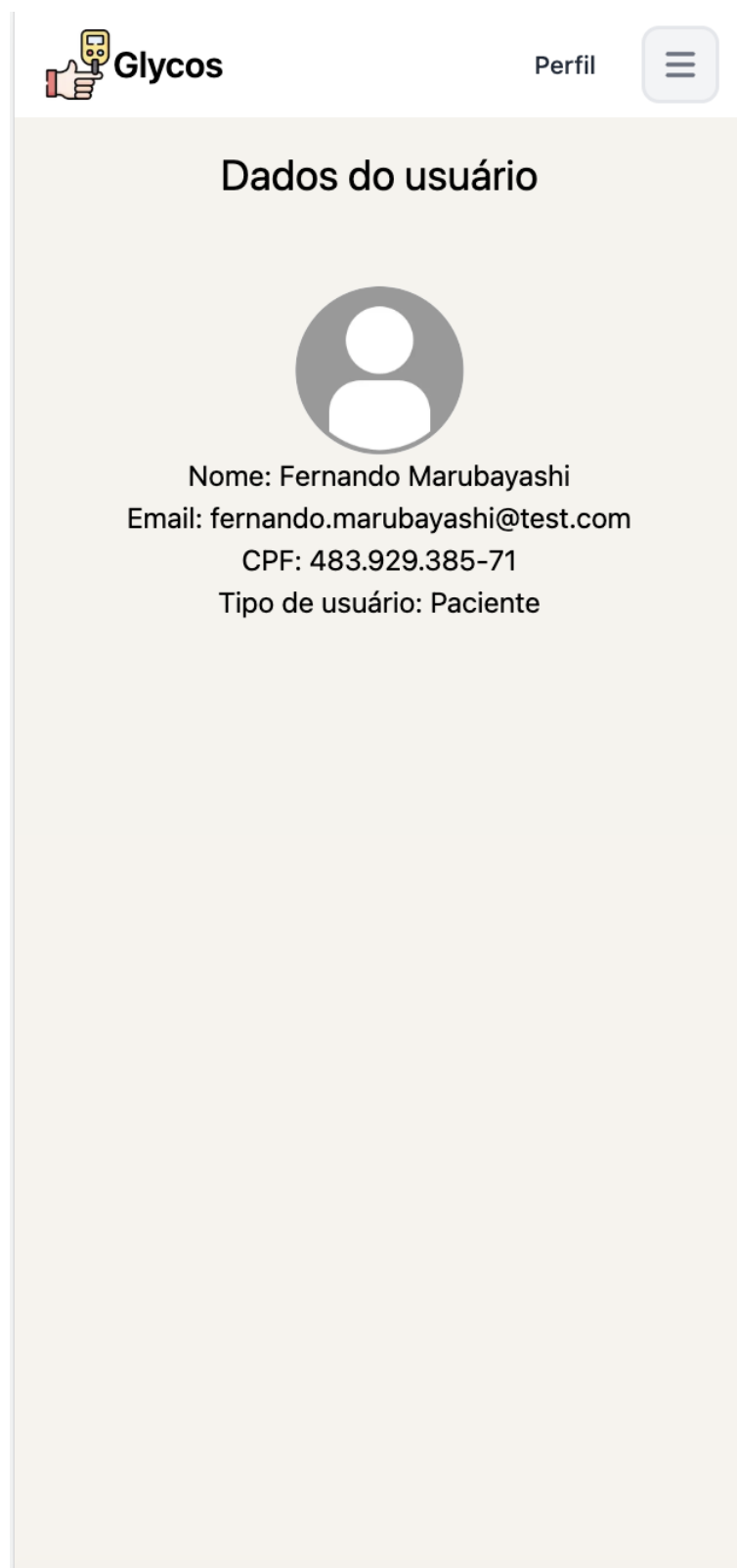


Figura 23 – Página de perfil do usuário em dispositivo móvel. Fonte: Autor



Figura 24 – Página de perfil de usuário da PoC. Fonte: (SOUZA, 2021)

4 Homologação

Com o MVP feito foi possível partir para a parte de homologação, na qual diversos testes de bancada teriam de ser feitos para garantir que o sistema e o glicosímetro estariam de acordo com as opções mais tradicionais e de mercado. Além disso, a lógica fuzzy citada anteriormente precisaria ser testada com uma grande quantidade de dados para analisar se as tendências calculadas estavam corretas ou não. Os resultados e saídas foram passados para as doutoras citadas anteriormente para que fossem validados por elas também, assim como uma especialista em diabetes.

Os testes foram realizados em quatro modelos diferentes:

1. Medidas boas;
2. Medidas quase ideais;
3. Medidas não ideais;
4. Medidas ruins;

Dessa forma foi possível observar como o sistema e a lógica fuzzy se comportam com diferentes conjuntos de medidas. Para todos os casos foram inseridas 20 medidas, pois é o valor que o sistema utiliza para fazer os cálculos de tendência e foram analisadas as tendências que a lógica retornou.

4.1 Medidas boas

As entradas para as medidas boas foram:

- Glicose com valores normais = sempre;
- Exercício físico = sempre;
- Estresse = nunca;
- Medicação = sempre;

Na figura 25 pode-se ver o gráfico para essas entradas. Aqui fica evidente que, com valores normais de glicose e exercício e medicação constantes, a tendência é de estabilizar em valores normais.

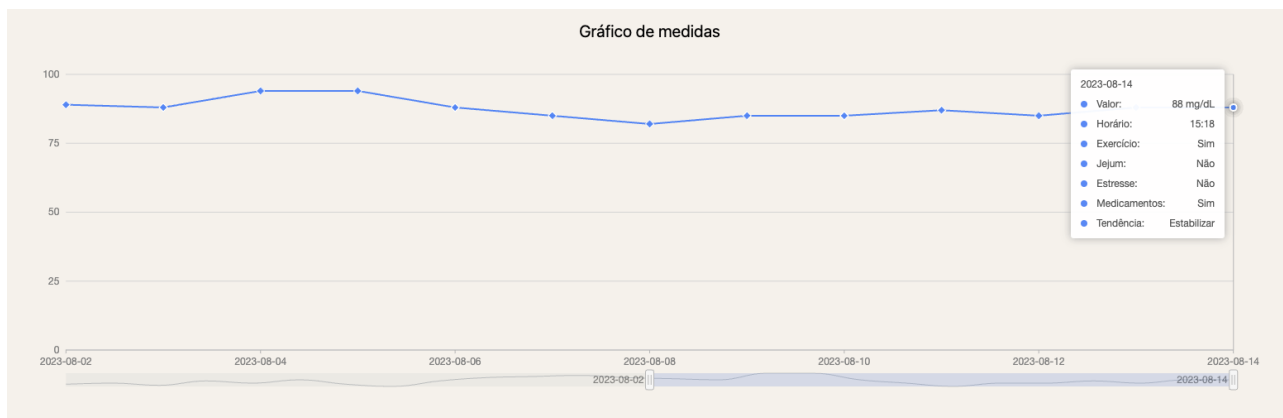


Figura 25 – Gráfico de medidas para valores bons. Fonte: Autor

4.2 Medidas quase ideais

As entradas para as medidas quase ideais foram:

- Glicose com valores elevados = predominantemente sim;
- Exercício físico = predominantemente sim;
- Estresse = predominantemente não;
- Medicação = sempre;

Na figura 26 pode-se ver o gráfico para essas entradas. Nesse caso é possível observar como a lógica fuzzy retorna uma tendência de estabilizar num valor alto de glicose, então uma medida preventiva que o médico do paciente poderia tomar, ao olhar para esses resultados, seria, por exemplo, uma alimentação mais saudável ou então a troca de medicação, ou uma nova rotina de exercício físico.

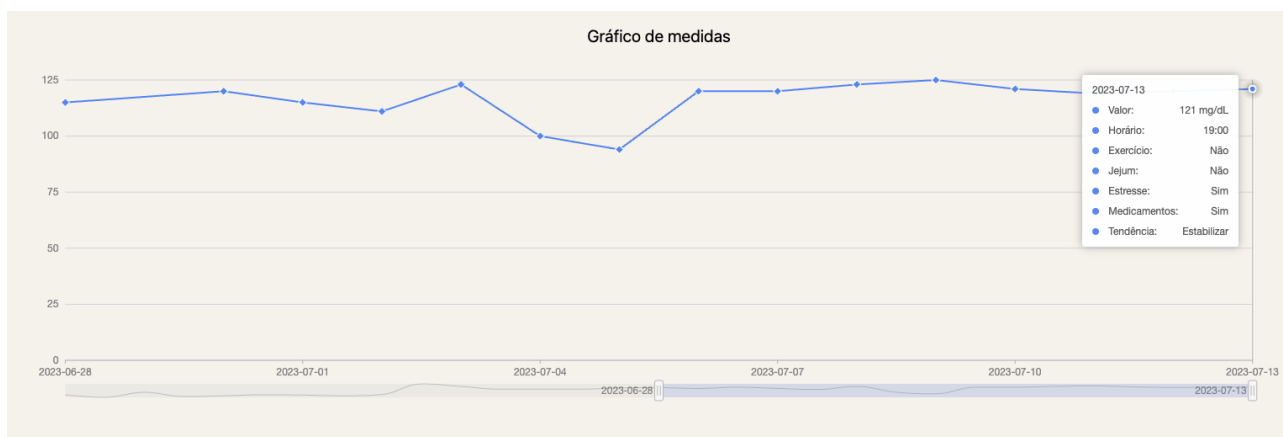


Figura 26 – Gráfico de medidas para valores quase ideais. Fonte: Autor

4.3 Medidas não ideais

As entradas para as medidas não ideais foram:

- Glicose com valores baixo = às vezes;
- Exercício físico = às vezes;
- Estresse = predominantemente sim;
- Medicação = às vezes;

Na figura 27 os valores de glicose no sangue estão variando bastante e o não uso da medicação está constante nas últimas medidas. O paciente também não está se exercitando com frequência. É importante citar que o exercício e a medicação são os fatores mais importantes para o aumento ou diminuição da glicose sanguínea segundo o artigo “Os efeitos do exercício físico sobre o Diabetes Mellitus tipo 2” (D., 2010). A lógica fuzzy indica a tendência de aumento da glicose que está correto devido aos dados apresentados.

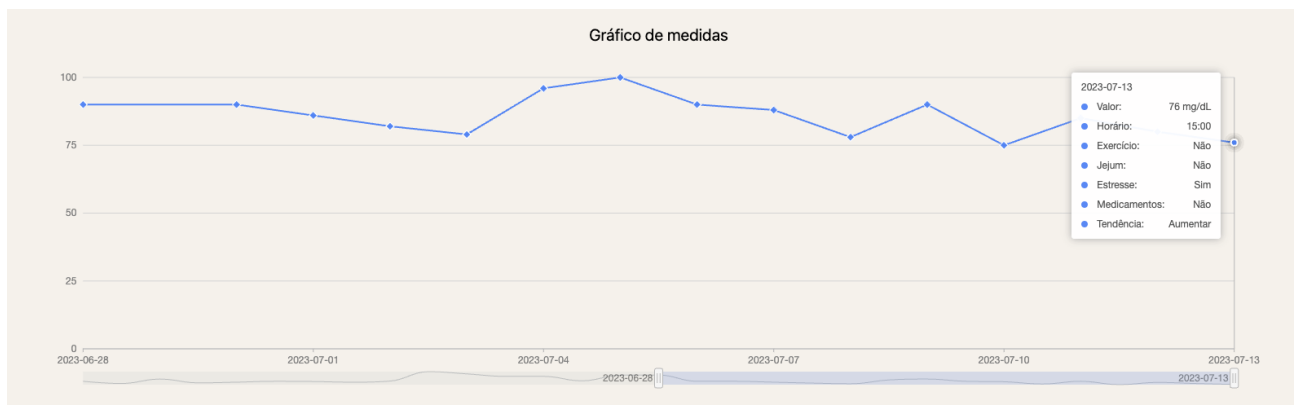


Figura 27 – Gráfico de medidas para valores não ideais. Fonte: Autor

4.4 Medidas ruins

As entradas para as medidas ruins foram:

- Glicose com valores altos = predominantemente sim;
- Exercício físico = nunca;
- Estresse = predominantemente sim;
- Medicação = nunca;

Por fim, na figura 28 estão os valores ruins, nos quais a falta de exercício físico e medicação estão sempre presentes, assim como o estresse mental ou físico. Com isso, assim como para os valores não ideais, a tendência é a glicose aumentar. Nesse caso uma medida preventiva mais urgente deveria ser implementada, uma vez que os valores estão altos e a tendência ainda é de aumentar. Medidas como a troca de medicação e reforço da importância do exercício físico, assim como possivelmente o monitoramento da glicose a partir de aparelhos CGM que, como vistos no capítulo de revisão 3.1, possuem um custo mais elevado, mas trazem medidas mais precisas.

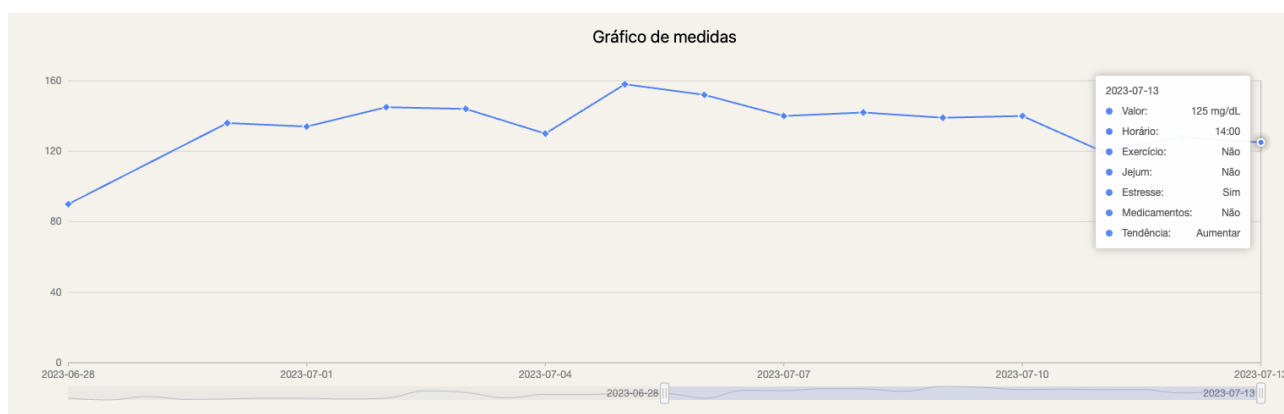


Figura 28 – Gráfico de medidas para valores ruins. Fonte: Autor

5 Conclusões finais

Ao observar os resultados do experimento e os objetivos do projeto, assim como os resultados esperados, pode-se concluir que o projeto obteve êxito no que se propôs a fazer. O produto final foi um MVP que possui as condições mínimas de produto para realizar as medidas, prever a tendência de glicose e exibir os dados para os usuários. Além disso, as tecnologias utilizadas garantem a modernização da plataforma, trazendo-a para um contexto atual. O sistema para visualização dos dados, assim como a instalação do PWA se encontra no domínio glycos-mvp.vercel.app.

Quanto aos resultados esperados, a revisão foi feita corretamente, o design e software da PoC foi trocada com sucesso visto que o aplicativo atende a todos os requisitos citados anteriormente e pode ser instalado com facilidade em qualquer dispositivo móvel. Os experimento também foram bem sucedidos, uma vez que os resultados comprovam o funcionamento de todas as funcionalidades do sistema. Os cálculos de tendência foram os esperados e podem ser explicados com base nos materiais de estudo, a comunicação entre o sistema e o glicosímetro foi validado a partir da simulação que envia dados para o broker, e a API foi validada pela exibição dos dados nas telas do *front-end*.

5.1 Trabalhos Futuros

Este trabalho abre espaço para que futuros trabalhos expandam em cima da tecnologia criada. O software foi feito com a escalabilidade em mente, então existe a facilidade de dar continuidade no desenvolvimento caso outros módulos de funcionalidades sejam implementadas.

Além disso, existem pontos de melhoria no trabalho que já foram encontrados e que poderiam ser a continuidade deste. Dentre eles estão:

1. Segurança de dados reforçada com criptografia de senhas e dados;
2. Possibilidade de resposta às observações médicas;
3. Página do usuário mais complexa;
4. Testes com dados reais que não foram realizados neste trabalho;
5. Sistema de notificações para alerta do médico e lembrar de medir a glicose;

Primeiramente, a melhoria da segurança de dados é algo de extrema importância, principalmente por se tratar de dados médicos. Existem medidas preventivas como trata-

mento de formulários para evitar injeções SQL, senhas para acesso à plataforma e banco de dados. Porém, caso o produto seja lançado ao público, será necessário reforçar melhor o acesso ao banco de dados. Uma solução possível seria utilizar serviços pagos como o *Amazon Web Service* (AWS) para guardar os dados de forma protegida e cujo acesso é restrito, assim como diz a documentação do mesmo (AWS..., 2023).

Para as observações médicas seria interessante permitir que o usuário responda às observações de seu médico, melhorando a comunicação entre ambos e podendo tirar dúvidas quanto às instruções ou alguma informação que tenha ficado confusa para o paciente. Uma opção para essa melhoria seria implementar uma espécie de linha do tempo de observações, as quais apenas os médicos poderiam postar mensagens e os usuários poderiam respondê-las como comentários. Dessa forma, toda comunicação ficaria restrita ao aplicativo, sem a necessidade de utilizar diversos meios para estabelecer a comunicação entre médico e paciente.

A página do usuário foi a página menos trabalhada nesse projeto, pois não tinha a mesma importância para o MVP quanto as outras. Algumas melhorias mapeadas são:

- Explicitar as medicações tomadas pelo paciente;
- Inserir foto de perfil;
- Explicitar o nome e CRM do médico responsável;

Para esse trabalho os testes realizados foram com dados fictícios, simulando dados reais. Ao tentar utilizar dados reais anônimos primeiramente seria necessário realizar um pedido ao Comitê de Ética em Pesquisa para qualquer pesquisa que envolva dados humanos, essa informação é encontrada no site do mesmo (COMITÊ..., 2021). Uma vez que esses pedidos costumam demorar grande tempo para serem aprovados, para este trabalho não foi possível utilizar dados reais de pacientes com diabetes. Para futuros trabalhos seria interessante realizar o pedido com máxima antecedência a fim de validar o sistema também com dados reais. Além disso, a validação do *front-end* com usuários reais serviria para entender se as telas estão fáceis de entender e utilizar.

Por fim, o sistema de notificações seria um próximo passo importante do projeto. À cada medida do paciente, o médico poderia receber uma notificação para entrar no sistema e acompanhá-la. Ou caso uma medida esteja muito alta ou muito baixa, o médico seria alertado para intervir e prestar ajuda ao paciente, assim como os familiares. Uma proposta de plataforma de notificações seria o Twilio cuja documentação propõe ter suporte para NodeJS e Express, as duas tecnologias utilizadas no *back-end* do MVP (TWILIO, 2023).

Referências

- API REST. 2023. <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Accessed: 2023-08-09. Citado na página 17.
- AWS banco de dados na nuvem. 2023. <<https://aws.amazon.com/pt/products/databases/>>. Accessed: 2023-08-13. Citado na página 47.
- BICKEL, D. *How Does a PWA Work? (and Other Questions You Were Afraid to Ask)*. 2022. <<https://mercuryworks.com/blog/everything-you-wanted-to-know-about-pwas>>. Accessed: 2023-08-22. Citado na página 25.
- BORGES, V. A.; KATO, E. R. R.; LEAL, A. M. O. Pedido nacional de invencao, modelo de utilidade, certificado de adicao de invencao e entrada na fase nacional do pct. "dispositivo de controle de glicemia com interface auditiva, alertas e conectividade wi-fi.". 2019. Citado 3 vezes nas páginas 9, 13 e 14.
- CAMPAGNOLI, M. P. Glicose alta, baixa e normal: Entenda o que significa. 2023. Accessed: 2023-08-30. Citado na página 20.
- COMITê de Ética em Pesquisa (CEP/HFA). 2021. <<https://www.gov.br/defesa/pt-br/assuntos/hfa/ensino-e-pesquisa/comite-de-etica-em-pesquisa-cep-hfa-1>>. Accessed: 2023-08-13. Citado na página 47.
- CONTINUOUS Glucose Monitoring. <<https://my.clevelandclinic.org/health/drugs/11444-glucose-continuous-glucose-monitoring>>. Accessed: 2023-07-09. Citado na página 14.
- D., J. L. B. Os efeitos do exercício físico sobre o diabetes mellitus tipo 2. *UNISUL*, 2010. Disponível em: <https://repositorio.animaeducacao.com.br/bitstream/ANIMA/12773/1/101765_Lunara.pdf>. Citado na página 44.
- EMQX. 2023. <<https://www.emqx.com/en/blog/how-to-use-mqtt-in-nodejs>>. Accessed: 2023-08-09. Citado 2 vezes nas páginas 17 e 22.
- E.S., B. et al. Associação entre diabetes mellitus e doenças oculares em pessoas com deficiência visual. *Revista de enfermagem UERJ*, 2020. Citado na página 36.
- EXPRESSJS. 2023. <<https://expressjs.com/pt-br/>>. Accessed: 2023-07-16. Citado na página 19.
- INTERNATIONAL Diabetes Federation. 2021. <<https://www.diabetesatlas.org>>. Accessed: 2023-08-13. Citado na página 9.
- MEAN fasting blood glucose. 2023. <<https://www.who.int/data/gho/indicator-metadata-registry/imr-details/2380>>. Accessed: 2023-08-01. Citado na página 20.
- MORAGA, C. Introduction to fuzzy logic. *Facta universitatis - series: Electronics and Energetics*, v. 18, p. 319–328, 09 2005. Citado na página 10.

- MQTT. 2023. <<https://github.com/mqttjs/MQTT.js#readme>>. Accessed: 2023-08-13. Citado na página 22.
- NELSON, C. Continuous glucose monitoring (cgms) vs. blood glucose monitoring (finger stick): What's the difference? 2021. Accessed: 2023-07-09. Citado na página 14.
- NEXT-PWA. 2023. <<https://github.com/shadowwalker/next-pwa>>. Accessed: 2023-08-01. Citado na página 25.
- NEXT.JS. 2023. <<https://nextjs.org/>>. Accessed: 2023-08-01. Citado na página 25.
- NODEJS. 2023. <<https://nodejs.org/en>>. Accessed: 2023-08-10. Citado 2 vezes nas páginas 16 e 18.
- NOTÍCIAS, A. C. de. *Medida provisória aumenta salário mínimo para R\$1.320 a partir de maio*. 2023. <>. Accessed: 2023-07-09. Citado na página 14.
- O que é MVC? 2020. <<https://www.treinaweb.com.br/blog/o-que-e-mvc>>. Accessed: 2023-07-09. Citado na página 17.
- QUAL a diferença entre monitoramento contínuo de glicose e ponta de dedo? 2023. <<https://www.freestyle.abbott/br-pt/blog/monitoramento-contínuo-de-glicose-ou-ponta-de-dedo.html>>. Accessed: 2023-08-22. Citado na página 14.
- RATAKONDA, M. C. Fundamentals of glucose monitoring sensors. 2021. Citado na página 14.
- SANTOS, M. M. de Azevedo dos. *TECNOVIGILÂNCIA – Estudo de Caso - GLICO-SÍMETRO*. 2013. <https://www.saude.pr.gov.br/sites/default/arquivos_restritos/files_documento/2020-05/estudocasoglicosimetro.pdf>. Citado na página 14.
- SOUZA, G. d. Um sistema de coleta e análise de dados em nuvem utilizando técnicas de inteligência artificial para auxílio ao tratamento e autocuidado da diabetes mellitus. *Universidade Federal de São Carlos — Centro de Ciências Exatas e de Tecnologias — Departamento de Computação*, 2021. Citado 12 vezes nas páginas 6, 7, 11, 12, 14, 15, 21, 28, 32, 35, 38 e 41.
- TAILWINDCSS. 2023. <<https://tailwindcss.com/>>. Accessed: 2023-08-22. Citado na página 26.
- TWILIO. 2023. <<https://www.twilio.com/blog/server-notifications-node-express>>. Accessed: 2023-08-13. Citado na página 47.
- TYPEORM. 2023. <<https://typeorm.io/>>. Accessed: 2023-07-13. Citado na página 23.
- TYPESCRIPT. 2023. <<https://www.typescriptlang.org/>>. Accessed: 2023-08-10. Citado 2 vezes nas páginas 16 e 18.

VALENZUELA, F. et al. An iot-based glucose monitoring algorithm to prevent diabetes complications. *Applied Sciences*, v. 10, n. 3, 2020. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/10/3/921>>. Citado na página 13.

VINCENZI, A. Prática de devops com docker. 2020. Citado na página 22.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338–353, 1965. ISSN 0019-9958. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S001999586590241X>>. Citado na página 19.