



*Trabalho de Conclusão de Curso*

# Sistema inteligente baseado em conhecimento para acompanhamento de pré-diabéticos

de Jacques Wolbeck Saraiva de Melo Godoy Amorim

orientado por  
Prof. Dr. Leandro Dias da Silva

Universidade Federal de Alagoas  
Instituto de Computação  
Maceió, Alagoas  
18 de Outubro de 2023

UNIVERSIDADE FEDERAL DE ALAGOAS  
Instituto de Computação

## SISTEMA INTELIGENTE BASEADO EM CONHECIMENTO PARA ACOMPANHAMENTO DE PRÉ-DIABÉTICOS

Trabalho de Conclusão de Curso submetido  
ao Instituto de Computação da Universidade  
Federal de Alagoas como requisito parcial  
para a obtenção do grau de Engenheiro de  
Computação.

Jacques Wolbeck Saraiva de Melo Godoy Amorim

*Orientador: Prof. Dr. Leandro Dias da Silva*

**Banca Avaliadora:**

Baldoino Fonseca dos Santos Neto                      Dr., UFAL  
Álvaro Alvares de Carvalho César Sobrinho      Dr., UFAPE

Maceió, Alagoas  
18 de Outubro de 2023

**Catálogo na Fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

A524s Amorim, Jacques Wolbeck Saraiva de Melo Godoy.  
Sistema inteligente baseado em conhecimento para  
acompanhamento de pré-diabéticos / Jacques Wolbeck Saraiva de  
Melo Godoy Amorim. – 2023.  
37 f. : il.

Orientador: Leandro Dias da Silva.  
Monografia (Trabalho de conclusão de curso em Engenharia de  
Computação) - Universidade Federal de Alagoas, Instituto de Computação.  
Maceió, 2023.

Bibliografia: f. 36-37.

1. Diabetes mellitus 2. Estado pré-diabético. 3. Sistemas inteligentes. 4.  
Aplicativos móveis. I. Título.

CDU: 004.81:631

# Dedicatória

Dedico este trabalho a minha família.

*Jacques Wolbeck Saraiva de Melo Godoy Amorim,*

# Agradecimentos

Aos meus pais Wolbeck e Genyenne, que sempre me deram suporte e incentivo para alcançar meus objetivos de vida. Agradeço também ao meu irmão João e minha namorada Tiffany pelo apoio e incentivo em momentos difíceis até a conclusão deste TCC.

Ao corpo docente do Instituto de Computação, em especial ao meu orientador Prof. Dr. Leandro Dias da Silva que se prontificou a me auxiliar no desenvolvimento deste TCC, sempre estando disponível para me ajudar e dar soluções que eu necessitava, e também, ao Prof. Dr. Thiago Damasceno Cordeiro, que me abriu portas para oportunidades relevantes durante minha graduação.

O agradecimento final vai para Deus, por ter me dado saúde e uma excelente família e amigos, que me impulsionaram na chance de alcançar grandes conquistas em minha vida.

18 de Outubro de 2023, Maceió,  
Alagoas

*Jacques Wolbeck Saraiva de Melo*  
*Goody Amorim*

*Temos o destino que merecemos. O nosso destino está de acordo com os nossos méritos.*

*Albert Einstein*

# Resumo

No cenário atual, nota-se que as doenças metabólicas ganharam mais visibilidade para os pesquisadores e estudiosos da área da saúde, principalmente por essa temática abordada comprometer o estilo de vida da humanidade. O Diabetes *Mellitus* tipo 2 é uma dessas doenças crônicas que com o passar dos anos tem se tornado mais presente na vida dos seres humanos. Apesar de não ter cura, é possível reduzir a progressão da doença mediante a adoção de cuidados preventivos de saúde em fases anteriores ao seu desenvolvimento pleno. Neste trabalho é proposto, através de um sistema inteligente, o auxílio de pessoas que estejam no quadro de desenvolvimento ou que já possuem pré-diabetes. O objetivo é fornecer um acompanhamento contínuo dos indicativos da doença para um diagnóstico rápido, visando a possibilidade de prever ou até reverter um quadro de pré-diabetes. No que difere de quando se atinge o estágio do Diabetes, propriamente dito, onde apenas o controle da doença se torna a solução para evitar possíveis complicações. Este sistema inteligente, por meio de um aplicativo móvel, pode beneficiar principalmente pessoas que possuam dificuldade de acesso a profissionais para acompanhar sua condição de saúde de forma contínua.

***Palavras-chave: Diabetes Mellitus; Pré-diabetes; Sistema inteligente; Aplicativo Móvel.***

# Abstract

In the current scenario, it is noted that metabolic diseases have gained more visibility for researchers and scholars in the area of health, mainly because this topic compromises humanity's lifestyle. Type 2 Diabetes *Mellitus* is one of those chronic diseases that over the years has become more present in the lives of human beings. Although there is no cure, it is possible to reduce the progression of the disease by adopting preventive health care in stages prior to its full development. This work proposes, through an intelligent system, the help of people who are in the development stage or who already have pre-diabetes. The objective is to provide continuous monitoring of disease indicators for a quick diagnosis, aiming at the possibility of predicting or even reversing pre-diabetes. This differs from when the Diabetes stage itself is reached, where only controlling the disease becomes the solution to avoid possible complications. This intelligent system, through a mobile application, can mainly benefit people who have difficulty accessing professionals to monitor their health condition on an ongoing basis..

***Keywords: Diabetes Mellitus; Pre-diabetes; Intelligent system; Mobile Application.***

# Lista de Figuras

3.1	Diagrama de metodologia . . . . .	18
3.2	Padrão de arquitetura do app. Adaptado de (Vahid Dejwakh, 2021) . . . . .	22
3.3	Padrão BLoC. Adaptado de (Cap Sistema, 2020) . . . . .	23
4.1	Telas de autenticação de usuário . . . . .	24
4.2	Telas do questionário . . . . .	25
4.3	Telas de taxas . . . . .	26
4.4	Telas de medidas . . . . .	26
4.5	Telas de consultas . . . . .	27
4.6	Telas de dicas . . . . .	27
4.7	Telas de recomendações 1 . . . . .	28
4.8	Telas de recomendações 2 . . . . .	29
5.1	Histograma da base de dados (link da imagem) . . . . .	30
5.2	Correlação entre os atributos da base de dados (link da imagem) . . . . .	32
5.3	Nível de importância dos atributos para o modelo gerado (teste 1) . . . . .	33
5.4	Nível de importância dos atributos para o modelo gerado (teste 2) . . . . .	34

# Lista de Tabelas

3.1	Tabela de atributos do conjunto de dados . . . . .	20
5.1	Tabela de métricas de avaliação do primeiro teste . . . . .	33
5.2	Tabela de métricas de avaliação do segundo teste . . . . .	34

# Lista de Abreviaturas

**API** Application Programming Interface.

**BLoC** Business Logic Object Components.

**BRFSS** Behavioral Risk Factor Surveillance System.

**CDC** Centers for Disease Control and Prevention.

**IC** Instituto de Computação.

**mHealth** Mobile Health.

**SMOTE** Synthetic Minority Over-sampling Technique.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Objetivos . . . . .	13
1.1.1	Objetivo geral . . . . .	13
1.1.2	Objetivos específicos . . . . .	13
1.2	Organização do texto . . . . .	14
<b>2</b>	<b>Fundamentação Teórica</b>	<b>15</b>
<b>3</b>	<b>Metodologia</b>	<b>18</b>
3.1	Visão Geral . . . . .	18
3.2	Ferramentas e tecnologias . . . . .	18
3.3	Detalhamento das Etapas . . . . .	19
3.3.1	Pré-Processamento . . . . .	19
3.3.2	Análise dos Dados . . . . .	19
3.3.3	Aprendizado de Máquina . . . . .	20
3.3.4	Implementação da API . . . . .	21
3.3.5	Desenvolvimento do Aplicativo . . . . .	21
<b>4</b>	<b>Funcionalidades do Sistema</b>	<b>24</b>
4.1	Autenticação de Usuário . . . . .	24
4.2	Captação de Dados . . . . .	25
4.3	Agendamento de Consultas . . . . .	26
4.4	Módulo de Dicas . . . . .	27
4.5	Recomendações de Saúde . . . . .	28
<b>5</b>	<b>Resultados</b>	<b>30</b>
<b>6</b>	<b>Conclusão</b>	<b>36</b>
	<b>Bibliografia</b>	<b>37</b>

# Capítulo 1

## Introdução

O Diabetes *Mellitus* tipo 2 é uma doença silenciosa, em que o paciente apresenta uma forma de distúrbio metabólico, onde os níveis de açúcar no sangue estão em condições elevadas (hiperglicemia) (Khan et al., 2019). Esse estado está ligado diretamente a situação em que o corpo não consegue utilizar apropriadamente a insulina que produz ou esta produção não é suficiente para o controle da taxa de glicemia (Sociedade Brasileira de Diabetes, 2023), sendo este hormônio, o principal responsável por estabilizar os níveis de açúcares do sangue (glicose). Por outro lado, o pré-diabetes representa um estágio intermediário entre a capacidade normal de tolerância à glicose e o Diabetes de fato (Beulens et al., 2020). No entanto, diferente da condição de diabético, para qual nenhuma cura completa foi descoberta até o momento, a condição de pré-diabético possui tratamentos que demonstram uma taxa de sucesso significativa em sua prevenção de progressão da doença (Khan et al., 2019). Isso evita que ela atinja seu estágio mais grave, ou seja, o Diabetes.

Segundo (Atlas, 2021), só na América Central e Sul, 32 milhões de adultos vivem com Diabetes, e dentre esses, 33% não são diagnosticados. O Diabetes *mellitus* tipo 2, domina com 90% os casos da doença, de acordo com (Sociedade Brasileira de Diabetes, 2023), e por possuir um caráter silencioso, oferece uma grande dificuldade no seu combate prematuro. Essa presença da doença na vida da população mundial vem em uma crescente, ano após ano, e de acordo com (Internacional Diabetes Federation, 2021) os casos do Diabetes entre adultos na faixa de 20-79 anos podem chegar a mais de 23 milhões até 2045 só no Brasil. Um acompanhamento contínuo de um profissional de saúde é essencial para um combate eficiente a doença, no entanto, existem empecilhos no caminho que afetam os pacientes. Estes obstáculos podem ser impostos pela falta de informações sobre a doença e sua real gravidade, dificuldade de acesso ao acompanhamento de um especialista ou até mesmo a negligência do paciente em relação ao seu estado de saúde. Isso conduz um paciente que ainda esteja no estágio intermediário (pré-diabetes), a perder sua única oportunidade de retardar ou reverter a evolução para o Diabetes e suas complicações.

O avanço tecnológico ocasionou o desenvolvimento de muitas soluções na área da

saúde. Um exemplo disso são os sistemas inteligentes combinados ao conceito de *mhealth*, a qual se refere a uma prática médica, que por meio de dispositivos móveis realiza trabalhos de prevenção, monitoramento e diagnósticos de doenças (Portal Telemedicina, 2023). Muitos desses sistemas se propõe a auxiliar pacientes no controle do Diabetes, suprindo a ausência de um especialista para acompanhar o seu quadro de saúde. Isso pode ser feito através de aplicativos, plataformas online, entre outros, que monitoram as taxas e fornecem dicas sobre como manter uma condição de saúde estável, prevenindo assim possíveis complicações. Entretanto, sistemas que envolvem o pré-diabetes ainda foram pouco explorados como foco, indicando uma grande necessidade de soluções para esse campo.

Este trabalho tem o propósito de reimplementar e aprimorar o *Assistente Inteligente para Auxílio na Prevenção do Diabetes tipo 2* (Marinho et al., 2019), visando a adição de novas ferramentas que facilitarão o acompanhamento não só do Diabetes tipo 2, como o também o seu estágio anterior, o pré-diabetes, tendo como base a análise dos dados que foram coletados do *CDC – 2015 BRFSS Survey Data and Documentation* (CDC, 2015) para extrair novas percepções sobre a doença.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolver um sistema inteligente, que por meio de um aplicativo seja capaz de monitorar, avaliar e recomendar dicas de saúde a um paciente, afim de prevenir ou regredir um quadro de pré-diabetes, além de fornecer suporte para evitar o avanço da doença para estágios mais avançados (Diabetes).

### 1.1.2 Objetivos específicos

- Pré-processamento e análise dos dados presentes no conjunto de dados;
- Implementação do código utilizando algoritmos tradicionais de aprendizado de máquina, com o objetivo de criar um modelo preditivo de classificação, tendo como base o conjunto de dados;
- Desenvolvimento de uma API, que irá integrar o modelo preditivo com o aplicativo;
- Aprimorar o aplicativo existente, que será responsável por captar os dados dos pacientes e partir destes dados, realizar o acompanhamento, indicando quais condições de saúde requerem atenção.

## 1.2 Organização do texto

O presente trabalho apresenta a seguinte organização:

- Capítulo 2: Serão abordados os princípios dos sistemas inteligentes e o aprendizado de máquina, além de contextualizar a integração dessas áreas no campo da saúde
- Capítulo 3: É descrita as etapas realizadas para desenvolver o sistema, bem como, todas as tecnologias e ferramentas aplicadas no desenvolvimento
- Capítulo 4: Detalhamento das funcionalidades principais do sistema
- Capítulo 5: Discussão dos resultados obtidos de acordo com os objetivos apresentados
- Capítulo 6: Por fim, são apresentados as conclusões sobre o trabalho e uma discussão sobre limitações e possibilidades de trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

O campo de estudo na computação que envolve sistemas inteligentes vem conquistando bastante destaque entre as novas soluções tecnológicas. Seu foco está contido no desenvolvimento de sistemas nos quais os componentes sejam capazes de coletar, processar e interpretar dados, a fim de com base nas informações captadas realizar tarefas ou tomar decisões de maneira autônoma ou semi autônoma. Essa linha de estudo engloba, por exemplo, sistemas de inteligência distribuída (sistemas multiagentes), sistemas que se desenvolvem e se adaptam ao ambiente (sistemas evolutivos) e sistemas que têm a capacidade de aprender com a experiência adquirida (aprendizagem de máquina) (PPGC UFPel, 2023).

A aprendizagem de máquina é um subcampo da inteligência artificial que se utiliza de dados e algoritmos para simular o processo de aprendizado humano, aprimorando progressivamente sua precisão. Utilizando métodos estatísticos, os algoritmos são treinados para realizar classificações, previsões e identificar informações cruciais acerca dos dados utilizados (IBM, 2023). Essas descobertas orientam posteriormente a tomada de decisões em sistemas inteligentes por exemplo. Os modelos de aprendizado de máquina se enquadram em três categorias principais, são elas:

- **Aprendizado supervisionado:** Esta categoria é definida pelo uso de conjuntos de dados rotulados para treinar os algoritmos, ocasionando um aprendizado capaz de classificar dados ou prever resultados com precisão;
- **Aprendizado não supervisionado:** Esse tipo de aprendizagem faz uso do treinamento de algoritmos tendo como base um conjunto de dados que não possui rótulos ou categorias predeterminadas, a fim de capacitá-los a encontrar padrões ou estruturas presentes na amostra. Tais algoritmos revelam padrões ocultos e agrupamentos de dados de forma autônoma, sem a necessidade de interferência humana (IBM, 2023);
- **Aprendizado por reforço:** Nesta categoria, um agente aprende a interagir com um ambiente realizando ações e recebendo recompensas ou penalidades com base em

suas ações. O intuito da aprendizagem por reforço é desenvolver uma estratégia, que consiste em mapear os diferentes estados para as ações, visando maximizar a expectativa de recompensa ao longo do tempo (Geeks for Geeks, 2023) .

O enfoque central deste trabalho situa-se em um sistema inteligente baseado em conhecimento no qual é realizado o uso da análise de uma grande quantidade de dados, juntamente com a aplicação de algoritmos de aprendizagem de máquina. Esses procedimentos visam extrair das informações adquiridas o conhecimento necessário para recomendar as melhores decisões possíveis de alcançar os objetivos propostos, aplicando os modelos de classificação desenvolvidos. Sistemas deste tipo podem ser definidos como sistemas de recomendação, sendo amplamente aplicados a área da saúde. Seu propósito quando utilizado nesta área, está no foco em monitorar condições de saúde, melhorar índices de acertos em diagnósticos, reduzir custos e também a partir dos dados coletados conseguir trazer novas percepções sobre as doenças que antes não foram notadas. Assim o sistema obtém a capacidade de recomendar ao paciente quais são as melhores soluções para se alcançar uma melhora em sua condição de saúde, como também, indicar a necessidade de consulta com um profissional caso apenas o monitoramento por meio de sistemas não seja suficiente.

Estudos na área, utilizam implementações de algoritmos de aprendizado de máquina para beneficiar profissionais da medicina e melhorar eficiência dos sistemas de saúde, como mostra o estudo elaborado por (Dinh et al., 2019), em que foi realizado uma abordagem baseado em dados para prever o Diabetes e doenças cardiovasculares. Nesse estudo, são apresentados resultados após se utilizar modelos de aprendizagem de máquina supervisionados em um conjunto de dados, onde para o Diabetes por exemplo, foi possível identificar os cinco melhores preditores em pacientes com Diabetes: (comprimento abdominal, idade, maior peso auto-relatado, comprimento da perna e ingestão de sódio). Isso demonstra o potencial de sistemas desse nível, nos quais, apenas pela análise de um conjunto de dados foi possível retirar percepções sobre características que pacientes diabéticos apresentam, facilitando assim diagnósticos prévios do mesmos.

Outros estudos, como (Yamaguchi et al., 2019) e (Waki et al., 2015), aliam as tecnologias de captação de dados presentes em smartphones para potencializar os desenvolvimentos de aplicativos capazes de dar suporte aos pacientes portadores do Diabetes tipo 2. As soluções propostas envolvem o auto monitoramento de peso corporal, níveis de açúcar no sangue, atividades físicas representadas por contagem de passos e dieta. Dessa forma, por meio de dispositivos móveis, o acompanhamento de doenças podem ser realizados.

Já (Kriventsov et al., 2020) faz um trabalho mais preciso na questão de monitoramento do Diabetes. Isso ocorre devido à sua integração com dispositivos de medição de glicose os CGM (*continuous glucose monitoring*). O aplicativo utiliza modelos preditivos para gerar resultados ao paciente, indicando se é necessário corrigir a ingestão de insulina,

alimentos ou outros comportamentos, a fim de evitar possíveis casos de hipoglicemia ou hiperglicemia.

O propósito deste trabalho é se diferenciar das soluções atualmente existentes, concentrando-se mais na condição do pré-diabetes, pois como mostrado, os estudos visam principalmente a questão de monitoramento e controle de pacientes já diabéticos. Através da análise dos dados de pacientes pré-diabéticos, será possível extrair informações que indiquem quais características são mais dominantes em pacientes desse estado. Isso permitirá que o sistema inteligente por meio de um aplicativo, realize um acompanhamento e diagnóstico aprimorado nestes pacientes, contribuindo assim na prevenção da evolução da doença.

# Capítulo 3

## Metodologia

### 3.1 Visão Geral

A metodologia de pesquisa aplicada a este trabalho (Figura 3.1), consiste inicialmente em três etapas: executar um pré-processamento no conjunto de dados, analisar os dados com intuito de selecionar os atributos mais relevantes no contexto do pré-diabetes e utilizar algoritmos de aprendizagem de máquina para gerar um modelo preditivo de classificação. Por fim são realizadas mais duas etapas, que correspondem a: implementar um *API* para realizar a integração do modelo treinado ao aplicativo e realizar o desenvolvimento do aplicativo tendo como base o trabalho realizado em (Marinho et al., 2019).

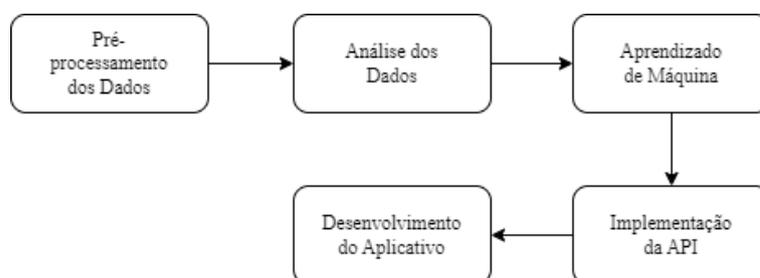


Figura 3.1: Diagrama de metodologia

### 3.2 Ferramentas e tecnologias

- A etapa inicial, contendo o pré-processamento, análise dos dados e o treinamento de algoritmos de aprendizado de máquina, ocorreu no ambiente *Google Colaboratory*. Já implementação foi realizada utilizando a linguagem de programação Python, com o suporte das bibliotecas *pandas*, *seaborn*, *scikit-learn (sklearn)* e *imbalanced-learn (imblearn)*;
- Para desenvolver a *API*, o framework *FastAPI* foi adotado junto da linguagem Python. Além disso, para criar um ambiente isolado para a *API*, o software *Doc-*

*ker* foi utilizado, gerando um container da aplicação que engloba todos os recursos necessários;

- A criação do aplicativo foi desempenhada com o uso do framework *Flutter* e sua linguagem de programação específica *Dart*, ambos são tecnologias do Google. Para lidar com questões relacionadas ao armazenamento de dados, autenticação de usuários e relatórios de falhas do aplicativo, foram utilizadas as funcionalidades fornecidas pelo *Firebase*. Especificamente, recursos como *Firestore* para armazenamento de dados, *Google Authentication* para autenticação de usuários e o *Crashlytics* para rastreamento e registro de falhas foram incorporados ao *app*.

## 3.3 Detalhamento das Etapas

### 3.3.1 Pré-Processamento

O conjunto de dados utilizado neste trabalho tem sua origem proveniente do *Behavioral Risk Factor Surveillance System (BRFSS)* (CDC, 2015). Entretanto, devido ao fato de que os dados brutos originais não eram tratados e continham um grande volume de atributos irrelevantes para este estudo, foi necessário e mais vantajoso utilizar uma variação desta base de dados presente em (Alex Teboul, 2021). Nessa versão, os dados já se encontravam em um formato adequado para algoritmos de aprendizagem de máquina, e os atributos escolhidos tinham foco nos indicadores do Diabetes. O conjunto selecionado continha dados referentes a pessoas classificadas como: não diabéticas, pré-diabéticas e diabéticas, portanto como etapa inicial de pré-processamento, a coluna contendo informações referentes aos diabéticos foi excluída, para que o estudo fosse direcionado apenas para os casos de pré-diabetes. No total, restaram 253.680 linhas de dados, e dentre essas, 23.650 eram duplicados. Consequentemente, esses dados duplicados foram excluídos, assim como a maior parte dos dados referentes a pessoas não diabéticas, pois havia apenas 4.629 dados pertencentes a pessoas pré-diabéticas, dessa maneira não era necessário um grande volume de dados apenas para uma classe, evitando assim um grande desbalanceamento entre as amostras e uma sobrecarga futura ao aplicar algoritmos de aprendizagem nesses dados.

### 3.3.2 Análise dos Dados

Com o término do pré-processamento, chegou a etapa de análise. Ao todo a base de dados continha 21 atributos, com variáveis categóricas e numéricas apresentados na Tabela 3.1. A intenção dessa etapa foi avaliar, por meio de gráficos de setores, histogramas e mapas de calor, como os atributos estavam distribuídos e o quanto correlacionados eles estavam com a classe do pré-diabetes. Isso auxiliou a definir quais deles seriam relevantes tanto para o modelo de classificação quanto para o aplicativo.

Atributos	Significado	Variável
HighBP	Pressão alta	Categórica
HighChol	Colesterol alto	Categórica
Age	Idade	Numérica
BMI	Índice de massa corporal	Numérica
Sex	Gênero	Categórica
GenHlth	Saúde no geral	Categórica
PhysActivity	Realizou atividade física nos últimos 30 dias	Categórica
Smoker	Consumiu mais de 100 cigarros durante a vida	Categórica
Fruits	Consome frutas uma ou mais vezes por dia	Categórica
Education	Grau de escolaridade	Categórica

Tabela 3.1: Tabela de atributos do conjunto de dados

Os outros atributos que não estão na tabela foram: CholCheck, Stroke, PhysActivity, DiffWalk, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, MentHlth, HeartDiseaseorAttack e Income.

### 3.3.3 Aprendizado de Máquina

Após realizar o pré-processamento e análise dos dados, restou a última etapa para finalmente gerar o modelo de classificação para o pré-diabetes<sup>1</sup>. No início, os dados foram separados em "features", que são os atributos escolhidos após todo o processo anterior e "target", que nada mais é do que as classes necessárias para rotular o paciente (não pré-diabético ou pré-diabético). Em seguida, foi necessário balancear os dados devido à diferença entre amostras das classes. Utilizou-se a técnica de balanceamento conhecida como *SMOTE*, que é definido como uma técnica estatística para aumentar de forma equilibrada o número de casos em um conjunto de dados. Ela gera novas instâncias a partir das amostras minoritárias existentes, combinando características do caso alvo com características de seus vizinhos (Azure Machine Learning, 2023).

Depois de equilibrar os dados, eles foram separados e padronizados em 25% para teste e 75% para treino, os algoritmos utilizados para gerar os modelos preditivos foram: *Logistic Regression*, *K-Neighbors*, *Decision Tree*, *Random Forest*, *SVC* e *Gradient Boosting*. Para avaliar os modelos e saber qual o melhor para o nosso objetivo, foram utilizadas várias métricas de avaliação como: *acurácia*, *precisão*, *revocação (recall)*, *score F1* e *curva ROC*, os resultados de cada modelo gerado são apresentados no capítulo 6.

<sup>1</sup>Pré-Processamento, Análise dos dados e Aprendizado de Máquina: Google Colab notebook

### 3.3.4 Implementação da API

O objetivo da implementação de uma *API* para esse sistema, foi devido a necessidade de reduzir o tamanho do aplicativo e evitar utilizar ferramentas não nativas do *Flutter* para integrar o modelo preditivo ao aplicativo, prevenindo possíveis falhas que bibliotecas não reconhecidas pela Google poderiam oferecer. A implementação da *API*<sup>2</sup> foi realizada de maneira ágil, pois o código do modelo de classificação foi o mesmo já descrito nas etapas anteriores, houve a apenas a necessidade de remover linhas de código desnecessárias para esse objetivo e adaptar para se utilizar o framework *FastAPI* definindo seus "endpoints" e criando um container virtual com *Docker* para facilitar a execução da aplicação e sua comunicação com o aplicativo.

### 3.3.5 Desenvolvimento do Aplicativo

O aplicativo já possuía uma versão base implementada, porém com a necessidade de adicionar novas funcionalidades e facilitar o desenvolvimento, foi necessário reimplementá-lo em uma nova tecnologia. O *Flutter* foi escolhido, pois possui a capacidade de gerar com o mesmo código versões para diversas plataformas, como: *Android*, *IOS* e *Web*, proporcionando um alcance maior de público para o aplicativo<sup>3</sup>. Por conta da mudança de tecnologia e reimplementação, foi essencial reestruturá-lo em relação ao padrão de arquitetura, gerência de estado e padrão de projeto.

#### Padrão de Arquitetura

Para a ideia inicial de padrão de arquitetura do *app* foi escolhido o *Model-View-Controller (MVC)*, ele é comumente utilizado para implementar interfaces de usuário, dados e lógica de controle, enfatizando a separação entre a lógica de negócios e a exibição do aplicativo (MDN Web Docs, 2023).

- Camada Model: Determina quais dados o aplicativo deve englobar em seu escopo;
- Camada View: É definido como os dados do aplicativo devem ser apresentados na tela;
- Camada Controller: Responsável por conter a lógica que atualiza tanto o *model* quanto o *view* de acordo com a interação do usuário no aplicativo.

Foi necessário a adição de mais uma camada na arquitetura do *app* para complementar esse padrão. Tal camada é responsável por comportar a gerência de estado do aplicativo. Ele trabalha de forma conjunta com a camada *controller*, notificando a *view* quando há alterações de estado na aplicação, atualizando assim os componentes das telas.

---

<sup>2</sup>API: repositório GitHub

<sup>3</sup>Aplicativo: repositório GitHub

A Figura 3.2 ilustra a divisão da estrutura geral do aplicativo. É possível observar que a camada de gerência de estado (*state management*) atuando como intermediária entre o view e controller, enquanto o model e as comunicações com a *API* do modelo de classificação e *Firebase* têm suas ações coordenadas pelo controller.

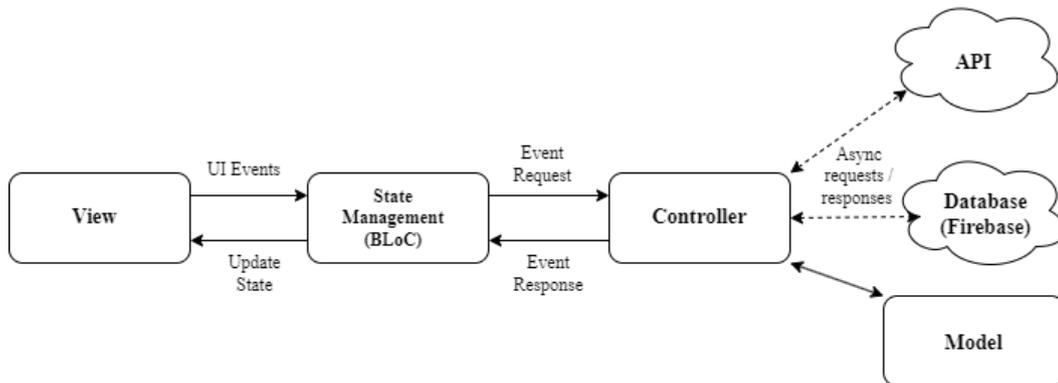


Figura 3.2: Padrão de arquitetura do app. Adaptado de (Vahid Dejwakh, 2021)

### Gerência de Estado

Para esta camada da arquitetura foi escolhido um padrão de gerenciamento de estado bem adaptado ao framework *Flutter*, chamado *BLoC (Business Logic Component)*. Ele foi projetado para separar a lógica de negócios da interface de usuário do aplicativo, tornando o código mais organizado e facilitando possíveis expansões e testes no *app*. Para sua implementação, são necessários três componentes:

- **Bloc:** Este componente contém a lógica de negócios e gerencia o estado da aplicação. Por meio do recebimento de eventos, ele produz estados como saída;
- **Event:** Os eventos são ações que ocorrem na interface de usuário ou em outras partes do aplicativo. Eles são responsáveis por provocar as mudanças de estado;
- **State:** Os estados representam as diferentes condições em que o aplicativo pode apresentar em resposta aos eventos.

Toda autenticação de usuário e atualização de dados do aplicativo, está sob responsabilidade dessa camada de gerência, onde a cada mudança de estado são emitidas notificações na tela.

A Figura 3.3 representa como a comunicação entre a camada de visualização (*view*) e o gerenciamento de estado (*BLoC*) é realizada. O usuário (*actor*) interage com a tela do aplicativo e, em seguida, um evento (*event*) é emitido. Nesse evento, pode haver uma solicitação de login ou de alteração de dados, por exemplo. Essa solicitação chega ao gerenciador de estados, que se comunica com a camada de controle (*controller*). Após

receber a resposta do controle, um novo estado (state) é emitido para o aplicativo, atualizando a tela e mostrando ao usuário se a ação foi bem-sucedida ou se ocorreu algum erro durante o processo.

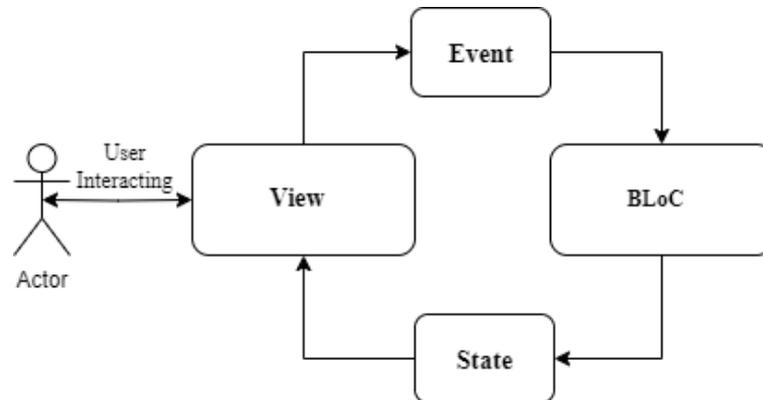


Figura 3.3: Padrão BLoC. Adaptado de (Cap Sistema, 2020)

### Padrão de Projeto

Na escolha do padrão de projeto, optou-se pelo uso do *Singleton*. Especificamente, esse padrão foi aplicado às classes que lidam com o *database* e a *API*, pois estas são frequentemente utilizadas por diversas partes do sistema. A ideia por trás desse padrão está na condição em que apenas uma instância da classe pode existir e esta será utilizada por toda aplicação, estabelecendo assim um ponto central de acesso a classe (Lutti, 2018). Isso evita a criação de múltiplas instâncias durante a execução do sistema, reduzindo o consumo desnecessário de memória que seria ocasionado por essa condição.

# Capítulo 4

## Funcionalidades do Sistema

### 4.1 Autenticação de Usuário

O aplicativo possui uma autenticação de usuário totalmente integrada ao *Firebase Authentication*, permitindo que os usuários realizem o login diretamente com uma conta *Gmail* já cadastrada no smartphone ou registrem uma conta usando outro provedor de serviço de e-mail, como o *Outlook*. Por estar integrado ao *Firebase*, o *app* está totalmente de acordo com as políticas de privacidade estabelecidas pelo *Google*, que é a proprietária da tecnologia, tornando todo o procedimento de login e recuperação de senha eficiente e seguro para os usuários (Figura 4.1).

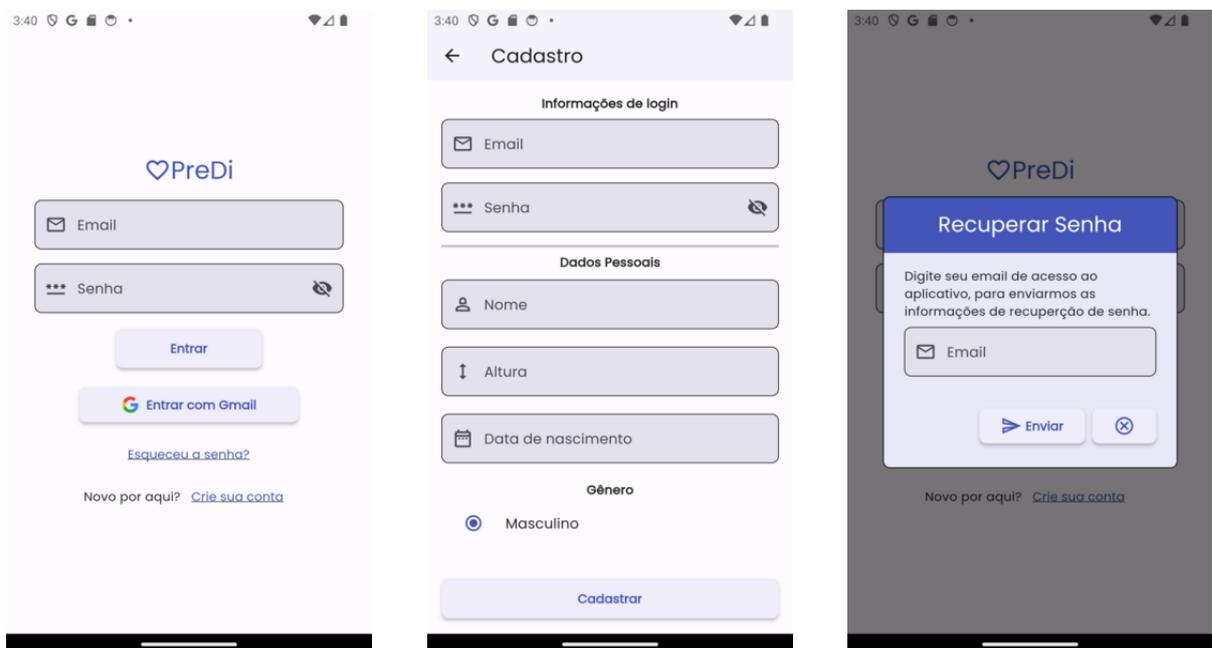


Figura 4.1: Telas de autenticação de usuário

## 4.2 Captação de Dados

Ao ser realizado o primeiro login, o aplicativo conduz um questionário com o objetivo de obter respostas referentes aos atributos selecionados durante fase de análise dos dados (Figura 4.2). Essas respostas são indispensáveis para que o sistema, posteriormente, comunique-se com a *API* e gere o relatório do paciente, exibindo detalhes sobre possíveis condições críticas de saúde associadas ao pré-diabetes. O sistema base do aplicativo já incluía a captação de resultados de exames *Hemoglobina Glicada*, *Glicose em Jejum*, *Glicose após 75g* (Figura 4.3), além de informações relacionadas ao peso e à circunferência abdominal (Figura 4.4), que são suficientes para indicar se o paciente possui o Diabetes tipo 2.

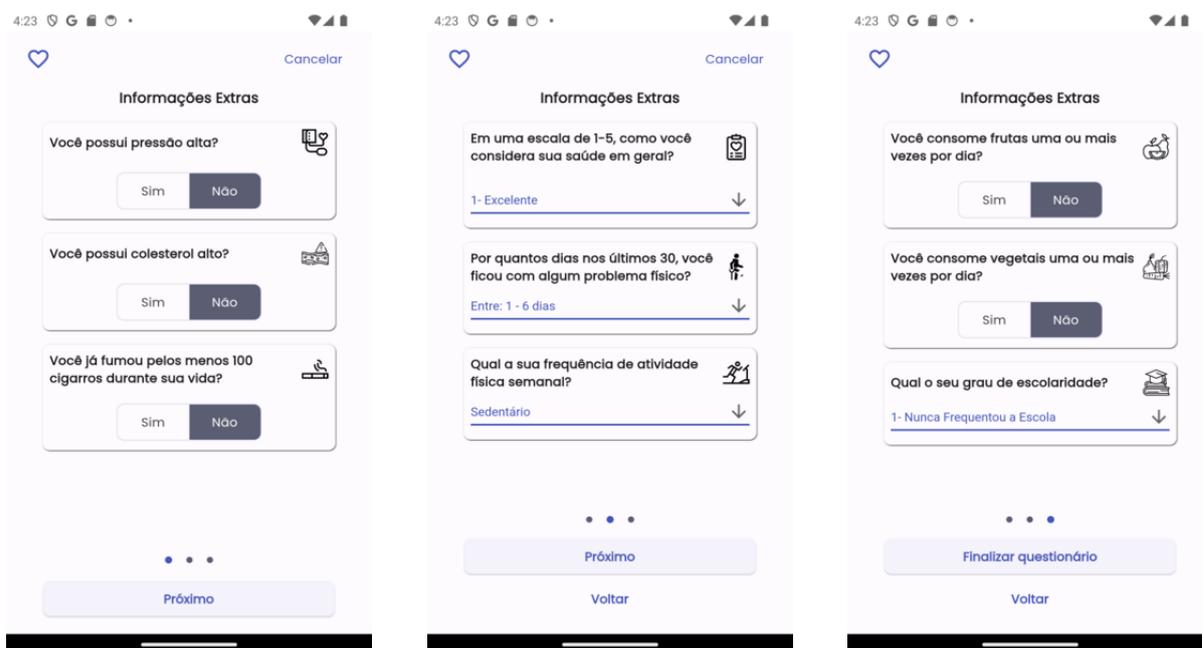


Figura 4.2: Telas do questionário

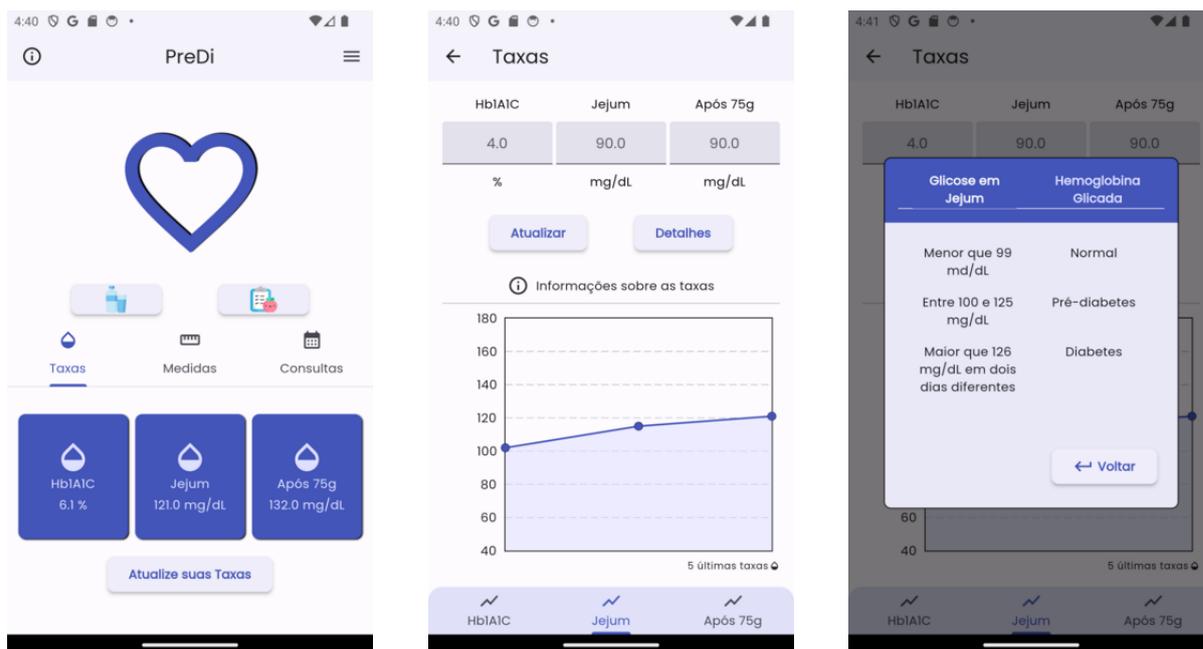


Figura 4.3: Telas de taxas

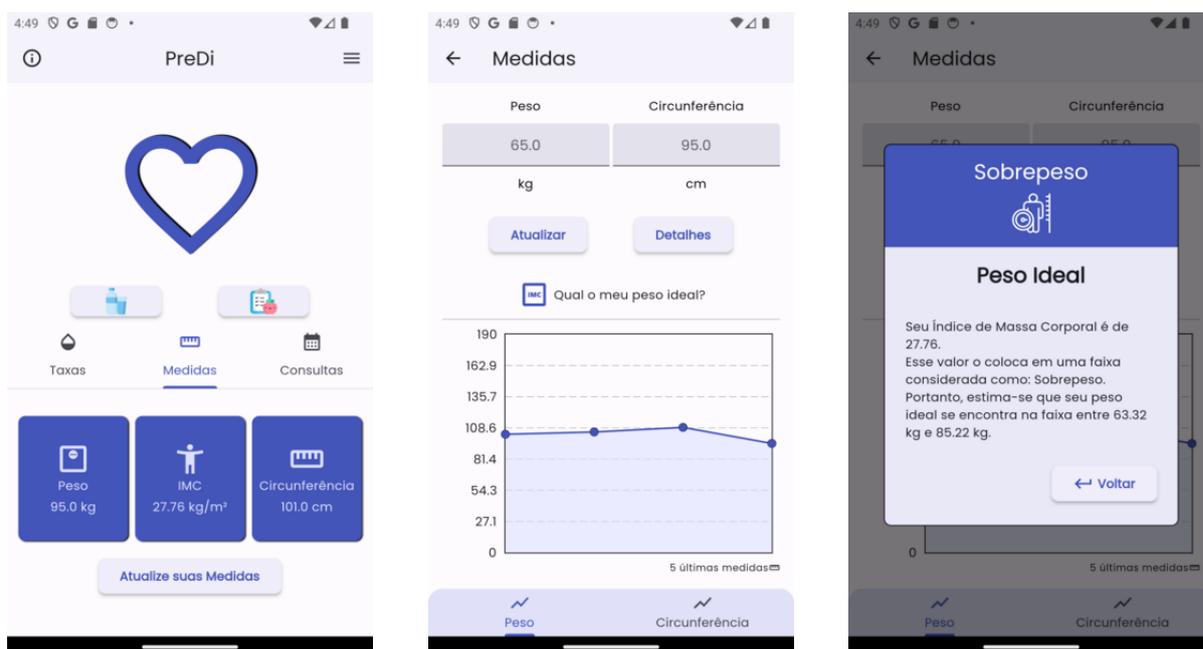


Figura 4.4: Telas de medidas

### 4.3 Agendamento de Consultas

O aplicativo possui um sistema para agendamento de consultas, que atua como lembrete de horário, data, local e tipo de exame que o paciente tem a realizar. O usuário registra suas próximas consultas no *app* e, à medida que as datas se aproximam, são enviadas notificações no smartphone para relembrar os compromissos agendados (Figura 4.5).

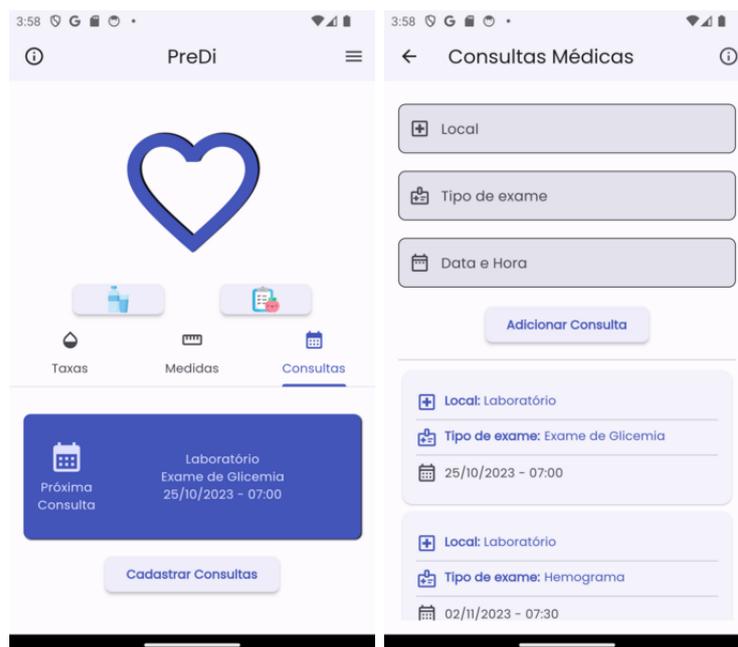


Figura 4.5: Telas de consultas

## 4.4 Módulo de Dicas

Com base nas informações coletadas pelo *app*, o sistema pode regularmente fornecer algumas dicas e estabelecer metas diárias visando a manutenção do equilíbrio na saúde do paciente, contribuindo para o controle do Diabetes ou dificultando desenvolver a doença (Figura 4.6).

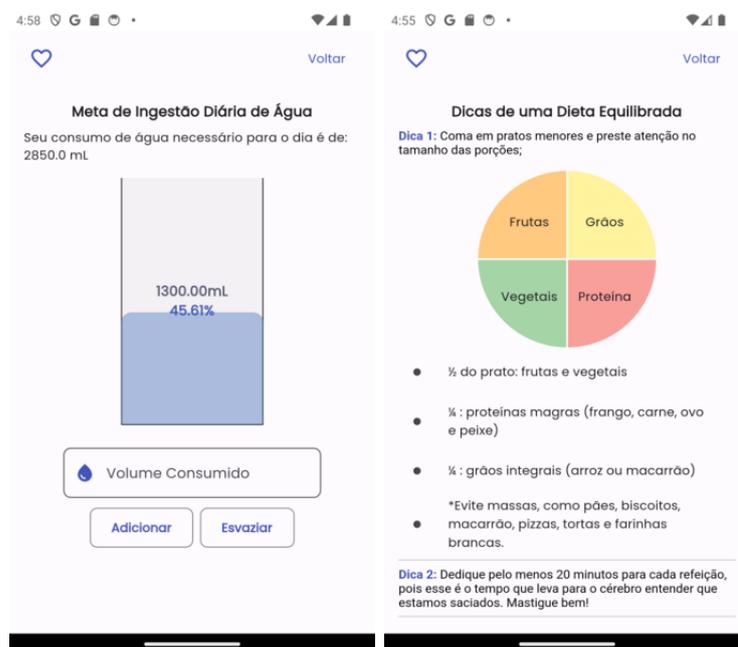


Figura 4.6: Telas de dicas

## 4.5 Recomendações de Saúde

Ao pressionar o ícone do aplicativo no centro da tela inicial (Figura 4.7), a comunicação com a *API* é iniciada e os dados previamente coletados são enviados. Após o término dessa comunicação, o *app* é capaz de gerar um relatório contendo informações sobre a situação atual do paciente em relação ao pré-diabetes, que estão de acordo com os dados fornecidos anteriormente (Figura 4.8).

A condição de saúde apresentada no relatório é criada por meio de uma árvore de decisão implementada no código do aplicativo, que leva em consideração a resposta obtida através da *API*, que indica se há indícios de pré-diabetes, assim como os dados coletados no questionário, taxas e medidas. Caso o relatório assinale que o paciente apresenta pré-diabetes ou uma condição normal, porém com indícios de progressão para pré-diabetes, será criada uma seção de detalhes. Nessa seção, serão fornecidas recomendações para melhorar a condição de saúde do paciente, que não esteja dentro das condições normais e que influencie diretamente no possível desenvolvimento ou agravamento da doença.

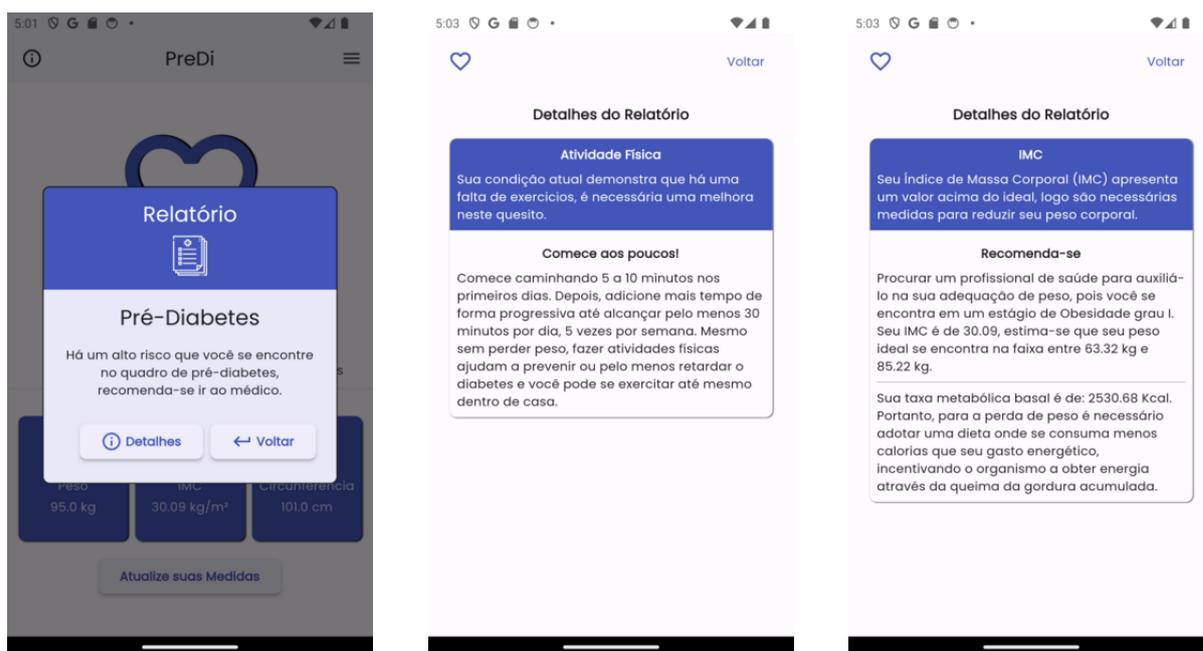


Figura 4.7: Telas de recomendações 1

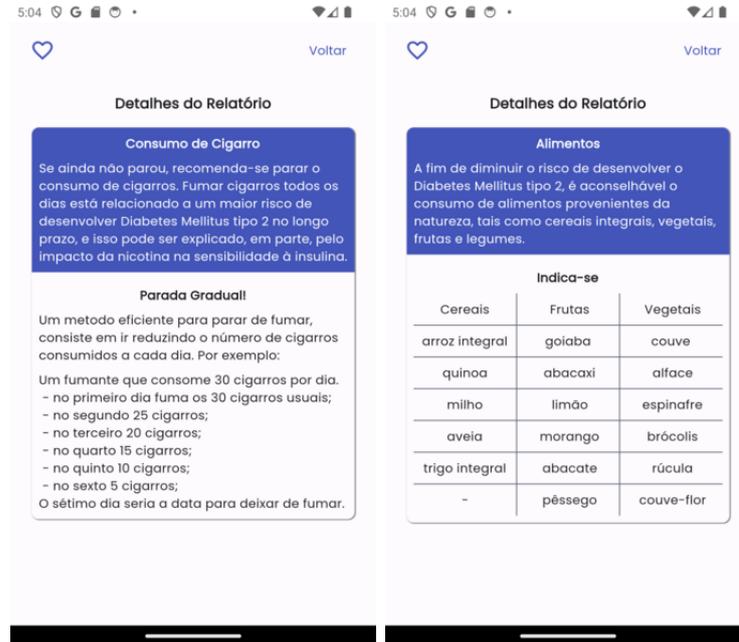


Figura 4.8: Telas de recomendações 2

# Capítulo 5

## Resultados

Durante o período de análise dos dados, foram gerados diversos gráficos com base nos 21 atributos presentes no conjunto (Figura 5.1 e Figura 5.2). O objetivo geral dessa análise era avaliar quais atributos apresentavam correlação relevante entre si e também correlação com a variável "target" (não pré-diabético ou pré-diabético). Os resultados destes gráficos foram fundamentais para identificar quais atributos teriam menor impacto no modelo de classificação e também não seriam adequados para inclusão no aplicativo.

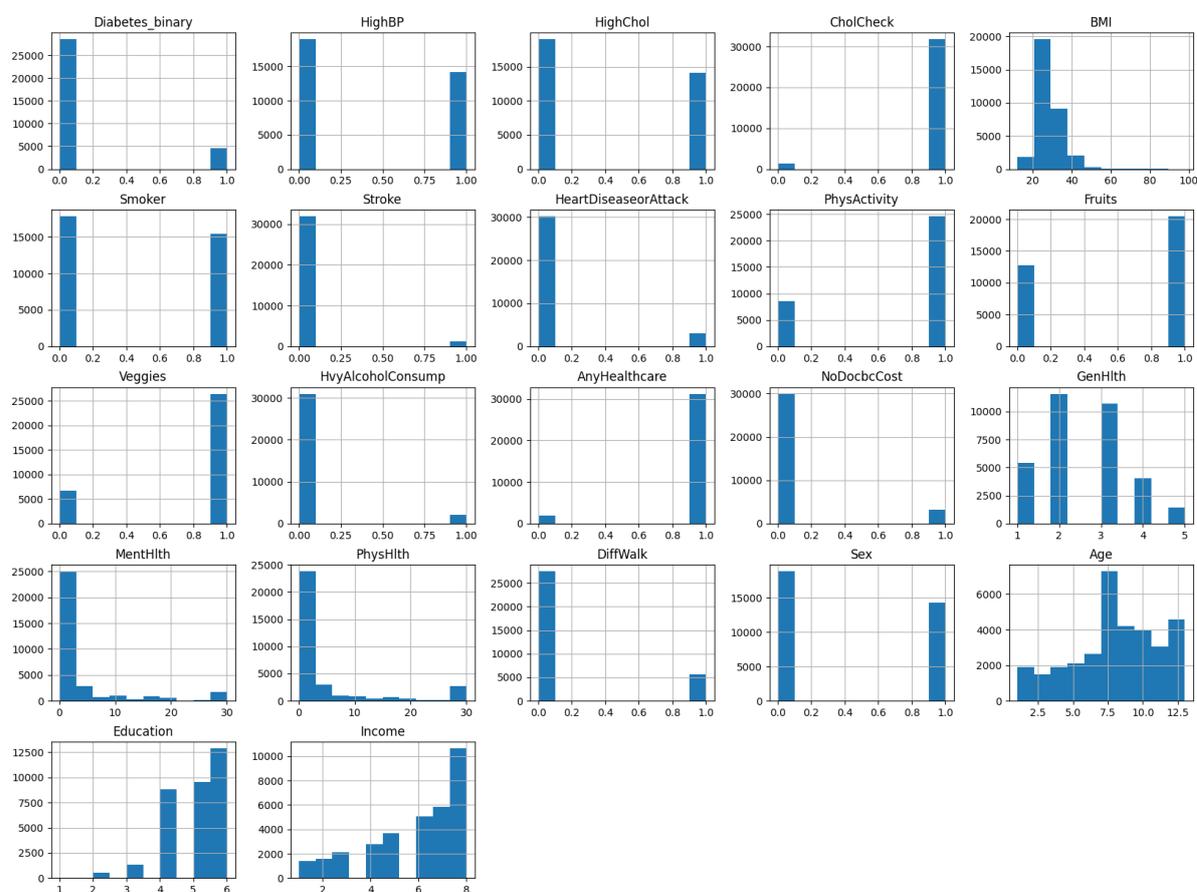


Figura 5.1: Histograma da base de dados ([link da imagem](#))

Com base na Figura 5.2, foi possível obter as primeiras percepções sobre o conjunto de dados. Foi observada uma correlação significativamente forte entre alguns atributos, o que indica que alguns deles poderiam representar redundância, ou seja, forneceriam praticamente a mesma informação. Os atributos "*Education*" e "*Income*" foram identificados como tais, em que o primeiro representa o nível de escolaridade do paciente e o segundo a escala de renda familiar. Assim, foi analisado qual dos atributos poderia ser mais benéfico ao *app*, pois não seria necessário manter ambos para gerar o modelo e isso teria um impacto mínimo nas métricas de avaliação. Como resultado, optou-se por se manter o atributo "*Education*", visto que seria mais simples para o aplicativo coletar as informações relacionadas a esse atributo.

A dupla seguinte de atributos que apresentava uma correlação significativa incluía "*PhysHlth*" e "*DiffWalk*". O primeiro indicava quantos dias o paciente teve problemas físicos ou lesões no último mês, enquanto o segundo indicava se o paciente tinha dificuldades para andar. Entre esses dois, escolheu-se "*PhysHlth*", pois as informações sobre o paciente obtidas por esse atributo eram mais úteis para o módulo de dicas do *app*. Além disso, "*PhysHlth*" também tinha uma alta correlação com "*GenHlth*". Após testes posteriores, ficou detectado que ambos os atributos contribuía bastante para a eficiência do modelo de classificação, então ambos foram mantidos. Portanto, após a análise inicial deste mapa de calor com as correlações, o número de características úteis para o sistema caíram de 21 para 19.

Outras observações pertinentes estão relacionadas à correlação dos atributos com a variável "*target*". As correlações apresentaram índices baixos, no entanto, foi possível identificar quais atributos eram os mais relevantes para a criação de um modelo preditivo competente. Dessa forma, os atributos relevantes permaneceriam sempre na seleção para gerar o modelo. Os cinco que mais se destacaram foram: "*GenHlth*", "*HighBp*", "*HighChol*", "*BMI*" e "*Age*".

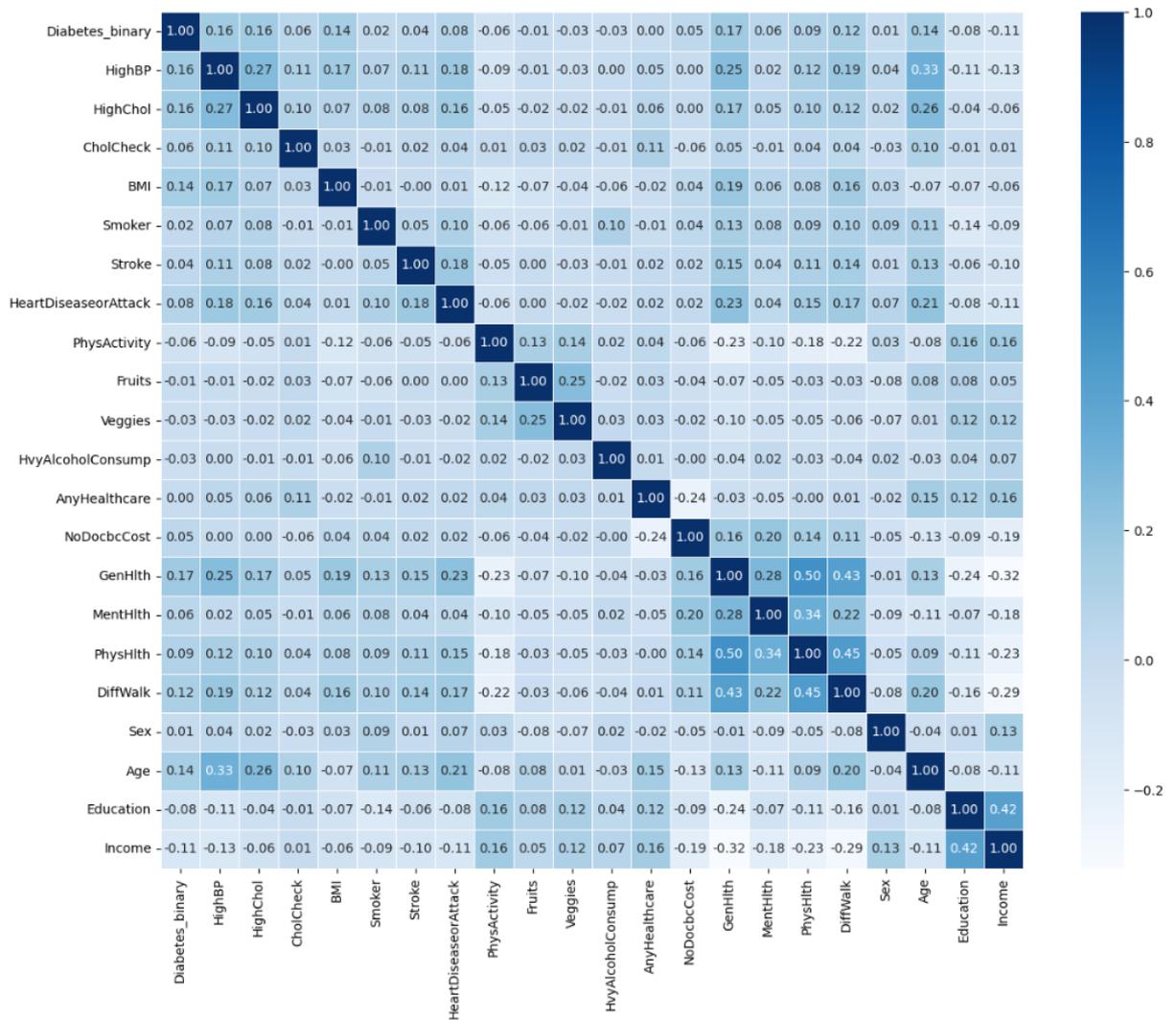


Figura 5.2: Correlação entre os atributos da base de dados (link da imagem)

Após a conclusão das fases de pré-processamento e análise dos dados, chegou o momento da etapa de aprendizado de máquina (Capítulo 3). Nesta etapa, foi feita a primeira tentativa de criar um modelo preditivo eficiente. Essa tentativa levou em consideração os 19 atributos e os resultados das métricas de avaliação para cada modelo e seus respectivos algoritmos, estão presentes na Tabela 5.1.

model	accuracy	precision	recall	f1	roc auc
Random Forest	0.858075	0.841145	0.883867	0.861977	0.858002
Decision Tree	0.803915	0.774886	0.858262	0.814446	0.803762
KNN	0.749825	0.718914	0.822723	0.767324	0.749619
SVM	0.727585	0.714624	0.760319	0.736764	0.727493
Gradient Boosting	0.713484	0.705598	0.735413	0.720197	0.713422
Logistic Regression	0.692928	0.697914	0.683364	0.690562	0.692955

Tabela 5.1: Tabela de métricas de avaliação do primeiro teste

Através das métricas de avaliação, é possível observar que o modelo que obteve melhores resultados foi aquele que utilizou o Random Forest como algoritmo. A partir desse modelo, foi possível criar o gráfico que exibe os atributos mais relevantes, conforme ilustrado na Figura 5.3.

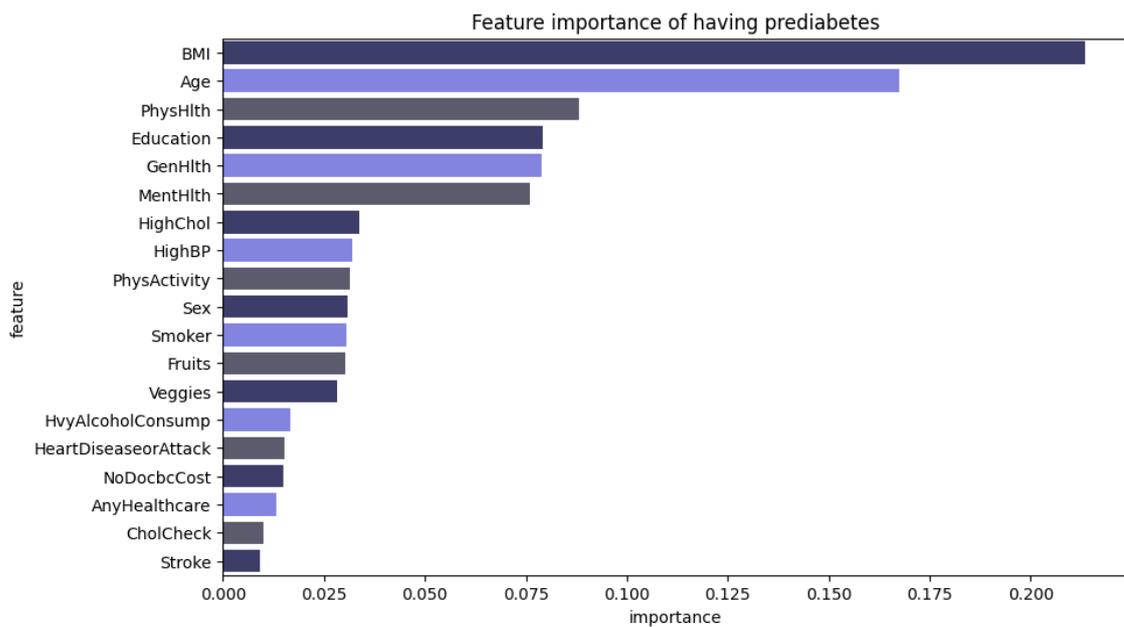


Figura 5.3: Nível de importância dos atributos para o modelo gerado (teste 1)

Na segunda tentativa de criar um modelo, foi removido do teste os atributos com importância abaixo de 0.025, uma vez que demonstraram ter pouca relevância para nosso objetivo. Um atributo de grande importância que também foi removido foi o *"MentHlth"*, pois o valor atribuído a ele indica o nível de saúde mental do paciente durante um determinado mês. No entanto, para o contexto geral do aplicativo, não seria interessante incluí-lo, já que ele envolve uma área muito ampla e não haveria uma forma eficiente de coletar esses dados. Logo, o número de atributos caiu de 19 para 12, e as seguintes métricas de avaliação foram obtidas (Tabela 5.2):

model	accuracy	precision	recall	f1	roc auc
Random Forest	0.853585	0.827805	0.893942	0.859603	0.853471
Decision Tree	0.808545	0.779126	0.862740	0.818804	0.808392
KNN	0.742318	0.706786	0.830698	0.763749	0.742069
SVM	0.696647	0.689539	0.718483	0.703714	0.696585
Gradient Boosting	0.688438	0.679253	0.717364	0.697788	0.688357
Logistic Regression	0.658903	0.663471	0.648804	0.656055	0.658931

Tabela 5.2: Tabela de métricas de avaliação do segundo teste

Percebe-se que nesta segunda tentativa, o modelo baseado em Random Forest continuou sendo o melhor, no entanto, a métrica de acurácia ("accuracy") apresentou uma queda de aproximadamente 0.52%. Essa redução ocorreu devido à remoção do atributo "MentHlth", mas o impacto para o modelo foi muito baixo.

A diminuição dos atributos ocasionou uma maior facilidade de integração com o aplicativo, já que as informações relacionadas aos atributos restantes podiam ser facilmente captadas pelo aplicativo. Além disso, o tempo gasto no processamento dos modelos diminuiu, pois haviam menos dados para se testar.

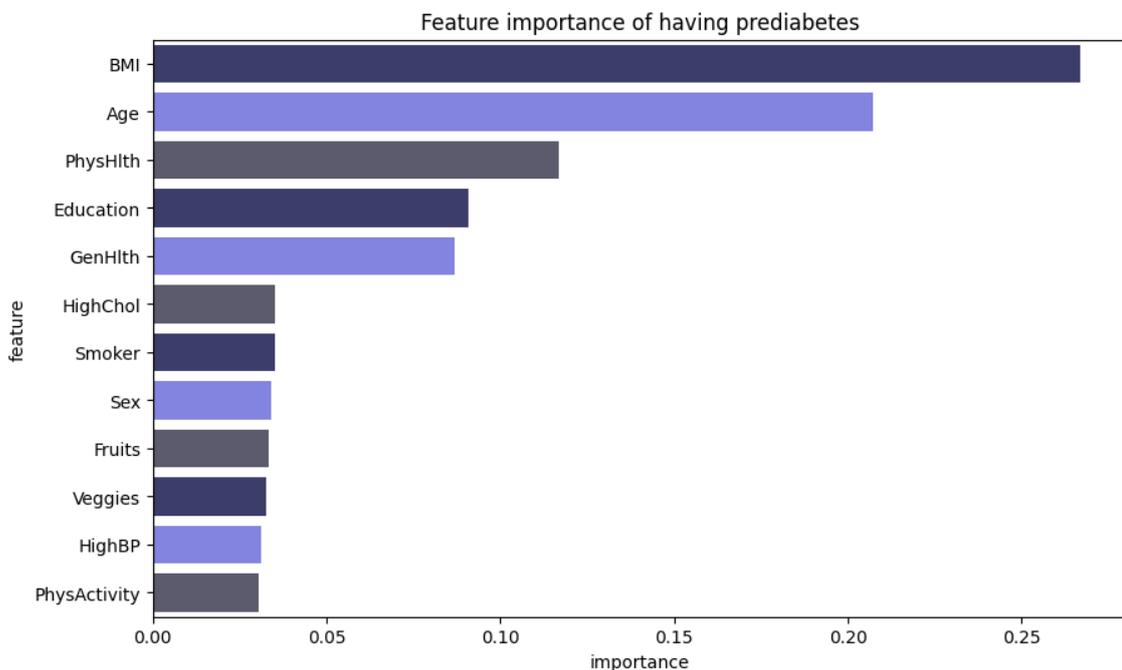


Figura 5.4: Nível de importância dos atributos para o modelo gerado (teste 2)

Na Figura 5.4 é ilustrada a lista final das 12 "features" mais relevantes para diagnosticar o pré-diabetes, de acordo com o estudo realizado na base de dados apresentada neste trabalho. Com o resultado obtido, foi possível determinar quais características do paciente são determinantes para indicar a presença da doença ou sinais de um eventual

desenvolvimento, auxiliando no monitoramento aprimorado dos pacientes. Portanto, o objetivo esperado de criar um sistema inteligente baseado em conhecimento para acompanhar pre-diabéticos pode ser alcançado.

# Capítulo 6

## Conclusão

Neste trabalho, foi apresentado um sistema inteligente baseado em conhecimento, cujo objetivo é realizar o acompanhamento de pacientes em situações de risco relacionadas ao diabetes e sua fase intermediária (pré-diabetes). Com base nos estudos conduzidos sobre o conjunto de dados apresentado anteriormente, foi possível desenvolver funcionalidades que fortaleceram a aplicação, permitindo assim, alcançar o propósito de avaliar, fornecer dicas e recomendações para monitorar e melhorar a saúde do paciente.

Em relação as limitações do sistema, o seu funcionamento passa a depender inteiramente da conexão de internet disponível no dispositivo móvel. Se em algum momento não houver conexão, a aplicação deixa de funcionar e a experiencia do usuário se torna prejudicada. Outra limitação está relacionada à quantidade de recomendações de saúde que o sistema pode gerar. Essa quantidade é bastante reduzida, tornando-as repetitivas, fato que pode levar o usuário a perder o interesse em continuar utilizando o aplicativo.

Para trabalhos futuros, pretende-se resolver as limitações mencionadas desenvolvendo uma versão do aplicativo que funcione mesmo sem conexão com a internet. Isso permitirá que o sistema utilize tanto armazenamento local quanto em nuvem (Firebase), que já está implementado. Em relação às recomendações, propõe-se realizar um estudo específico com profissionais de saúde para ampliar a gama de recomendações possíveis geradas pelo aplicativo. Por fim, é interessante trabalhar em uma possível integração do aplicativo com dispositivos capazes de medir a glicemia do paciente, recebendo esses dados e possibilitando um diagnóstico em tempo real de uma possível condição problemática do paciente.

# Bibliografia

- Alex Teboul (2021). Diabetes health indicators dataset. <https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?resource=download>.
- Atlas, I. D. (2021). *IDF Diabetes Atlas 2021*. Internacional Diabetes Federation, 10 edition.
- Azure Machine Learning (2023). Smote. Disponível em: <https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/smote?view=azureml-api-2>. Acesso em 19 de junho de 2023.
- Beulens, J., Rutters, F., Rydén, L., Schnell, O., Mellbin, L., Hart, H., and Vos, R. (2020). Risk and management of pre-diabetes. *European Journal of Preventive Cardiology*, 26(2<sub>suppl</sub>): 47 – 54.
- Cap Sistema (2020). Introdução ao padrão bloc. Disponível em: <https://capsistema.com.br/index.php/2020/10/21/introducao-ao-padrao-bloc/>. Acesso em 8 de julho de 2023.
- CDC (2015). Behavioral risk factor surveillance system. Disponível em: [https://www.cdc.gov/brfss/annual\\_data/annual\\_2015.html](https://www.cdc.gov/brfss/annual_data/annual_2015.html). Acesso em 20 de setembro de 2023.
- Dinh, A., Miertschin, S., Young, A., and Mohanty, S. D. (2019). A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. *BMC Medical Informatics and Decision Making*, 19(1).
- Geeks for Geeks (2023). What is machine learning? Disponível em: <https://www.geeksforgeeks.org/ml-machine-learning/>. Acesso em 20 de setembro de 2023.
- IBM (2023). What is machine learning? Disponível em: <https://www.ibm.com/topics/machine-learning>. Acesso em 20 de setembro de 2023.
- Internacional Diabetes Federation (2021). Brazil: Diabetes report 2000 — 2045. <https://diabetesatlas.org/data/en/country/27/br.htmlr>.

- Khan, R. M. M., Chua, Z. J. Y., Tan, J. C., Yang, Y., Liao, Z., and Zhao, Y. (2019). From pre-diabetes to diabetes: Diagnosis, treatments and translational research. *Medicina*, 55(9).
- Kriventsov, S., Lindsey, A., and Hayeri, A. (2020). The diabits app for smartphone-assisted predictive monitoring of glycemia in patients with diabetes: Retrospective observational study. *JMIR Diabetes*, 5(3):e18660.
- Lutti (2018). Singleton design pattern. Disponível em: <https://www.opus-software.com.br/insights/singleton-design-pattern/>. Acesso em 4 de julho de 2023.
- Marinho, T., Filho, W. F., Freire, G., Costa, M., da Silva, L., Álvaro C. Sobrinho, and Costa, E. (2019). Assistente inteligente para auxiliar na prevenção do diabetes tipo 2. In *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 121–126, Porto Alegre, RS, Brasil. SBC.
- MDN Web Docs (2023). Mvc. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. Acesso em 4 de julho de 2023.
- Portal Telemedicina (2023). O que é mhealth? Disponível em: <https://portaltelemedicina.com.br/o-que-e-mhealth>. Acesso em 05 de Julho de 2023.
- PPGC UFPel (2023). Sistemas inteligentes. Disponível em: <https://wp.ufpel.edu.br/computacao/ppgc/sistemas-inteligentes/>. Acesso em 21 de Julho de 2023.
- Sociedade Brasileira de Diabetes (2023). Diabetes. <https://diabetes.org.br>.
- Vahid Dejwakh (2021). Understanding the model-view-controller (mvc) pattern. Disponível em: <https://vahid.blog/post/2021-04-16-understanding-the-model-view-controller-mvc-pattern/>. Acesso em 4 de julho de 2023.
- Waki, K., Aizawa, K., Kato, S., Fujita, H., Lee, H., Kobayashi, H., Ogawa, M., Mouri, K., Kadowaki, T., and Ohe, K. (2015). Diabetics with a multimedia food recording tool, foodlog: Smartphone-based self-management for type 2 diabetes. *Journal of diabetes science and technology*, 9(3):534—540.
- Yamaguchi, S., Waki, K., Nannya, Y., Nangaku, M., Kadowaki, T., and Ohe, K. (2019). Usage patterns of gluconote, a self-management smartphone app, based on researchkit for patients with type 2 diabetes and prediabetes. *JMIR Mhealth Uhealth*, 7(4):e13204.