

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA BIOMÉDICA

DANILLO RODRIGUES DA SILVA

**MOGLIE: PROTÓTIPO DE APLICAÇÃO MOBILE PARA
AUTOGERENCIAMENTO DA DIABETES MELLITUS**

Uberlândia,
2021

DANILLO RODRIGUES DA SILVA

**MOGLIE: PROTÓTIPO DE APLICAÇÃO *MOBILE* PARA
AUTOGERENCIAMENTO DA DIABETES MELLITUS**

Trabalho de Conclusão de Curso apresentado à
Faculdade de Engenharia Elétrica da
Universidade Federal de Uberlândia como
requisito parcial para obtenção do título de
bacharel em Engenharia Biomédica.

Orientadora: Prof.^a Dra. Ana Cláudia Patrocínio

Uberlândia,

2021

Dedico este trabalho à minha família.

AGRADECIMENTOS

Agradeço aos meus pais, por batalharem tanto e investirem na minha educação, sem o apoio deles eu jamais poderia completar o ensino superior.

A minha irmã Danielle, por ser minha companheira na vida e nas jornadas.

Ao meu companheiro Eduardo, pelos 10 anos de amizade, pelos momentos que passamos juntos e por estar ao meu lado sempre.

As minhas colegas de turma Thaynara, Ana Flávia, Bruna e Mirella, por terem me apoiado nos momentos de dificuldade e me acompanhado durante a graduação.

Aos meus amigos João Victor e Carolina Beatriz, por terem feito os anos em Uberlândia mais alegres, por me darem apoio durante os altos e baixos da vida acadêmica e por serem uma segunda família.

Aos colegas do PET Engenharia Biomédica, por enriquecerem tanto a minha graduação.

A minha orientadora, Prof.^a Dra. Ana Claudia Patrocínio, pelo apoio, paciência e conhecimento.

“Educação é uma descoberta progressiva
de nossa própria ignorância.”
(Voltaire)

RESUMO

O Diabetes Mellitus (DM) é uma doença grave e crônica que acomete a milhões de pessoas ao redor do mundo. Essa patologia, quando não controlada adequadamente, pode acarretar no surgimento de inúmeras comorbidades e complicações. As soluções m-Health podem ser um aliado ao tratamento, a partir da continuação de comportamentos positivos contra a doença. O presente trabalho apresenta o desenvolvimento do protótipo de um aplicativo mobile para o autogerenciamento do diabetes mellitus, contemplando os principais recursos presentes nos aplicativos existentes no mercado, com uma abordagem simples e acessível, focada no autocuidado e na usabilidade. O Rational Unified Process (RUP) foi escolhido como metodologia para implementar a solução, sendo assim, as atividades metodológicas foram organizadas dentro das fases do RUP: Concepção, Elaboração, Construção e Transição. Foi apresentado um comparativo entre os principais aplicativos usados no gerenciamento e monitoramento à Diabetes Mellitus e uma solução que engloba o maior número de funcionalidades foi desenvolvida. Todos os casos de uso definidos foram divididos em 4 iterações, e ao final de cada iteração um protótipo executável foi gerado. O trabalho focou no desenvolvimento da primeira e da segunda iteração. Os conceitos de engenharia de software e o paradigma de orientação a objetos foram explorados e utilizados na implementação do aplicativo MOGLIE, nativo para dispositivos moveis com o sistema operacional Android. Desenvolvido e testado no ambiente de desenvolvimento integrado Android Studio, o aplicativo apresenta funções como registrar glicemia, medicamentos, refeições, exercícios, criar e gerenciar lembretes, consultar índice e carga glicêmica dos alimentos, entre outros. Foram utilizados os bancos de dados SQLite para armazenamento interno no dispositivo e o *Firebase* Realtime Database para armazenamento de um backup na nuvem.

Palavras-chave: Aplicativo, Diabetes Mellitus, Desenvolvimento de software mobile, protótipo.

ABSTRACT

Diabetes Mellitus (DM) is a serious and chronic disease that affects millions of people around the world. This pathology, when left untreated, can lead to the emergence of comorbidities and complications. m-Health solutions can be an ally in treatment, based on the continuation of positive behaviors against the disease. This paper presents the prototype development of a mobile application for self-management of diabetes mellitus, contemplating the main features present in existing applications on the market, with a simple and accessible approach, focused on self-care and usability. The Rational Unified Process (RUP) was chosen as the methodology to implement the solution, thus, how methodological activities were organized within the RUP phases: Conception, Elaboration, Construction and Transition. A comparison was presented between the main applications used in the management and monitoring of Diabetes Mellitus and a solution that encompasses the greatest number of features was developed. All defined use cases were divided into 4 iterations, and at the end of each iteration an executable prototype was generated. The work focused on the development of the first and second iterations. The concepts of software engineering and object-oriented paradigm were explored and used in the implementation of the MOGLIE application, native to motion devices with the Android operating system. Developed and tested in the Android Studio integrated development environment, the application features functions such as recording blood glucose, medications, meals, exercises, creating and managing reminders, consulting the index and glycemic load of food, among others. SQLite databases were used for internal storage on the device and Firebase Realtime Database for storage of a backup in the cloud.

Keywords: Application, Diabetes Mellitus, Mobile Software Development, Prototype.

LISTA DE ILUSTRAÇÕES

Figura 1: Número de pessoas com diabetes em diferentes regiões do mundo em 2019 e projeção para 2045	14
Figura 2: Número de mortes devido a diabetes em adultos (20-79 anos) por idade e sexo em 2019:.....	17
Figura 3: Capturas de tela do aplicativo “MySugr: Diário da Diabetes”.....	23
Figura 4: Capturas de tela do aplicativo “Diabetes: M”	23
Figura 5: Capturas de tela do aplicativo “SocialDiabetes”	24
Figura 6: Capturas de tela do aplicativo “Dieta para diabéticos”.....	25
Figura 7: Capturas de tela do aplicativo “Índice e Carga Glicêmicas: alimentos para diabetes”	26
Figura 8: Capturas de tela do aplicativo “Diabete – Diário Glucose”.....	26
Figura 9: Exemplo de diagrama de caso de uso	33
Figura 10: Exemplo de um Diagrama de Atividades Simples.	34
Figura 11: Uso de Diagrama de Atividades para Representar Casos de Uso.	35
Figura 12: Exemplos de Diagramas de Classes.....	36
Figura 13: Exemplo de diagrama de sequência.	37
Figura 14: Fases do RUP.	39
Figura 15: Exemplo de modelagem de banco de dados para uma loja.....	41
Figura 16: Comparação entre o banco de dados NoSQL e SQL.....	42
Figura 17: Diagrama de casos de uso do aplicativo.	47
Figura 18: Diagrama de atividades para o caso de uso “Criar Conta – Fazer Login”.....	50
Figura 19: Tela inicial do aplicativo (A), tela de cadastro (B), tela de login (C) e menu de opções (D).	51
Figura 20: Diagrama de atividades para os casos de uso "Registrar Glicemia" e "Gerar Gráfico do nível de glicemia x data".	52
Figura 21: tela “Meu Diário” (A), tela de registro de Medicamento (B), tela de registro de Glicemia (C) e tela de registro de Refeição (D)	53
Figura 22: tela da opção “Gráfico” (A), gráfico com zoom (B), diário com registros de glicemia (C) tela de registrar glicemia (D).....	54
Figura 23: Diagrama de Classes das classes que envolvem cadastro de usuário, login e acesso ao menu do aplicativo.	55

Figura 24: Diagrama de classes das classes que envolvem a funcionalidade “Meu Diário” do aplicativo	56
Figura 25: Diagrama de classes das classes que envolvem a funcionalidade “GRÁFICO” do aplicativo	57
Figura 26: Novo menu de opções (A), continuação do menu de opções (B), tela do “Meu Diário” com as novas opções “Exercício” e “Outros” (C) tela de registro de exercícios (D)..	59
Figura 27: Exemplo de anotações no diário (A), tela de feedback a um nível normal de glicose (B), tela de feedback a um nível baixo de glicose (C) e tela de feedback a um nível alto de glicose (D).	60
Figura 28: tela da opção “Nutrição” (A), resultado da busca (B), detalhes do alimento (C) e alteração do alimento (D)	61
Figura 29: Alimentos filtrados pelo ícone verde (A), filtros disponibilizados ao clicar no botão “Filtro” (B), alimentos filtrados (C) e tela informativa sobre IG e CG (D).....	62
Figura 30: Diagrama de atividades para a opção Nutrição do menu.....	63
Figura 31: tela “Gerenciar Lembretes” (A), tela “Novo lembrete” (B), tela “Definir data” (C) e tela Definir horário (D).....	64
Figura 32: Definir intervalo de repetição do lembrete (A), visão geral do lembrete preenchido (B), lista de lembretes criados (C) e notificação do lembrete na tela do celular (D).....	65
Figura 33: Formulário do perfil (A) e formulário do perfil preenchido (B)	66
Figura 34: Modelo de dados do aplicativo MOGLIE.....	67
Figura 35: Estrutura do armazenamento dos dados na nuvem.....	68
Figura 36: Usuários cadastrados no aplicativo MOGLIE.....	68
Figura 37: Diagrama Entidade-Relacionamento do Banco de dados do aplicativo.	69
Figura 38: Diagramas de sequência das funcionalidades Login (A) e Gráfico (B).....	71
Figura 39: Diagrama de sequência da funcionalidade "Meu Diário".....	72
Figura 40: Gráfico de monitoramento de permissões, negações e erros do <i>Firebase Realtime Database</i>	73

LISTA DE QUADROS

Quadro 1: Taxa de mortalidade por diabetes (a cada 100 mil habitantes), por macrorregião geográfica brasileira, segundo a faixa etária, no ano de 2017.....	18
Quadro 2: Comparativo entre funcionalidades dos aplicativos analisados	27
Quadro 3: Atividades metodológicas e tarefas para o desenvolvimento do aplicativo.....	44
Quadro 4: Requisitos funcionais do aplicativo.	45
Quadro 5: Requisitos não-funcionais do aplicativo.	45
Quadro 6: Principais atores do aplicativo.	46
Quadro 7: Iterações do aplicativo desenvolvido	48

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVO GERAL	13
1.1.1 Objetivos Específicos	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 DIABETES MELLITUS	14
2.1.1 Comorbidades e Complicações	16
2.1.2 Autocuidado	18
2.2 APLICATIVOS M-HEALTH.....	20
2.2.1 Uso de dispositivos móveis no Brasil	21
2.2.2 Aplicativos para autocuidado e gerenciamento do Diabetes Mellitus.....	22
2.2.2.1 <i>MySugr: Diário da Diabetes</i>	22
2.2.2.2 <i>Diabetes:M</i>	23
2.2.2.3 <i>SocialDiabetes</i>	24
2.2.2.4 <i>Dieta para Diabéticos</i>	25
2.2.2.5 <i>Índice e Carga Glicêmicas: alimentos para diabetes</i>	25
2.2.2.6 <i>Diabete – diário glucose</i>	26
2.3 DESENVOLVIMENTO DE SOFTWARE.....	28
2.3.1 Aplicações móveis nativas	28
2.3.2 Levantamento e Análise de Requisitos.....	29
2.3.3 Paradigma da Orientação a Objetos	30
2.3.4 Diagramas UML	31
2.3.4.1 <i>Diagrama de casos de uso</i>	32
2.3.4.2 <i>Diagrama de atividades</i>	33
2.3.4.3 <i>Diagrama de classes</i>	36
2.3.4.4 <i>Diagrama de sequência</i>	37
2.3.5 RUP - Rational Unified Process	37
2.4 BANCO DE DADOS	39
2.4.1 SQLite	42
2.4.2 Firebase Realtime Database	43
3 METODOLOGIA	44
3.1 VISÃO GERAL DO PROTÓTIPO	44
3.2 LEVANTAMENTO DE REQUISITOS	45

3.3	ANÁLISE E PROJETO DE ARQUITETURA	46
3.4	DESENVOLVIMENTO, TESTES, IMPLANTAÇÃO E PUBLICAÇÃO	49
4	RESULTADOS.....	50
4.1	1º ITERAÇÃO.....	50
4.2	2º ITERAÇÃO.....	58
4.3	PERSISTÊNCIA DE DADOS.....	66
4.4	DESEMPENHO DO APLICATIVO	70
5	DISCUSSÃO.....	74
6	CONCLUSÃO	79
7	REFERÊNCIAS	80

1 INTRODUÇÃO

O Diabetes Mellitus (DM) é uma doença grave e crônica que acomete a milhões de pessoas ao redor do mundo. De acordo com a International Diabetes Federation (IDF), aproximadamente 425 milhões de adultos no mundo possuíam Diabetes em 2017, onde 327 milhões estão na faixa dos 20 à 64 anos e 98 milhões de pessoas na faixa dos 65 à 79 anos. O Brasil é quinto maior país do mundo com mais pessoas diagnosticadas com diabetes, são 16,8 milhões de indivíduos acometidos com a doença (IDF, 2017).

Essa patologia, quando não controlada adequadamente, pode acarretar o surgimento de inúmeras comorbidades e complicações: desenvolvimento de doenças cardiovasculares, renais, infecções, neuropatias e amputações; incapacidade de realizar atividades diárias e laborais; gastos excessivos em saúde; hospitalizações frequentes; aposentadorias precoces e significativa redução da qualidade de vida (SCHAAN; HARZHEIM; GUS, 2004).

Apesar do diabetes ser uma doença crônica, é possível que os diabéticos tenham uma vida longa e de qualidade. O tratamento e o controle do diabetes são complexos e envolvem mudanças no estilo de vida do paciente, principalmente relacionadas a cuidados com a automonitoração da glicemia, prática regular de atividade física, administração de medicamentos (antidiabéticos orais e/ou insulina) e a adoção de uma alimentação saudável (AMERICAN DIABETES ASSOCIATION, 2015).

Para cuidar de doenças crônicas como o Diabetes, é fundamental adotar um modelo no qual os pacientes possam ser fortalecidos por meio do autocuidado (HSU et al., 2016). A tecnologia pode ser uma forte aliada para o gerenciamento do DM, afinal os aplicativos móveis para a saúde (*m-Health*) têm desempenhado um papel central para treinamento, autogerenciamento, monitorização remota e coleta de dados de tratamento de doenças crônicas (DEBON, 2019).

As soluções *m-Health* podem ser um aliado ao tratamento, a partir da continuação de comportamentos positivos contra a doença. A ferramenta é importante para a autopromoção da saúde, redução de gastos econômicos e acompanham o manejo nas doenças crônicas, monitoramento remoto, apoio psicológico e a tomada de decisão (FONSECA DE OLIVEIRA; ALENCAR, 2017; AZEVEDO et al., 2018).

Nesse contexto, a combinação de recursos para monitorar a dieta, o peso corporal, a pressão arterial, o histórico dos níveis de glicose no sangue, a interação medicamentosa, atividade física e o modo como o paciente está se sentindo (humor, sono, entre outros) podem colaborar

significativamente na conscientização das pessoas quanto as mudanças comportamentais, aumentar a adesão ao tratamento, incentivar a prática de hábitos saudáveis e proporcionar suporte no controle da diabetes (PAGLIALONGA; LUGO; SANTORO, 2018).

Com base no exposto e considerando a potencial contribuição para a sociedade e os diversos perfis de pessoas com diabetes, o presente trabalho apresenta o processo de desenvolvimento de um aplicativo *mobile* para o autogerenciamento do diabetes mellitus, contemplando os principais recursos existentes nos aplicativos disponíveis no mercado, com uma abordagem simples e acessível, focada no autocuidado e na usabilidade.

1.1 OBJETIVO GERAL

Desenvolver uma aplicação mobile para o autogerenciamento do diabetes mellitus, contemplando os principais recursos existentes nos aplicativos disponíveis no mercado, com uma abordagem simples e acessível, focada no autocuidado e na usabilidade.

1.1.1 Objetivos Específicos

- Apresentar um comparativo entre os principais aplicativos usados no gerenciamento e monitoramento do Diabetes Mellitus e propor uma solução que englobe o maior número de funcionalidades;
- Explorar os conceitos de engenharia de software e o paradigma de orientação a objetos, através da implementação de um aplicativo nativo para dispositivos móveis com o sistema operacional Android;
- Desenvolver um aplicativo com foco na usabilidade e autocuidado, capaz de auxiliar no monitoramento da saúde e incentivar a mudança no estilo de vida do usuário.

2 FUNDAMENTAÇÃO TEÓRICA

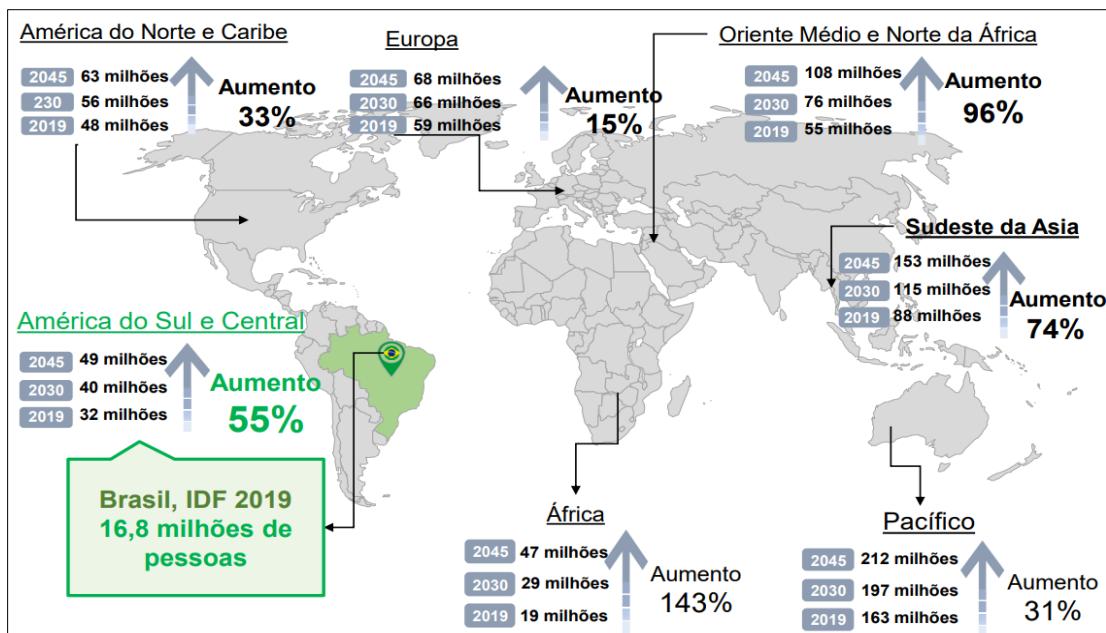
2.1 DIABETES MELLITUS

A Diabetes Mellitus (DM), popularmente chamada de diabetes, é uma condição crônica séria que ocorre quando há níveis elevados de glicose no sangue de uma pessoa incapaz de produzir ou usar o hormônio insulina (IDF, 2019).

A insulina é um hormônio essencial produzido no pâncreas. Ele permite que a glicose da corrente sanguínea entre nas células do corpo, onde é convertida em energia. A insulina também é essencial para o metabolismo de proteínas e gorduras. A falta de insulina, ou a incapacidade das células de responder a ela, leva a uma alta nos níveis de glicose no sangue (hiperglicemia), que é o indicador clínico da diabetes (IDF, 2019).

A Figura 1 exibe um infográfico sobre a quantidade de pessoas com diabetes em diferentes regiões do mundo e suas projeções para 2045. As informações foram extraídas do Atlas da Diabetes Mellitus publicado pela International Diabetes Federation (2019) e adaptadas pela Sociedade Brasileira de Diabetes.

Figura 1: Número de pessoas com diabetes em diferentes regiões do mundo em 2019 e projeção para 2045.



Fonte: *International Diabetes Federation* (2019).

Em 2017, existiam cerca de 425 milhões de pessoas diagnosticadas com diabetes (IDF, 2017). Em sua publicação mais recente, em 2019, a IDF atualizou número de diabéticos diagnosticados no mundo para 463 milhões, registrando um aumento de 38 milhões de casos comparado com os números de 2017. O Brasil é quinto maior país do mundo com mais pessoas diagnosticadas com diabetes, são 16,8 milhões de pessoas acometidas com a doença.

De acordo com Brandão e Pinheiro (2016), o diagnóstico do diabetes é feito através da detecção de hiperglicemia, ou seja, níveis elevados de glicose no sangue. Este diagnóstico é feito através de testes como: glicemia casual, glicemia de jejum, teste de tolerância à glicose (TTG) com sobrecarga de 75 g em duas horas e, em alguns casos, hemoglobina glicada (HbA1c).

Para a American Diabetes Association (2015), a Diabetes Mellitus é classificada de várias formas, as principais são:

- **Diabetes Mellitus tipo 1 (DM1):** cerca de 10% de todas as pessoas com diabetes têm diabetes tipo 1. O diabetes tipo 1 é causado por uma reação auto-imune em que o sistema de defesa do corpo ataca as células que produzem insulina. Como resultado, o corpo produz muito pouca ou nenhuma insulina. As causas exatas ainda não são conhecidas, mas estão ligadas a uma combinação de condições genéticas e ambientais. O diabetes tipo 1 pode afetar pessoas de qualquer idade, mas geralmente se desenvolve em crianças ou adultos jovens. Pessoas com diabetes tipo 1 precisam de injeções diárias de insulina para controlar os níveis de glicose no sangue.
- **Diabetes Mellitus tipo 2 (DM2):** é o tipo mais comum de diabetes, sendo responsável por cerca de 90% de todos os casos de diabetes. Geralmente é caracterizada por resistência à insulina. Os níveis de glicose no sangue continuam aumentando, liberando mais insulina, e isso pode eventualmente exaurir o pâncreas, resultando na produção de cada vez menos insulina pelo corpo, causando níveis de açúcar no sangue ainda mais elevados (hiperglicemia). O diabetes tipo 2 é mais comumente diagnosticado em adultos mais velhos, mas é observada em crianças, adolescentes e adultos jovens devido aos níveis crescentes de obesidade, sedentarismo e dieta inadequada.
- **Diabetes Mellitus gestacional (DMG):** o corpo da mulher tende a compensar o aumento da produção de hormônios que ocorre durante a gravidez produzindo mais insulina. Em certas mulheres, a compensação não ocorre, aumentando assim os níveis de glicose no corpo.

O diabetes também pode surgir como consequência de outras condições: Diabetes causado por doenças do pâncreas exócrino (como pancreatite, trauma, infecção, câncer pancreático e pancreatectomia); Diabetes devido a distúrbios endócrinos que causam secreção excessiva de hormônios que antagonizam a insulina; Diabetes induzido por drogas e produtos químicos de drogas que interrompem a secreção ou a ação da insulina; Diabetes relacionado à

infecção causado por infecção viral associada à destruição das células beta; Formas específicas incomuns de diabetes imunomediada (por exemplo, distúrbios imunológicos diferentes daqueles que causam diabetes tipo 1); Outras síndromes genéticas, às vezes associadas ao diabetes (ou seja, síndrome de PraderWilli, síndrome de Down, ataxia de Friedreich) e casos de diabetes recém-diagnosticados que não podem ser classificados em nenhuma das categorias descritas (IDF, 2019). Existe também o termo Pré-diabetes, que é um estágio que antecede o surgimento da DM2. Uma pessoa é diagnosticada como pré-diabética se esta apresenta níveis altos de glicose no sangue mas não o bastante para ser diagnosticada como DM2 (SOCIEDADE BRASILEIRA DE DIABETES, 2018).

2.1.1 Comorbidades e Complicações

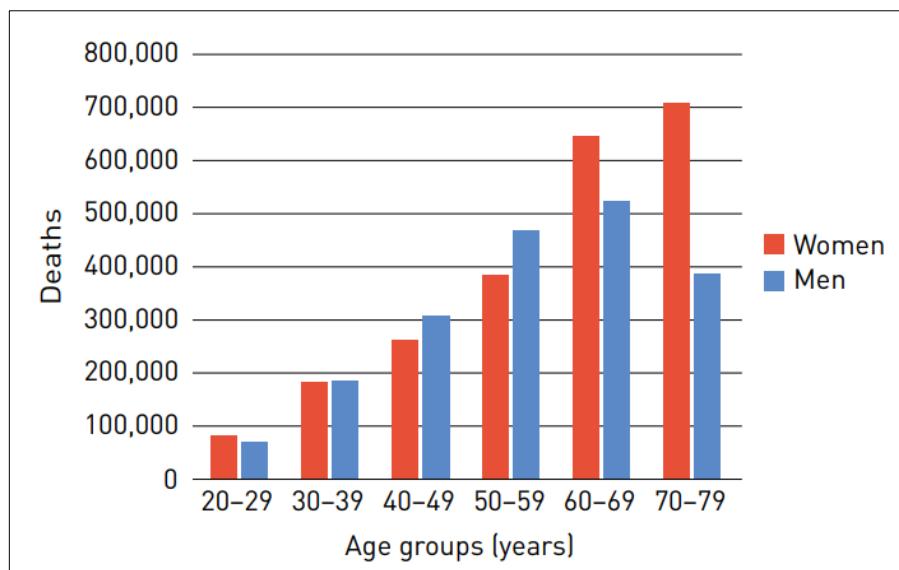
Quando o nível glicêmico permanece alto durante muito tempo da vida do paciente, complicações de saúde tendem a surgir e assim provocar o surgimento de outras doenças e comorbidades (LISANGELA, 2013). Frequentemente, o diabetes está associado a outras morbidades. Estudo utilizando os dados da PNS de 2013 mostrou que, dentre os indivíduos que declararam apresentar diabetes, 26,6% tinham relato de uma outra morbidade associada; 23,2%, de outras duas; e 32,0%, de outras três ou mais morbidades associadas. O diagnóstico de diabetes apareceu de forma isolada em apenas 18,1% dos indivíduos (RZEUWSKA et al,)

De acordo com a IDF (2019), as principais complicações do DM são:

- **Aumento do risco de Doença Cardiovascular (DCV):** os altos níveis de glicose no sangue tornam o sistema de coagulação sanguínea mais ativa, o que aumenta o risco de surgir coágulos. A doença cardiovascular é a causa mais comum de morte em pessoas com diabetes. Pressão alta, colesterol alto, glicose alta no sangue e outros fatores de risco contribuem para aumentar as chances de complicações cardiovasculares.
- **Perda de visão (retinopatia):** a maioria das pessoas com diabetes desenvolverá alguma forma de doença ocular, a presença de grandes quantidade de glicose no sangue prejudica os vasos capilares presentes no olhos, levando a hemorragia, bloqueio capilar, e, consequentemente, a perda da visão;
- **Problemas renais (nefropatia):** causada por danos a pequenos vasos sanguíneos. A nefropatia pode levar os rins a se tornarem menos eficientes ou a falharem completamente. A doença renal é muito mais comum em pessoas com diabetes do que naquelas sem doença.

- **Problemas nos nervos (neuropatia):** altas taxas de glicose no sangue podem causar problemas nos nervos em todo o corpo. Esta complicaçāo pode se passar desapercebida, causando doenças como disfunção autonômica cardíaca e o aparecimento de úlceras;
- **Pé diabético:** uma complicaçāo crônica e grave que consiste no aparecimento de lesões em tecidos profundos dos pés associadas com problemas neurológicos. Em estágio muito avançado, essa comorbidade pode resultar na amputação dos membros inferiores.

Figura 2: Número de mortes no mundo devido a diabetes em adultos (20-79 anos) por idade e sexo em 2019:



Fonte: *Internacional Diabetes Federation* (2019).

No tocante ao número de mortes resultantes da diabetes e suas complicações em 2019, estima-se que o número seja de 4,2 milhões entre adultos de 20 a 79 anos. A diabetes está associada a 11,3% das mortes globais em 2019, sendo que, quase metade (46,2%) das mortes por diabetes são em pessoas com menos de 60 anos, como mostra o gráfico da Figura 2.

No Brasil, são escassas as informações de base populacional sobre as complicações do diabetes. Merece destaque um estudo sobre a incidência de amputações de membros inferiores na região metropolitana do Rio de Janeiro, a qual foi de 13,9 por 100 mil habitantes para a população geral e de 180,6 por 100 mil habitantes para a população com diabetes, ou seja, uma taxa 13 vezes maior (SPICHLER et al, 2001).

Na Tabela 1 são apresentadas as taxas de mortalidade por diabetes, como causa básica, por faixa etária e macrorregião geográfica, para o ano de 2017, em que se pode observar o significativo crescimento do diabetes como causa de morte com o progredir da idade,

aumentando de forma exponencial da faixa etária de 0 a 29 anos para a de 60 anos ou mais, ou seja, com o envelhecimento populacional do Brasil atualmente, o diabetes certamente passará a ter maior contribuição para a mortalidade no país.

Quadro 1: Taxa de mortalidade por diabetes (a cada 100 mil habitantes), por macrorregião geográfica brasileira, segundo a faixa etária, no ano de 2017.

Faixa etária (anos)	Norte	Nordeste	Sudeste	Sul	Centro-Oeste	Total
0 a 29	0,6	0,7	0,7	0,5	1,7	1,1
30 a 39	2,6	3,3	2,8	2,5	2,8	2,8
40 a 49	10,2	12,4	8,4	8,4	14,8	9,7
50 a 59	46,4	41,7	28,3	30	31,9	33,3
60 e mais	255,6	263,4	150,9	181,7	188	90,1
Total	26,3	37,5	27,3	32,8	26,1	30,7

Fonte: DATASUS/MS, 2019.

Os dados apresentados reforçam a importância de tratar a diabetes e suas complicações. Para prevenir comorbidades e complicações é necessário dar atenção à saúde de modo eficaz. No diabetes, isso envolve prevenção do seu início, prevenção de suas complicações agudas e crônicas ou reabilitação e limitação das incapacidades produzidas. O autocuidado é uma é uma prática essencial nesse processo (GOLBERT et al, 2019).

2.1.2 Autocuidado

Apesar da DM ser uma doença crônica e grave, é possível ter uma vida com qualidade. Uma das coisas mais importantes é controlar o nível de glicose no sangue, para evitar complicações. A medição pode ser feita por meio de um monitor de glicemia ou por meio de bombas de insulina. Os dois tipos de aparelho devem ser adquiridos e usados com orientação da equipe multidisciplinar (SOCIEDADE BRASILEIRA DE DIABETES, 2018).

Para Organização Mundial de Saúde (OMS), um bom gerenciamento dos índices glicêmicos podem prevenir complicações e a morte prematura, isto é possível graças ao uso de intervenções que promovem estilos de vida mais saudáveis, como rastreamento e tratamento regular, para detecção precoce de complicações, que são frutos de uma boa educação do paciente quanto ao autocuidado (WHO, 2016).

Um estudo sobre controle glicêmico inadequado foi feito com 6.701 adultos com diabetes no Brasil (distribuídos em 12 centros de diabetes localizados em diferentes regiões do país), em que 979 (15%) pessoas tinham diabetes tipo 1 e 5.692 (85%) diabetes tipo 2. Os resultados mostraram que a prevalência de controle glicêmico inadequado foi de 76%. As características associadas aos adultos que apresentavam um bom controle incluíram: menos

anos de duração do diabetes, autocuidado, cuidado multiprofissional, participação em um programa de educação sobre a saúde do diabetes e satisfação com o tratamento atual do diabetes (MENDES et al, 2009).

Segundo Jongh (2012), o autocuidado é um conjunto de tarefas que uma pessoa pode executar para minimizar o impacto negativo que uma doença pode causar a sua saúde. Ele complementa ainda que o autocuidado requer que a pessoa tenha habilidades de automonitoramento dos sintomas e conhecimento das implicações, para assim ajustar a sua medicação, tratamento e comportamentos.

Abaixo são abordadas as principais tecnologias usadas para o autocuidado do DM (OLIVEIRA, 2017):

- **Bombas de infusão de insulina:** consiste em um aparelho com comando eletrônico semelhante a um pager que possui um reservatório de insulina e um cateter pelo qual ocorre a administração de insulina pelo tecido subcutâneo do paciente. As bombas injetam insulina no corpo do paciente de forma programada;
- **Sistemas de monitoramento contínuo de glicose (CGMS):** é um sistema que registra os níveis de glicose no tecido celular subcutâneo por meio de uma pequena cânula inserida sob a pele. Ela é capaz de medir também o fluido intersticial que correlaciona bem com a glicose plasmática;
- **Sensores de uso pessoal e de medida em tempo real:** nestes equipamentos são exibidos gráficos do padrão glicêmico que indicam tendências de elevação ou queda do nível de glicose no sangue. Eles podem ainda emitir alarmes sonoros em caso de hipo e hiperglicemia;
- **Aplicativos para smartphones:** são programas geralmente usados para registrar os níveis de glicose no sangue, calcular as doses de insulina, gerenciar hábitos saudáveis entre outras funcionalidades. São executados em aparelhos móveis, mais comuns em smartphones.

Estudos têm mostrado melhorias em resultados clínicos de pacientes que utilizam aplicativos móveis para melhorar o autocuidado da DM (NUNDY et al., 2013; ALOTAIBI; ISTE PANIAN; PHILIP, 2016).

A próxima seção trará informações sobre os aplicativos *m-health*, detalhando sobre as suas características e justificando a sua popularidade em torno de pacientes com DM, que pode servir como uma ferramenta para baixar os custos com tratamento do Diabetes Mellitus.

2.2 APLICATIVOS *M-HEALTH*

O acesso a novas tecnologias digitais como celulares e tablets vem conquistando espaço e despertando o interesse de profissionais de saúde, pesquisadores, população em geral, especialmente os idosos (NAHAR et al., 2017). De acordo com Pressman e Maxim (2016), aplicações móveis ou *mobile* são programas de software projetados especificamente para os dispositivos móveis com sistemas operacionais como Android, iOS ou Windows Phone, que são também chamados de plataformas móveis. Elas se utilizam de mecanismos de interação exclusivos da plataforma móvel, como também são consumidoras de serviços *web* através de *browsers* adaptados ao aparelho, interagem com diversos tipos de hardwares presentes no dispositivo (acelerômetro ou localização por GPS) e armazenam dados de forma persistente em seu armazenamento local.

Entre essas abordagens, encontram-se os aplicativos voltados para à área da saúde, conhecidos como *m-Health* (*mobile health* – saúde móvel), que consistem nas práticas de saúde que são amparadas por dispositivos portáteis como smartphones, tablets, aparelhos de monitoramento de pacientes, assistentes pessoais digitais, e outros aparelhos sem fio (WHO, 2018). O termo *m-Health* engloba, sobremaneira, conceitos como computação móvel, sensores médicos e tecnologias de comunicação para cuidados com a saúde (ISTEPANIAN et al. 2004).

Aplicações *m-Health* podem oferecer orientações sobre a doença de maneira individualizada, colaborando com a aproximação dos profissionais de saúde, pacientes e seus familiares. Essas soluções são importantes para a autopromoção da saúde e para a redução de gastos econômicos. Elas acompanham o manejo nas doenças crônicas por meio do monitoramento remoto, orientação psicológica e apoio na tomada de decisão (AZEVEDO et al., 2018), com recursos para monitorar a dieta, peso corporal, pressão arterial, humor, sono, entre outros (PAGLIALONGA; LUGO; SANTORO, 2018).

A utilização de aplicativos móveis na área da saúde permite a realização de atendimento e de cuidado com avaliação e resposta imediatas, favorecendo assim uma assistência mais efetiva. Estes aplicativos têm como característica principal a facilidade de utilização e mobilidade, sendo que seu uso na área da saúde vem crescendo rapidamente, devido ao fato de favorecer a precisão e agilidade para a tomada de decisão dos profissionais de saúde, além de facilitar a pesquisa científica no local de trabalho (ISTEPANIAN et al. 2004).

2.2.1 Uso de dispositivos móveis no Brasil

Para usufruir dos benefícios das tecnologias *m-Health*, é necessário que o usuário possua um dispositivo móvel, e, em alguns casos, com acesso a internet. Em 2019, 148,4 milhões de pessoas de 10 anos ou mais de idade tinham telefone móvel celular para uso pessoal, o que correspondia a 81,0% da população desta faixa etária, percentual um pouco maior que o estimado para 2018 (79,3%). Em relação ao sexo, 82,5% das mulheres e 79,3% dos homens tinham telefone móvel celular para uso pessoal no país (IBGE, 2019).

O acesso à internet por meio do celular é um recurso de comunicação e de obtenção de informação que está se integrando cada vez mais ao cotidiano dos brasileiros. Em 2019, a população acima dos 10 de idade do país era de 183,3 milhões, dessa quantidade, 78,3% (ou 143,5 milhões) haviam utilizado a internet nos últimos três meses a contar da data de realização da pesquisa. De 2018 para 2019, na população de 10 anos ou mais de idade que possuía um dispositivo móvel para uso pessoal no país, a parcela que tinha acesso à internet por meio deste aparelho aumentou de 88,5% para 91,0%. Entre os idosos de 60 anos ou mais, principais vítimas das complicações e comorbidades da diabetes, o percentual de pessoas que tinham telefone móvel celular para uso pessoal é de 67% (IBGE, 2019).

De acordo com um relatório feito pela App Annie (2018), cerca de 90% dos smartphones do Brasil possuem Sistema Operacional (SO) Android, enquanto que nos 10% restantes estão os demais SOs como iOS e Windows Phone. Isto acontece por que os aparelhos com Android são mais baratos que os smartphones com iOS e fornecem maior suporte e funcionalidades que os do Windows Phone (AVAZU, 2016).

Os aplicativos desenvolvidos para a área da saúde são úteis para promover a melhoria da qualidade de vida e da adesão ao tratamento, além de descomplexificar a comunicação entre o profissional de saúde e o paciente (FERGUSON, 2017). Eles podem ser encontrados de acordo com as necessidades específicas de cada usuário e estão disponíveis em lojas virtuais para serem instalados nos dispositivos móveis, sendo as duas principais a “Play Store” (para dispositivos com Sistema Operacional Android - Google), e “App Store” (para dispositivos com Sistema Operacional iOS - Apple). A próxima seção apresentará os principais aplicativos para o autocuidado e gerenciamento do Diabetes Mellitus disponíveis para o sistema operacional Android.

2.2.2 Aplicativos para autocuidado e gerenciamento do Diabetes Mellitus

No mercado atual de aplicativos mobile para o sistema operacional Android, existem aproximadamente 250 aplicativos relacionados a diabetes, cada um dele com inúmeras funcionalidades, como: recomendações alimentares; gerenciamento do nível de glicose no sangue; envio de dados via e-mail para seu médico; produção de gráficos diários e mensais dos dados glicêmicos; exportação de dados; envio de notificações para familiares e/ou médico; utilização tanto online, quanto off-line; notificação para verificação e medição da glicose no sangue; realização de backup dos dados na nuvem; orientações sobre atividades físicas; acompanhamento do nível de glicose com um medidor bluetooth ou manual; dentre outras (GOOGLE PLAY, 2016).

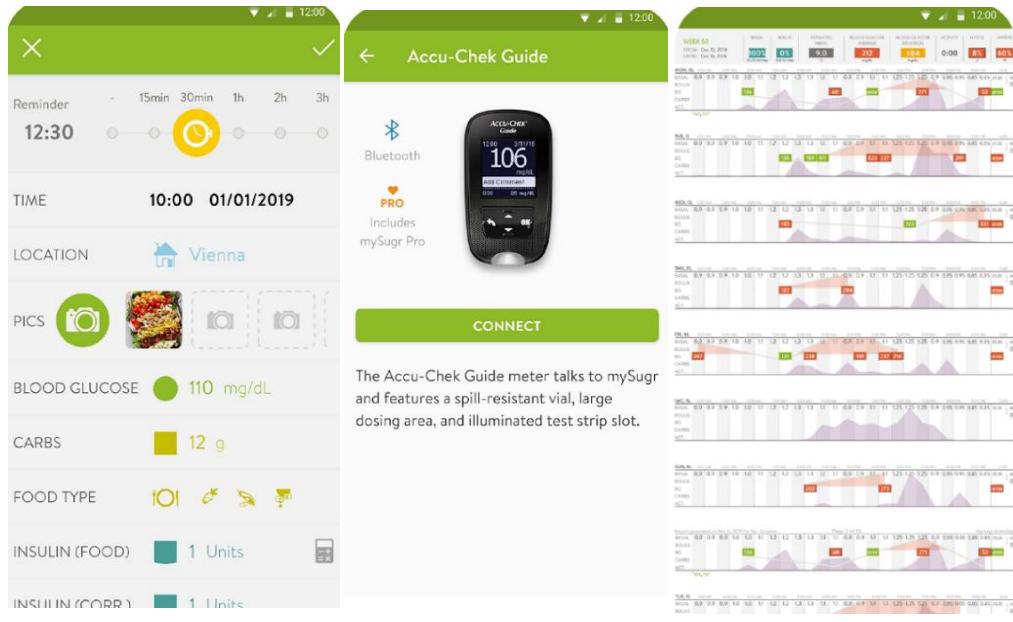
A seguir, serão listados apenas alguns (os mais baixados, utilizados e citados na literatura) dentre vários aplicativos para autocuidado destinados aos portadores de Diabetes Mellitus e suas principais funcionalidades, ao final da subseção estas funcionalidades são agrupadas e apresentadas em um quadro, para fins de comparação.

2.2.2.1 MySugr: Diário da Diabetes

O mySugr: Diário da Diabetes é um aplicativo que auxilia o paciente diabético a ter um melhor monitoramento da sua doença. Dentre suas funções, ele é capaz de armazenar os registros de glicemia, alimentação, carboidratos e medicações, bem como alertar sobre aferição da glicemia, configurar lembrete, emitir relatórios e outras funcionalidades. Estas funções possibilitam o usuário analisar e armazenar as informações referentes ao seu dia-a-dia, acompanhar sua evolução no tratamento e prevenção da diabetes, podendo inclusive ser enviadas a um profissional de saúde, caso necessário (MYSUGR GMBH, 2018)..

O aplicativo possui uma interface intuitiva e de fácil aprendizagem, além disso, suas principais funções não necessitam de conexão com a internet. O mySugr traz uma funcionalidade diferencial dos demais aplicativos, que é a rede de estatísticas que possibilita ao usuário verificar níveis glicêmicos em semana, quinzenas, mês e trimestre (MYSUGR GMBH, 2018)..

Figura 3: Capturas de tela do aplicativo “MySugr: Diário da Diabetes”

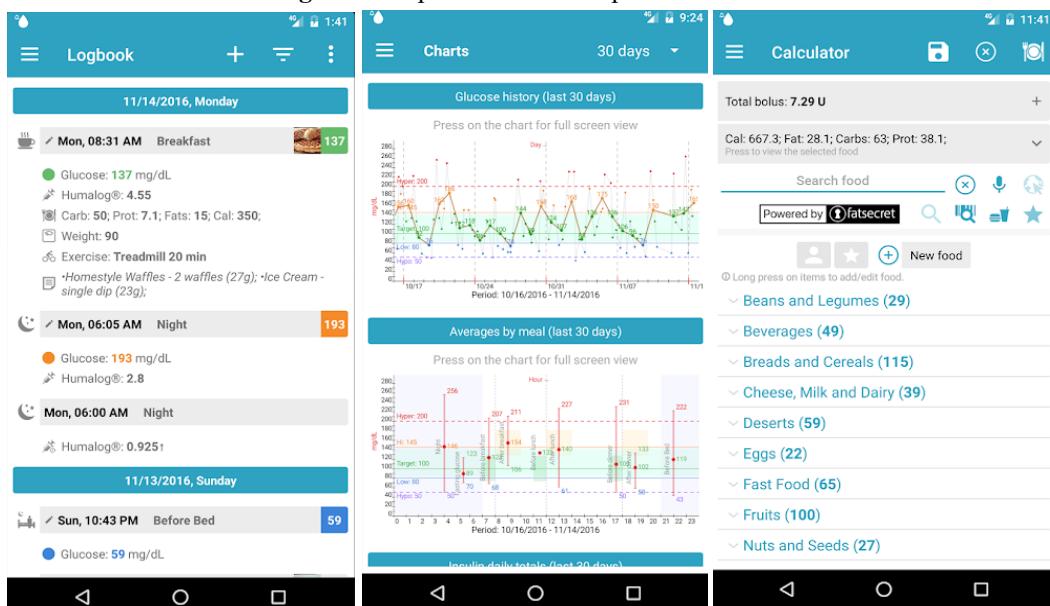


Fonte: MYSUGR GMBH (2018).

2.2.2.2 Diabetes:M

Aplicativo para pacientes diabéticos, que facilita o acompanhamento da doença. O usuário pode usufruir de funções como registros dos níveis de glicemia, alimentação, uso das insulinas e medicamentos orais. O aplicativo possui uma interface simples e de fácil aprendizagem. Ele possui ainda funções de guardar relatórios e alarmes que auxiliam na vida cotidiana dos pacientes/usuários (SIRMA MEDICAL SYSTEMS, 2018).

Figura 4: Capturas de tela do aplicativo “Diabetes: M”



Fonte: SIRMA MEDICAL SYSTEMS (2018).

Ele permite cadastrar e manter valores de glicose, pressão sanguínea, exercícios e níveis de hemoglobina glicada em um diário, fornecer relatórios, gráficos e estatísticas que podem ser distribuídas por e-mail com o profissional de saúde; e verificar valores importados dos glicosímetros (SIRMA MEDICAL SYSTEMS, 2018).

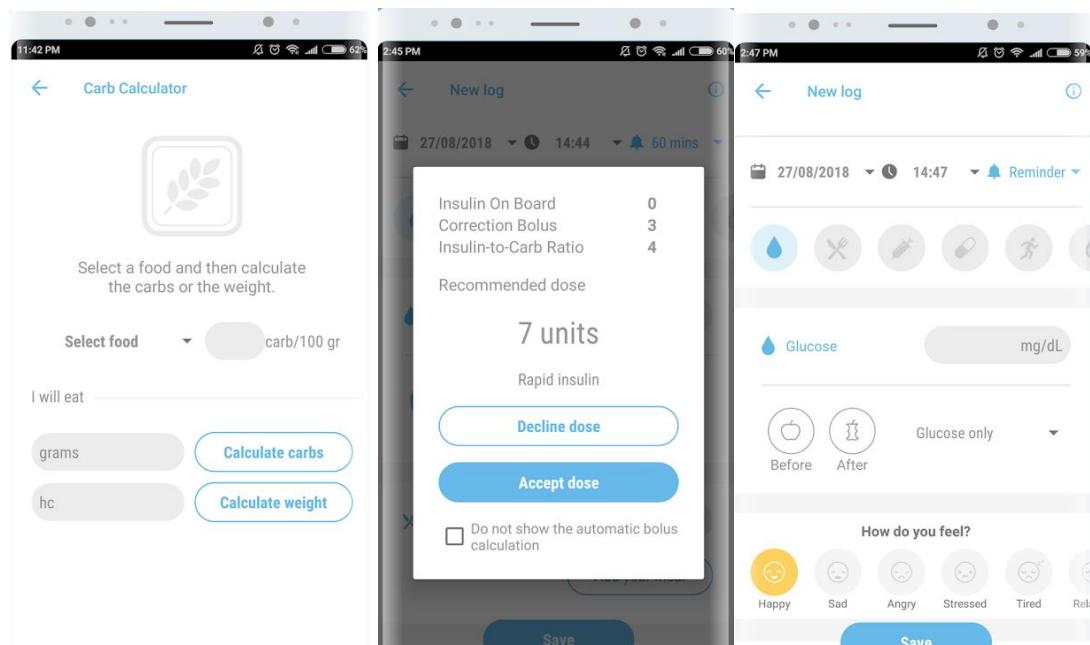
As principais funções não necessitam de internet. É importante ressaltar que para que os usuários possam utilizar mais funções do aplicativo, o mesmo deve adquirir a versão premium, que é paga (SIRMA MEDICAL SYSTEMS, 2018).

2.2.2.3 SocialDiabetes

SocialDiabetes é um aplicativo que ajuda no monitoramento do paciente diabético. Várias funções estão contidas nessa ferramenta, entre elas: registro dos níveis de glicemia, alimentação, tratamento por insulina ou medicamentos orais, registro de exercícios físicos, entre outras (SOCIAL DIABETES, 2018).

Ele possui uma interface dinâmica com a exibição configurável dos componentes. Contempla ainda as médias glicêmicas por dia, semana e mês. Além disso, pontos de hipoglicemias do mês anterior em relação ao mês atual e uma estimativa da Hemoglobina Glicada (HbA1c), a Figura 5 exibe algumas destas funcionalidades (SOCIAL DIABETES, 2018).

Figura 5: Capturas de tela do aplicativo “SocialDiabetes”



Fonte: SOCIAL DIABETES (2018).

2.2.2.4 Dieta para Diabéticos

O aplicativo Dieta para diabéticos foi criado para portadores de Diabetes Mellitus, ele possui a funcionalidade de mostrar informações sobre a quantidade e os tipos de alimentos que o diabético deve ingerir. Não possui nenhum tipo de ferramenta para registrar ou armazenar os níveis de glicose, mas possui um grande número de dietas para pessoas diabéticas, além de informações sobre a doença (GOOGLE PLAY, 2016).

Figura 6: Capturas de tela do aplicativo “Dieta para diabéticos”

Plano para diabéticos

Um plano alimentar para diabéticos não é muito diferente de um plano alimentar saudável e equilibrado aplicável à restante população. A diabetes é uma doença causada pela falta ou resistência à insulina, uma hormona produzida pelo pâncreas. Esta alteração resulta de uma falha na absorção da glicose (açúcar), levando a uma acumulação deste açúcar no sangue do organismo, que, por sua vez, é perigosa para o corpo e está associada a vários outros problemas. É uma doença crônica, cujo tratamento ideal consiste na conjugação de fatores como a alimentação, a medicação e a prática de exercícios.

Receitas diabéticos

- Quibe de vegetais assado
- Sobrecoxa de frango assada com iogurte e linhaça
- Caril de camarão com arroz basmati e legumes salteados
- Vitela estufada com quinoa e couve de bruxelas
- Filé mignon ao molho de cogumelos e castanha de caju
- Salada de macarrão integral
- Bolo de banana com aveia e nozes
- Salada tropical
- Pão de banana e noz sem açúcar

Filé de salmão grelhado

Ingredientes:

- ¼ xíc. de azeite
- ½ col. (chá) de cominho
- ¼ col. (chá) de tomilho
- 1 col. (sopa) de suco de limão
- 1 col. (sopa) de salsa picada
- ¼ xíc. alho-poró picado
- 600g de filés de salmão sal a gosto

Método:

Junte todos os temperos em uma tigela, coloque o peixe e deixe descansar por uma hora.

Fonte: GOOGLE PLAY (2016).

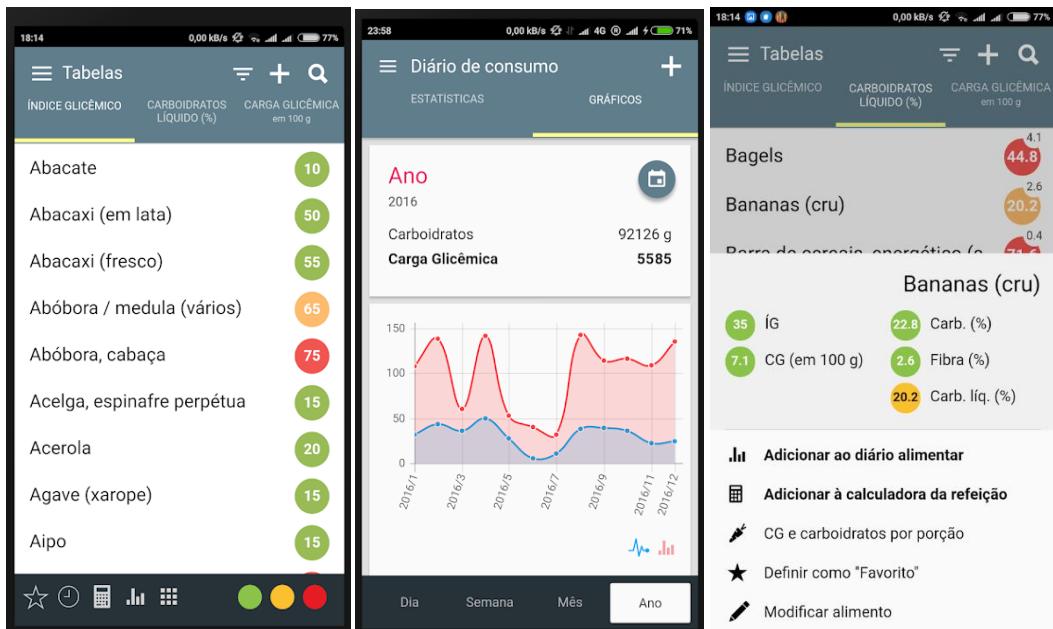
2.2.2.5 Índice e Carga Glicêmicas: alimentos para diabetes

Índice e Carga Glicêmica é um app que permite navegar, pesquisar e exibir o Índice Glicêmico de diferentes alimentos. O aplicativo também ajuda a manter o registro das medições do peso corporal e dos níveis de glicemia. Adicionalmente, o usuário pode acessar a Carga Glicêmica e teor de carboidratos nos alimentos. Há também uma calculadora de Carga Glicêmica em uma dada porção (GOOGLE PLAY, 2016).

Entretanto, a maioria das funcionalidades do aplicativo apenas podem ser utilizadas se o usuário adquirir a versão premium, que é paga. Entre as funcionalidades gratuitas, se destacam tabelas para consulta do índice glicemico de vários alimentos, além de informações sobre alimentação saudável e diabetes. As funiconalidades pagas incluem: lista de carga glicêmica dos alimentos, lista do teor de carboidratos, calculadora do teor de fibras nos alimentos e do

teor líquido de carboidratos, estatísticas com médias para períodos diários, semanais, mensais e anuais, gráficos e relatórios. Algumas dessas funcionalidades estão ilustradas na Figura 7 (GOOGLE PLAY, 2016).

Figura 7: Capturas de tela do aplicativo “Índice e Carga Glicêmicas: alimentos para diabetes”

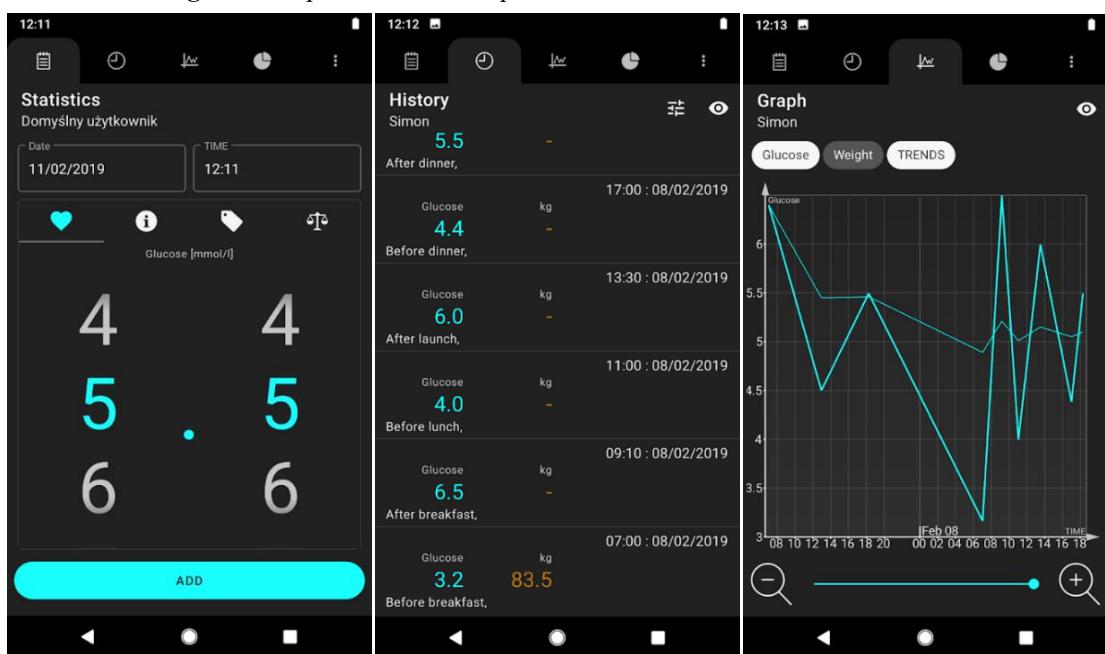


Fonte: GOOGLE PLAY (2016).

2.2.2.6 Diabete – diário glucose

Algumas telas do aplicativo Diabete-Diario Glucose estão dispostas na Figura 8 .

Figura 8: Capturas de tela do aplicativo “Diabete – Diário Glucose”



Fonte: GOOGLE PLAY (2016).

O aplicativo possui uma interface simples, contendo dados sobre o nível de glicose no sangue, gráficos interativos, função de enviar dados para o médico ou outra pessoa; função de exportar dados para .csv e .xml e função para delimitar marcadores de ações como, almoço, jantar, medicação, exercícios, correr ou qualquer outra coisa que envolva os níveis de glicose. Não possui nenhum tipo de informação alimentar, tabelas nutricionais, dicas, ou dietas (GOOGLE PLAY, 2016).

No Quadro 2 é exposto a comparação das funcionalidades dos aplicativos mobile descritos acima e do aplicativo a ser construído neste projeto.

Quadro 2: Comparativo entre funcionalidades dos aplicativos analisados

Funcionalidades	Aplicativos mobile analisados:						
	1	2	3	4	5	6	Protótipo a ser desenvolvido
<i>X: possui a funcionalidade</i> <i>\$: possui a funcionalidade apenas na versão paga</i> <i>-: não possui a funcionalidade</i>							
Cadastro de dados pessoais	X	-	X	-	X	-	X
Atribuir senha para acesso ao app	-	-	-	-	-	-	X
Registro e gerenciamento do nível de glicemia	X	X	X	-	\$	-	X
Registro e gerenciamento de medicamentos orais/insulina	X	-	X	-	-	-	X
Registro e gerenciamento de informações alimentares	X	-	X	-	\$	X	X
Registro e gerenciamento de exercícios físicos	X	X	X	-	-	-	X
Consultar informações nutricionais, dieta para diabéticos ou nível de glicose dos alimentos	\$	-	-	X	X	X	X
Sistemas de lembretes, alarmes ou notificações	X	X	-	-	-	-	X
Gerar gráficos dos dados glicêmicos	X	X	-	-	X	X	X
Exportar relatórios	X	X	-	-		X	X

Fonte: Autor.

2.3 DESENVOLVIMENTO DE SOFTWARE

Segundo Sommerville (2011), software é caracterizado como um programa de computador e toda a documentação associada a ele.

Os softwares podem ser de três tipos diferentes, todos eles oferecem o mesmo produto final, a informação. Um software *desktop* é aquele que utiliza o sistema operacional para obter dados e fornecer informações ao usuário, já o software *web*, utiliza o navegador para essa mesma função, e a internet para transmitir as informações. Um software *mobile*, popularmente conhecidos como aplicativo, também utiliza os sistemas operacionais para realizar o fluxo de informações, porém, apenas em dispositivos móveis (*smartphones* e tablets) (GONCALVES, 2016).

A seguir, serão apresentados os principais conceitos necessários para compreender o processo de desenvolvimento de software para dispositivos móveis.

2.3.1 Aplicações móveis nativas

Existem dois principais paradigmas de desenvolvimento móvel, o Nativo e o Multiplataforma, cada paradigma apresenta suas próprias abordagens de desenvolvimento. O paradigma Nativo traz a abordagem de desenvolvimento de aplicações nativas, ou seja, software *mobile* que são projetados especificamente para uma plataforma móvel. Já as abordagens de aplicações Web Móveis, Híbridas, Interpretadas e Multi Compiladas são construídas a partir do paradigma Multiplataforma, onde geram aplicativos para várias plataformas de uma só vez a partir de um único código (LATIF et al., 2017; BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017).

Cada paradigma possui um conjunto de pontos fortes e fracos que são comuns as abordagens que elas abrangem. Estas características devem ser levadas em consideração antes de se iniciar a construção de um aplicativo móvel (QUE; GUO; ZHU, 2017; BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017; JOBE, 2013).

Na abordagem de desenvolvimento de aplicações nativas, os desenvolvedores utilizam diversas ferramentas, linguagens e padrões próprios da plataforma, de modo que o resultado é um aplicativo móvel compatível apenas a uma plataforma móvel (QUE; GUO; ZHU, 2017). Estas aplicações estão sendo desenvolvidas por meio de pelo menos três linguagens de programação: Java para o Android, Objective-C para o iOS e C# para Windows Phone (PRESSMAN; MAXIM, 2016). As plataformas de desenvolvimento mais comuns para

desenvolvimento nativo são: Eclipse⁶ ou Android Studio⁷ para o Android, Xcode⁸ para o iOS e Visual Studio⁹ para o Windows Phone (LATIF et al., 2017).

Segundo Jobe (2013), as aplicações nativas são melhores na criação de conteúdo e apresentam melhor performance em relação ao hardware do dispositivo. As características comuns entre elas é que todas têm acesso direto ao hardware do dispositivo, dá suporte em todas as interfaces do usuário e interagem com todo o ambiente da plataforma móvel.

Por serem programados exclusivamente para o sistema operacional, o aplicativo nativo é mais rápido e confiável que os demais, pois apresenta uma melhor experiência para o usuário ao conseguir utilizar todos os recursos oferecidos pelos smartphones como câmera, GPS e notificações push. Aplicativos nativos também possuem um maior tempo de utilização do que os demais por poderem funcionar sem conexão internet (USEMOBILE, 2020).

Ao se programar um aplicativo nativo, os desenvolvedores respeitam a um padrão de design. Guias são oferecidos para cada sistema operacional, que contém as melhores práticas para proporcionar uma boa experiência ao usuário. Nesse tipo de aplicação, os desenvolvedores utilizam o chamado Ambiente de Desenvolvimento Integrado (IDE). Ele fornece ferramentas que auxiliam na construção do aplicativo e auxiliam a agilizar o processo (USEMOBILE, 2020).

2.3.2 Levantamento e Análise de Requisitos

Os requisitos são as descrições do que um software deve fazer, quais serviços disponibiliza e as restrições de seu funcionamento. O método de encontrar, avaliar e registrar esses serviços e restrições é conhecido como engenharia de requisitos (BUCKELL, 2017).

Para Sommerville (2011), podem ser definidos dois tipos de requisitos para um software: requisitos de sistema e requisitos de usuário. O primeiro descreve de forma detalhada e específica as funções, serviços e restrições do software, enquanto o segundo descreve de uma forma mais informal e geral quais os serviços e restrições o software fornecerá aos usuários. São também chamados de requisitos funcionais e não-funcionais, respectivamente.

Os requisitos funcionais descrevem o que este deverá fazer, ou seja, as funcionalidades do software, de forma geral ou específica, de acordo com o tipo de software a ser desenvolvido. É importante que, ao especificar os requisitos, todos os serviços e funções solicitados pelo cliente ou usuário final sejam definidos e essas definições sejam coerentes. Dessa forma, o software poderá ser entregue mais rápido, sem necessidade de rever o desenvolvimento por causa de requisitos mal definidos ou inconsistentes (SOMMERVILLE, 2011).

De acordo com Sommerville (2011), requisitos não funcionais são os requisitos que não estão propriamente associados aos serviços e funções específicos disponibilizados pelo software aos usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Para Guedes (2011), podem existir diversos tipos de requisitos não-funcionais, como de usabilidade, desempenho, confiabilidade, segurança ou interface.

Essa distinção se torna útil para informar diferentes tipos de usuários e desenvolvedores dos diferentes tipos de detalhamento dos requisitos, que podem utilizá-los de diversas formas. Nessa etapa, as informações são obtidas com os stakeholders (clientes, usuários finais ou pessoas da organização que serão afetadas pelo software) sobre as funcionalidades que software possuirá, as tarefas que ele irá ou não realizar.

As etapas que compõem o processo de levantamento de requisitos são: descoberta, classificação e organização, priorização e negociação, especificação. Para tal, geralmente são realizadas entrevistas com os stakeholders, que podem ser do tipo fechada, em que são definidas as perguntas da entrevista, e do tipo aberta, onde não são perguntas predefinidas, mas é feita uma análise com os stakeholders para uma melhor compreensão de suas necessidades (SOMMERVILLE, 2011).

Para Guedes (2011), após o levantamento dos requisitos vem a fase da análise desses requisitos, para avaliar se todas as necessidades dos usuários foram entendidas corretamente e determinar se algo precisa ser melhorado.

2.3.3 Paradigma da Orientação a Objetos

O Paradigma de Orientação a Objetos (POO) é um paradigma de análise, projeto e programação de software que adota o conceito de padrão como base. Ele estabelece a modelagem, a representação e a implantação de sistemas através de entidades ativas chamadas objetos que pertencem às classes. É, por conseguinte, um método diferenciado de outras linguagens empregadas na programação de software, não sendo linear e procedural. Além disso, permite a identificação e a estruturação de sistemas complexos através da decomposição de modelos hierárquicos e da abstração de contextos do mundo real (BOOCH, 2007).

Os elementos da orientação a objetos são definidos através da decomposição do problema e posteriormente representados para visualizar e manipular suas interações. No que concerne à definição destes elementos, tem-se que classes são conjuntos de objetos que compartilham uma estrutura e um comportamento em comum, enquanto que objetos são

entidades com comportamento, estado e identidade individuais organizados pelas classes a qual pertencem. Os objetos possuem atributos, que são suas características, e métodos, que implementam as suas habilidades. As classes abstratas fornecem uma estruturação de atributos e métodos para as classes complementares. Associação, por sua vez, é um recurso que determina se uma classe ou objeto faz parte de outra classe ou apenas utiliza um componente desta classe. Quanto à herança ou generalização, trata-se do mecanismo que permite uma classe herdar atributos e métodos de outras classes. Polimorfismo representa a capacidade de objetos responderem de distintas maneiras quando determinado procedimento é invocado (BOOCH, 2007). Booch (2007) descreve quatro conceitos fundamentais para POO:

- **Abstração:** consiste na decomposição do modelo em classes, definindo somente as principais características para possibilitar o entendimento do conjunto;
- **Encapsulamento:** agrupamento de diversos elementos em classes, segundo uma abstração e a sua separação entre a implementação e a interface, em que a implementação é formada pelos processos internos e a interface é o ponto de relacionamento das classes com o conjunto;
- **Modularidade:** as variadas partes de um objeto devem se comportar como módulos, conectando-se para formar uma estrutura complexa;
- **Hierarquia:** consiste na classificação e na ordenação das classes.

Os métodos de projeto baseados no paradigma da orientação a objetos (POO) são importantes meios para o desenvolvimento de software. Aryana (2007) observa que a orientação a objetos agrupa vários benefícios para desenvolvimento de um projeto, como a modularidade e a reusabilidade, e é largamente adotada por desenvolvedores da área de informática e instituições de ensino e pesquisa desta mesma área. Este mesmo autor indica a possibilidade de utilizar os recursos do POO no projeto de artefatos físicos e apropriar-se das vantagens das ferramentas de representação empregadas na POO.

Os recursos da orientação a objetos para modelar e manipular o projeto são baseados em diagramas específicos, denominados diagramas UML, desenvolvidos para visualizar, especificar, construir e documentar artefatos de um sistema de software, empregando diferentes tipos de diagramas apropriados a situações específicas de projeto (BOOCH, 2007).

2.3.4 Diagramas UML

De acordo com Guedes (2011), a UML – Unified Modeling Language ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar softwares baseados

no paradigma de orientação a objetos. Ela se tornou padrão de modelagem, adotada internacionalmente pela indústria de engenharia de software. A UML não é uma linguagem de programação e sim de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de software a definirem as características do sistema, tais como: seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema será implantado (GUEDES, 2011).

A Linguagem Unificada de Modelagem (UML) trabalha a representação de ideias, entidades e ações em padrões classificatórios. A modelagem torna-se necessária, porque a atividade de desenvolvimento de sistemas é bastante complexa, além disso, os sistemas geralmente envolvem grandes volumes de itens e são compostos por muitos módulos. Dessa forma, só se deve começar a desenvolver um sistema depois de uma definição dos requisitos e das funcionalidades que o sistema deve possuir (GUEDES, 2011).

Modela-se então esse conjunto de dados e informações através de uma linguagem de representação; os diagramas resultantes servem de meio de comunicação entre analistas e usuários e entre os próprios membros das equipes de analistas e programadores. A linguagem UML, ambiente no qual são usados os conceitos do POO para modelar sistemas, constitui-se por diversos tipos de diagramas. Iremos ver neste trabalho os diagramas de caso de uso, diagramas de atividades, diagramas de classe e diagrama de sequência (GUEDES, 2011).

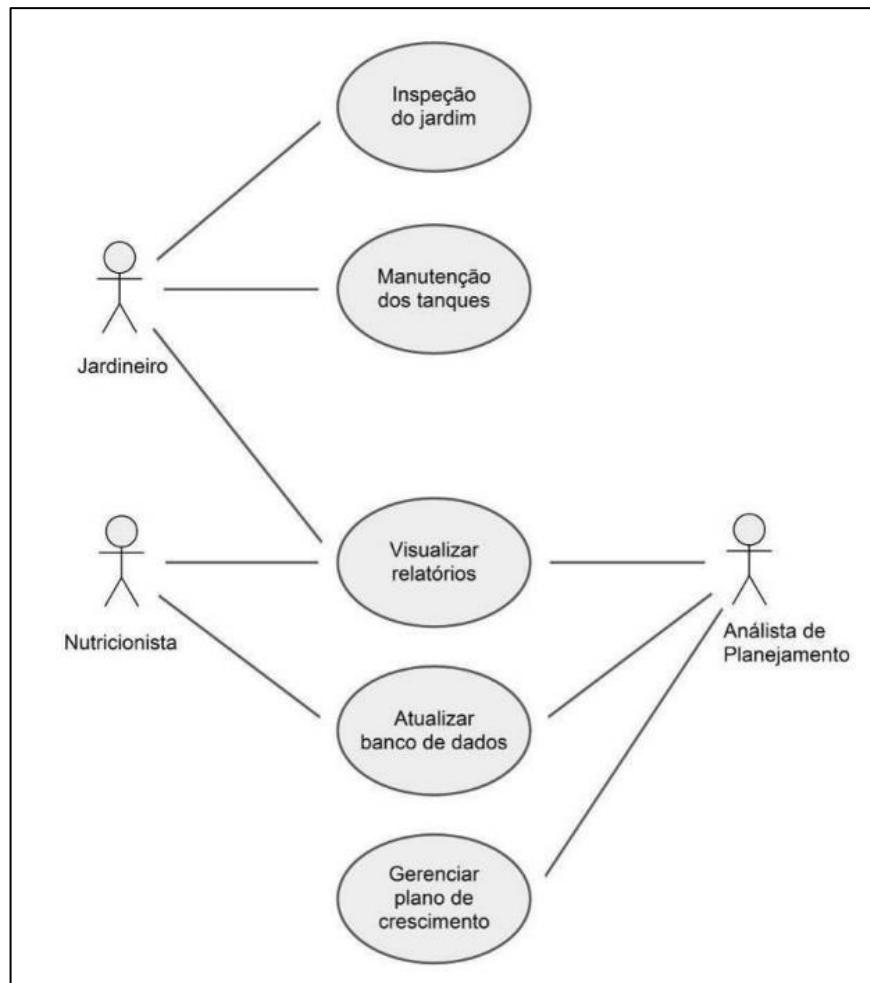
2.3.4.1 Diagrama de casos de uso

Segundo Guedes (2011), o diagrama de caso de uso é elaborado no etapa inicial da modelagem de um software. Ele pode ser consultado e possivelmente modificado durante o processo de levantamento de requisitos e serve como base para outros diagramas. O autor afirma que o diagrama de casos de uso é fundamental na identificação e compreensão dos requisitos, possibilitando especificar, visualizar, e documentar as características, funções e serviços do software que o usuário deseja. O diagrama também ajuda os desenvolvedores pontuarem quais os tipos de usuário que irão utilizar o software, quais papéis esses usuários irão cumprir e quais funções um usuário específico poderá requisitar.

Sommerville (2011) afirma que o diagrama de caso de uso detalha as interações entre os usuários e o sistema. Além disso, de acordo com Gomes e Wanderley (2003), os diagramas de caso de uso são usados para demonstrar as funcionalidades que o sistema a ser criado irá prover, esse diagrama revela ainda a forma como atores (pessoa ou outro sistema) interagem

com o sistema proposto. A Figura 4 mostra um exemplo representativo de um diagrama de caso de uso.

Figura 9: Exemplo de diagrama de caso de uso



Fonte: Booch (2007).

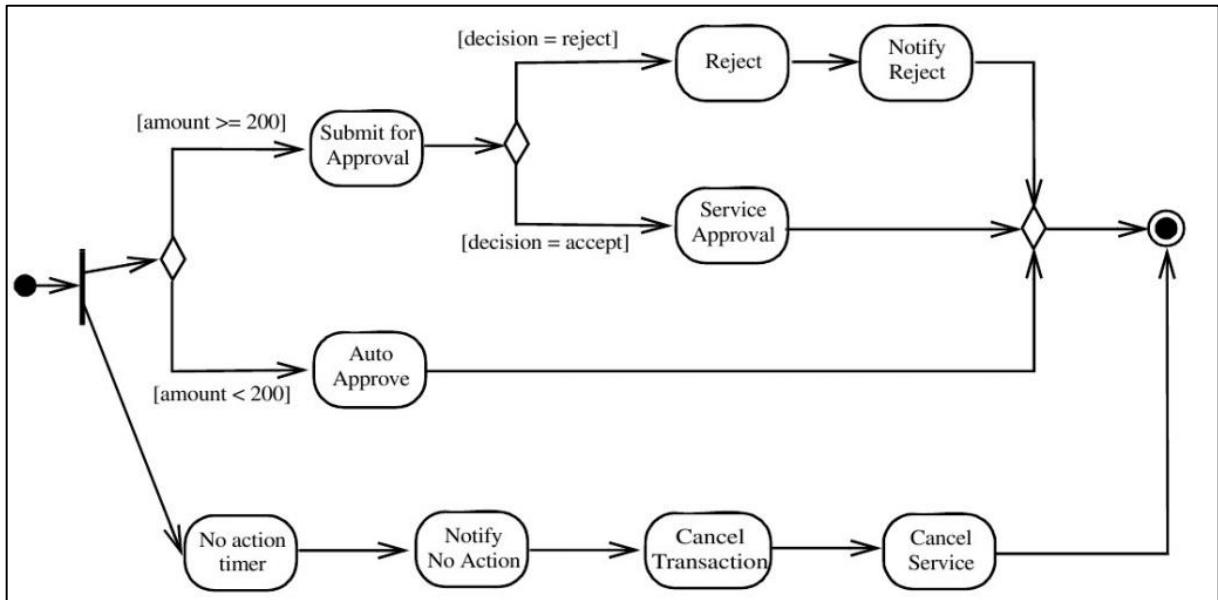
De acordo com Booch, Rumbaugh e Jacobson (2000), o caso de uso exemplifica o comportamento desejado de um potencial usuário, mas não pontua como este será executado. Segundo Sampaio (2007), os relacionamentos ilustrados pelo diagrama auxiliam o desenvolvedor a descrever casos de uso, e podem ser entre um ator e um caso de uso, entre atores ou entre casos de uso.

2.3.4.2 Diagrama de atividades

O diagrama de atividades é um diagrama UML utilizado para modelar o aspecto comportamental de processos. Neste diagrama, uma atividade é modelada como uma sequência estruturada de ações, controladas potencialmente por nós de decisão e sincronismo. Em seu aspecto mais simples, um diagrama de atividades pode ser confundido com um fluxograma.

Entretanto, ao contrário de fluxogramas, os diagramas de atividades UML suportam diversos outros recursos, tais como as partições e os nós do tipo *fork* e *merge*, além da definição de regiões de interrupção, que permitem uma modelagem bem mais rica do que simplesmente um fluxograma (GUDWIN, 2021). Um exemplo de um diagrama de atividade simples é mostrado na Figura 10 a seguir.

Figura 10: Exemplo de um Diagrama de Atividades Simples.



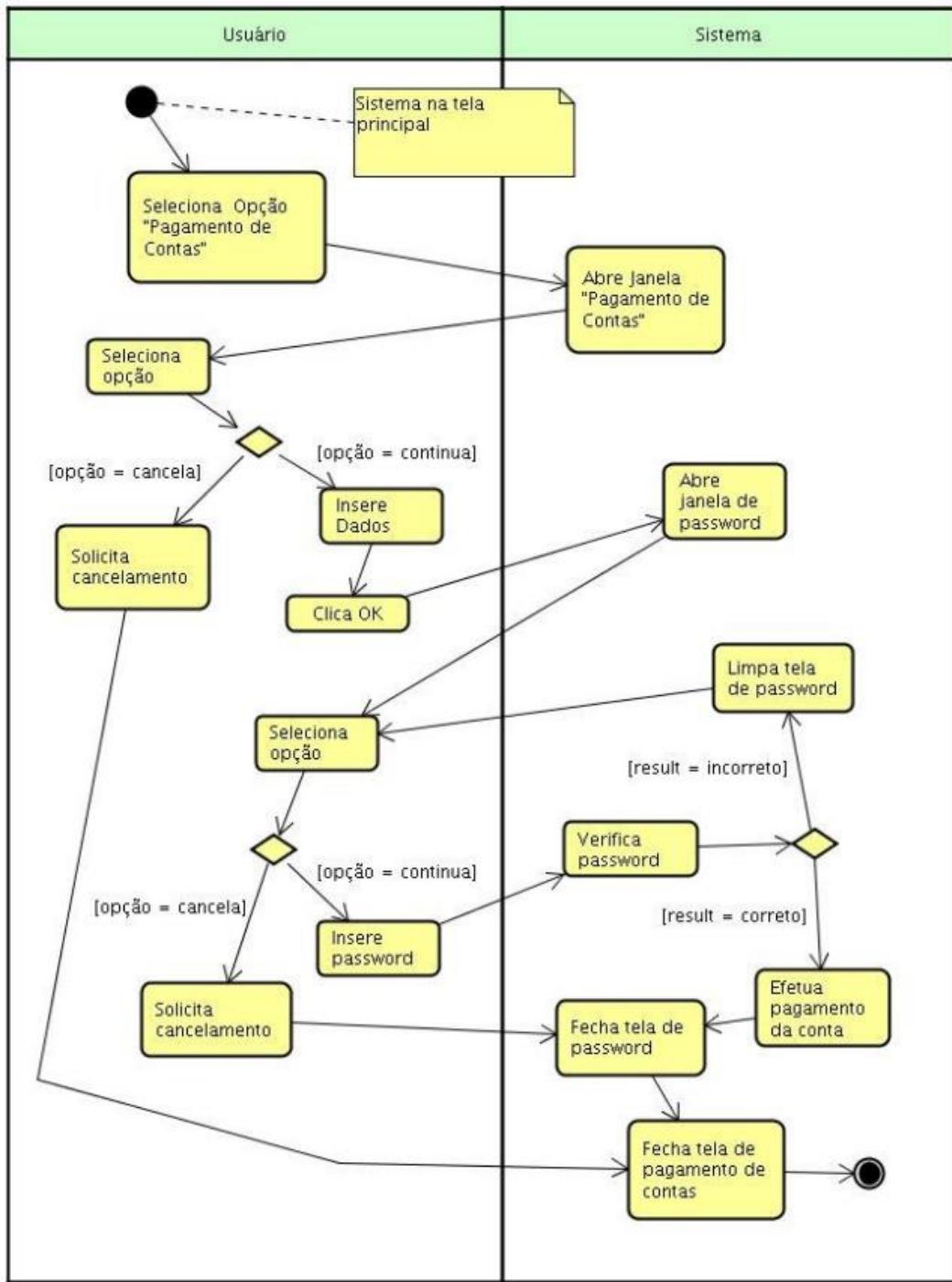
Fonte: Gudwin (2021).

Neste diagrama, o ponto preto à esquerda indica o início da atividade, as caixas com cantos arredondados representam ações, os pequenos losangos são nós de decisão e a barra vertical preta é um nó de sincronização do tipo *fork*. Os arcos conectando as ações representam a sequência em que as ações devem ser executadas, sendo que nos arcos que saem dos nós de decisão existem condições de guarda, que decidem qual será a próxima ação a ser executada. Essas condições de guarda devem ser proposições mutuamente exclusivas, de tal forma que para n arcos saindo de um nó de decisão, somente um deles pode ser verdadeiro a cada instante de tempo. Por fim, no canto direito do diagrama encontra-se um nó de finalização, que denota o final da atividade (GUDWIN, 2021).

A Figura 11 a seguir mostra um diagrama de atividades com partições. Diagramas com partições permitem que processos onde múltiplos agentes atuam, potencialmente em paralelo, possam ser representados. As partições servem para indicar que diferentes ações são executadas por diferentes agentes dentro de um processo. Sem o conceito de partições, os diagramas de atividade UML funcionam meramente como extensões de fluxogramas. A introdução do conceito de partição traz toda uma rica gama de representações para os diagramas UML, pois

permite representar casos de uso. No diagrama da Figura 11, uma partição é utilizada para representar o usuário e outra para representar o sistema (GUDWIN, 2021).

Figura 11: Uso de Diagrama de Atividades para Representar Casos de Uso.



Fonte: Gudwin (2021).

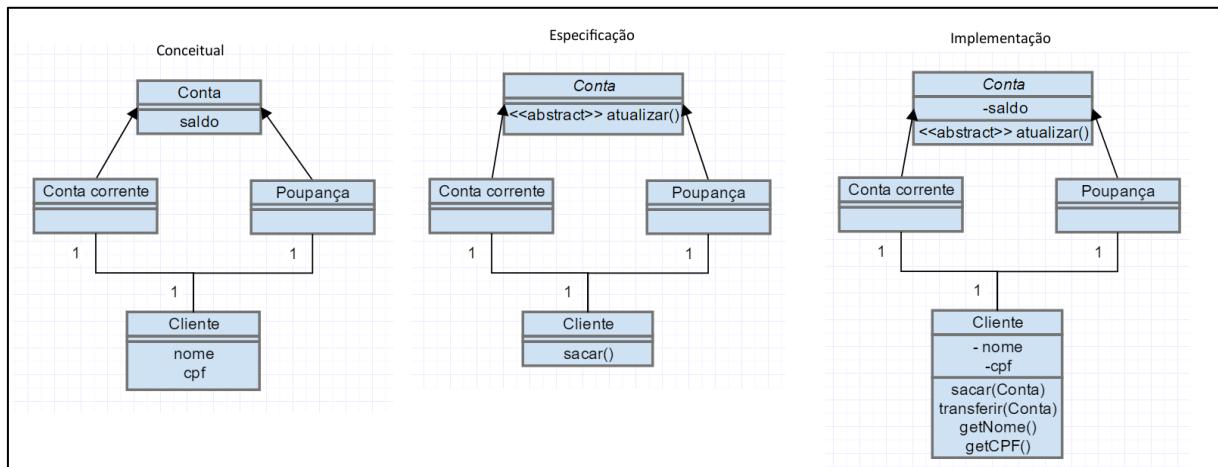
Diagramas de atividade como os da Figura 11 são utilizados para detalhar os casos de uso levantados durante a especificação do sistema. Nesses diagramas, assume-se que o usuário do sistema realiza certas ações e o sistema, em resposta, reage realizando certas tarefas. Com isso, o comportamento do sistema pode ser especificado.

2.3.4.3 Diagrama de classes

O diagrama de classe busca demonstrar como as classes funcionarão em um projeto de desenvolvimento, que utiliza a orientação a objetos, ele é composto por entidades e seus relacionamentos. Suas entidades podem ser divididas entre classes e interfaces. As classes são representadas no formato de retângulos onde a primeira linha é o nome da classe, a segunda seus atributos e a última os seus métodos. As classes também podem ser representadas concretamente ou de forma abstrata, bastando que para isso seu nome seja escrito em itálico (para classes abstratas). Para representar seus atributos e métodos como privados ou públicos pode ser adotado o sinal de “-” para privado e “+” para público (GUEDES, 2011). Os diagramas de classe são totalmente voltados a orientação a objetos, possuem três tipos de perspectivas de representação (exemplificados na Figura 12), são elas:

- **Conceitual:** os diagramas são representados em um domínio mais abstrato, sua forma de demonstrar os acontecimentos é mais voltada para o cliente.
- **Especificação:** nessa perspectiva os diagramas são voltados a um modo de representação da arquitetura do sistema, demonstrando seus principais métodos, mas não como será feita sua implementação. Voltada para pessoas que não necessariamente precisam saber como o projeto vai ser desenvolvido, mas sim somente como será feita sua estrutura.
- **Implementação:** maneira onde o diagrama demonstra os detalhes de como será a implementação, demonstrando a estrutura das classes e seus atributos. Perspectiva mais voltada aos desenvolvedores, que precisam saber os detalhes das classes (GUEDES, 2011).

Figura 12: Exemplos de Diagramas de Classes.



Fonte: Adaptado de (GUEDES, 2011).

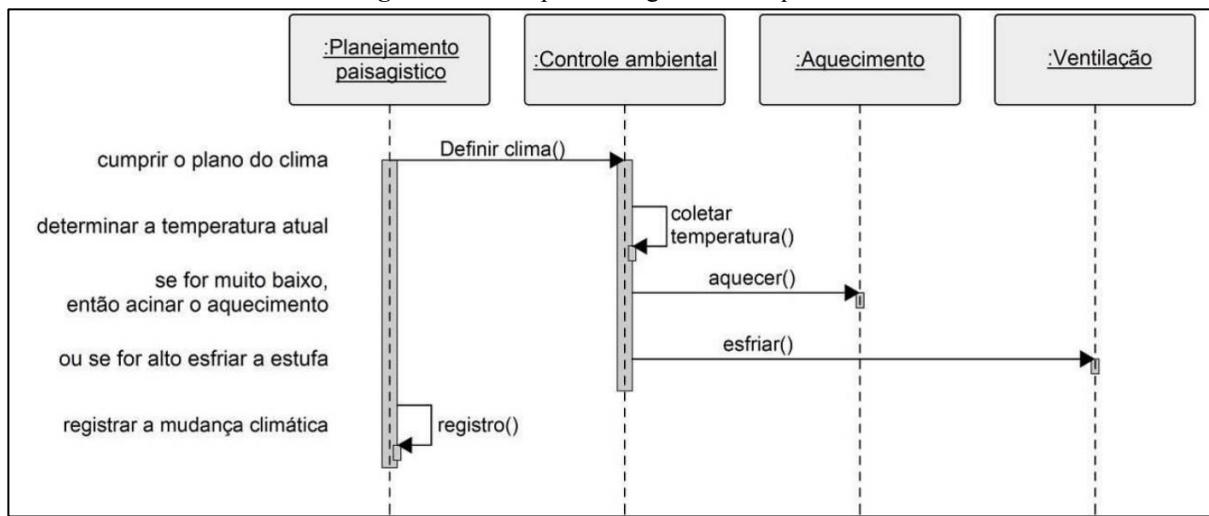
Graças a essas características os diagramas de classe são excelentes ferramentas para modelagem de sistemas, sejam eles complexos ou simples. Com o uso dos diagramas de classe os sistemas podem ser fielmente representados, de uma maneira que seja fácil de entender e ao mesmo tempo seja útil, facilitando o desenvolvimento do software (GUEDES, 2011).

2.3.4.4 Diagrama de sequência

O Diagrama de Sequência é uma das ferramentas UML usadas para representar interações entre objetos de um cenário, realizadas através de operações ou métodos (procedimentos ou funções). Ele descreve a maneira como os grupos de objectos colaboram em algum comportamento ao longo do tempo. Ele registra o comportamento de um único caso de uso e exibe os objectos e as mensagens passadas entre esses objectos no caso de uso (BOOCK, 2007).

Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento. O diagrama de sequência representa essa informação de uma forma simples e lógica. Ele dá ênfase a ordenação temporal em que as mensagens são trocadas entre os objetos de um sistema. Entende-se por mensagens os serviços solicitados de um objecto a outro, e as respostas desenvolvidas para as solicitações (BOOCK, 2007).

Figura 13: Exemplo de diagrama de sequência.



Fonte: Booch (2007).

2.3.5 RUP - Rational Unified Process

RUP é uma sigla que significa Rational Unified Process, que traduzindo para o português, quer dizer Processo Unificado da Rational. Esse termo é um processo que foi criado

pela empresa de engenharia de software, a Rational Software Corporation, com o intuito de orientar o desenvolvimento de um programa (EMANOELE, 2020).

De acordo com Jin e Liang (2016), o RUP foi criado com o intuito de ser um processo de desenvolvimento de software dirigido a casos de uso, centrado na arquitetura, iterativo e incremental. É dirigido a casos de usos, pois, é através deles que são identificados todos os atores e envolvidos, além de delimitar as funcionalidades do software. Possui um enfoque na construção de uma arquitetura executável, que seja flexível e resiliente a mudanças, uma vez que este é o ponto chave da vida do software, já que deverá suportar a inserção de vários componentes e ainda manter-se confiável e seguro. A iteração é nada mais que o ciclo completo do processo de desenvolvimento que pode ser repetido várias e várias vezes, culminando ao término de cada ciclo no lançamento de uma nova versão do software. E por fim, incremental devido a natureza do software ser muito complexa, a ponto de ser dividido em partes menores para serem construídas. Ao final de cada iteração um novo incremento do software é adicionado a arquitetura, resultando em um novo software mais completo.

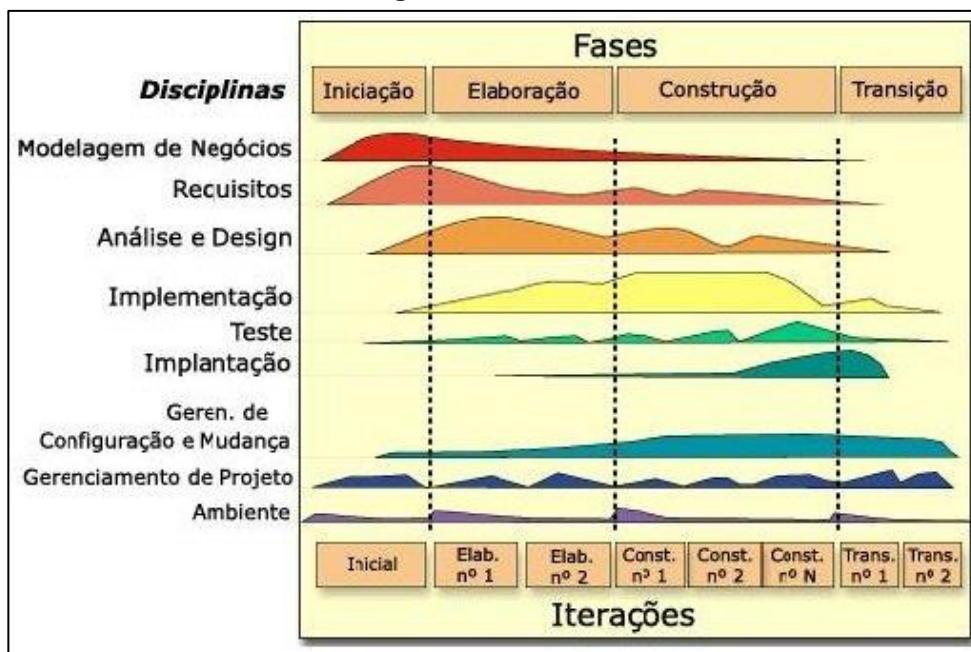
O principal objetivo do RUP é atender as necessidades dos usuários garantindo uma produção de software de alta qualidade que cumpra um cronograma e um orçamento previsíveis. O RUP é um processo dividido em quatro fases: Iniciação/Concepção, Elaboração, Construção e Transição. A Figura descreve as fases do RUP e as relaciona com as suas respectivas atividades genéricas. As fases presentes na Figura 2.13 são descritas a seguir, segundo a especificação da IBM (2004):

- **Iniciação/Concepção:** Nesta fase, identificam-se os casos de uso do sistema, seus atores e é delimitado o escopo do sistema. Nesta etapa também são analisados os riscos do projeto, estimativas de recursos necessários e planejamento dos incrementos e iterações que serão implementados. Nesta etapa um maior esforço é gasto nas atividades de modelagem de negócio e de elicitação de requisitos de software;
- **Elaboração:** Esta fase é responsável por preparar a arquitetura do software para receber os componentes. Neste ponto a deve ser criado um protótipo executável da arquitetura, onde ele será complementado a cada iteração. Esta fase compreende um maior esforço nas atividades de análise e projeto da arquitetura do software, como também na implementação, neste caso, da arquitetura;
- **Construção:** Nesta etapa são construídos e testados os componentes que serão vinculados a arquitetura do software. Nesta etapa também são criadas métricas que auxiliam no controle do tempo e dos recursos. Durante esta etapa, as atividades de

implementação e testes são mais atuantes e devem dar início as atividades de implantação;

- **Transição:** Etapa final onde o software é preparado para ser disponibilizado aos seus usuários ou é lançada uma nova versão. Ao chegar neste ponto, o software estará se preparando para lançar uma nova versão, por isso, a atividade de implantação possui maior prioridade.

Figura 14: Fases do RUP.



Fonte: Pressman e Maxim (2016).

Segundo Pressman e Maxim (2016), James Rumbaugh, Grady Booch e Ivar Jacobson começaram então a trabalhar em um conjunto de modelos que representassem melhor os conceitos de projetos Orientados a Objetos em cada uma das atividades de cada fase do RUP. Foi ai que surgiu a UML (Unified Modeling Language), que contém uma notação robusta para modelagem e desenvolvimento de sistemas orientados a objetos. Seus modelos foram utilizados nas suas respectivas atividades dentro de cada fase deste projeto.

2.4 BANCO DE DADOS

Segundo Korth (1994), um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, isto é, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, podemos afirmar que se trata de um banco de dados. Para Date (1991), um banco de dados é um sistema

computadorizado cujo principal propósito é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando quiser solicitá-las.

Existem várias opções de bancos de dados disponíveis no mercado e dois tipos principais: os bancos de dados relacionais e os não relacionais (FERNANDES, 2020).

Banco de Dados Relacional (BDR) é um método de persistência de dados estável, segura e rápida (COELHO; SARTORELLI, 2004). Os bancos de dados relacionais permitem aos seus usuários (incluindo desenvolvedores) escreverem consultas (*queries*) que não foram antecipadas por quem projetou a base de dados. Como resultado, bancos de dados relacionais podem ser utilizados por várias aplicações de infinitas maneiras diferentes. Isto tem tornado os bancos de dados relacionais muito populares em todas as esferas do desenvolvimento de softwares (COSTA, 2011).

O modelo relacional armazena as informações em tabelas unidimensionais, com linhas e colunas, onde cada linha da tabela representa um registro no banco de dados. É possível aos usuários criar, modificar e consultar dados dentro de um banco de dados através da linguagem SQL (Structured Query Language – Linguagem de Consulta Estruturada, em português) (COSTA, 2011).

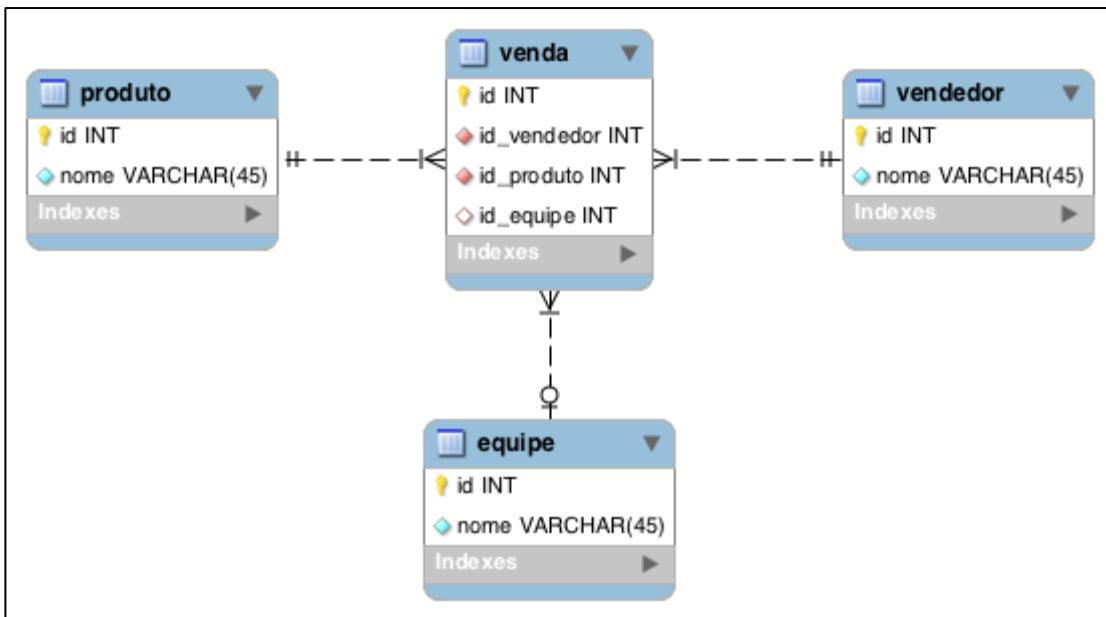
Dentro de um banco de dados podem existir uma ou diversas tabelas, cada uma pode armazenar dados referentes a uma determinada funcionalidade do sistema, por exemplo, em um banco de dados que armazena informações sobre produtos e clientes pode existir uma tabela para armazenar as informações do cliente e outra para armazenar os dados dos produtos. Para representar todas as tabelas que serão utilizadas em um sistema, podemos utilizar um modelo de entidade relacionamento (FERNANDES, 2020).

Em engenharia de software, um modelo entidade relacionamento (MER) é um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional. Os principais componentes dos Modelos Entidade-Relacionamento (MER) são as entidades (coisas, objetos) suas relações e armazenamento em bancos de dados (FERNANDES, 2020). Um exemplo de modelo entidade relacionamento pode ser visto na Figura 15.

No exemplo da Figura 15, podemos ver as 4 tabelas que compõem o banco de dados: produto, venda, vendedor, e equipe. Cada uma possui suas variáveis (que dentro do banco de dados será uma coluna da tabela), listadas pelo nome e tipo. Dessa forma, é possível ter uma visão geral do sistema e construir o banco de dados de uma maneira mais fácil e intuitiva. Cada

tabela em um banco de dados relacional contém um ou mais dados em colunas, e cada linha, também chamada de registro, contém uma instância exclusiva de dados ou chave para os dados definidos pelas colunas. Cada tabela normalmente possui uma coluna de chave primária, um registro único dentro da tabela para identificar os registros. O relacionamento entre tabelas pode ser definido através do uso de chaves estrangeiras – um campo em uma tabela que se vincula à chave primária de outra tabela (FERNANDES, 2020).

Figura 15: Exemplo de modelagem de banco de dados para uma loja.



Fonte: Autor.

Além dos bancos de dados relacionais, existem os bancos de dados não relacionais. NoSQL (Not Only SQL) é o termo utilizado para banco de dados não relacionais de alto desempenho, onde geralmente não é utilizado o SQL como linguagem de consulta. O NoSQL foi criado para ter uma performance melhor e uma escalabilidade mais horizontal para suprir necessidades onde os bancos relacionais não são eficazes. Os dois principais tipos de bancos de dados NoSQL são Documento e Chave-valor (FERNANDES, 2020).

No tipo Documento os dados são armazenados como documentos. Os documentos podem ser descritos como dados no formato de chave-valor, como por exemplo, o padrão JSON (*JavaScript Object Notation*). Normalmente possuem poderosas linguagens de consulta, esses bancos de dados de documentos são ótimos para usos gerais. Eles podem ser facilmente escalados horizontalmente para acomodar grandes volumes de dados (FERNANDES, 2020).

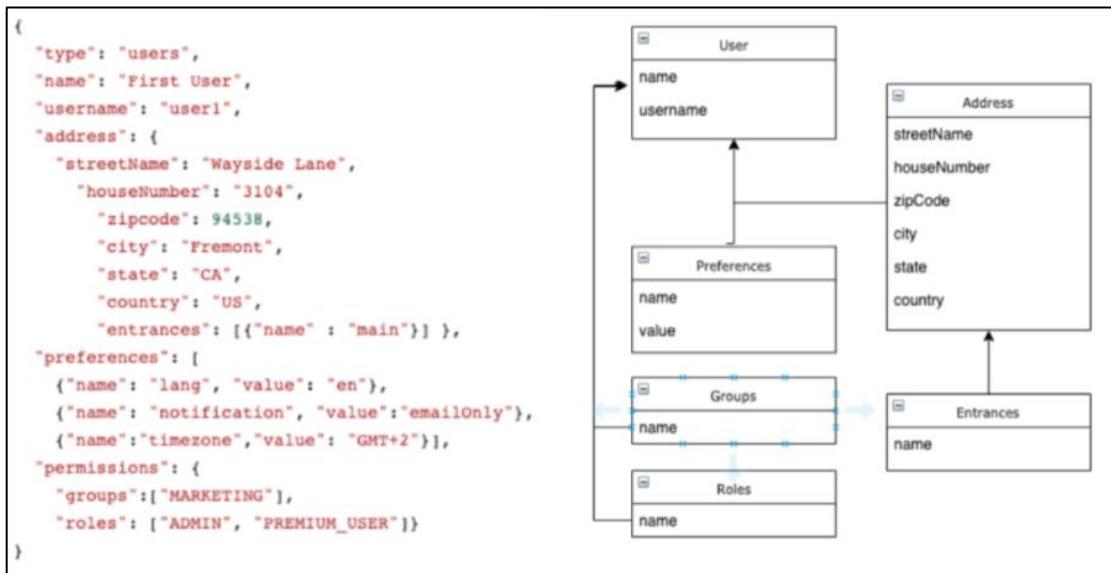
No tipo chave-valor cada item contém chaves e valores, esses valores podem ser qualquer tipo de dado, um texto, um número, um JSON e eles podem ser recuperados fazendo referência a sua chave, fazendo com que sua consulta seja muito simples. Esses bancos são

ótimos para quando você precisa armazenar grandes quantidades de dados, mas não precisa executar consultas complexas neles (FERNANDES, 2020).

Os modelos de dados NoSQL permitem por exemplo, que os dados relacionados sejam feitos em uma única estrutura de dados. Diferentemente dos bancos relacionais, a estrutura de dados não precisa ser definida previamente, portanto, uma mesma “tabela” pode ter dados com propriedades diferentes (FERNANDES, 2020).

A Figura 16 faz uma comparação entre um banco de dados NoSQL no padrão JSON (esquerda) e um banco de dados SQL representado por um modelo entidade relacionamento (direita).

Figura 16: Comparação entre o banco de dados NoSQL e SQL.



Fonte: Rosa, 2019.

Em um projeto, ambas as soluções podem ser utilizadas para diferentes casos de uso. Por isso, o mais comum em soluções escalares de sucesso é a utilização de uma arquitetura híbrida, aproveitando o melhor dos dois modelos (FERNANDES, 2020).

Neste trabalho, utilizou-se um banco de dados SQL (SQLite) para armazenar os dados localmente no aparelho, e um banco de dados NoSQL (*Firebase Realtime Database*) para armazenar um backup dos dados na nuvem. Estes dois bancos de dados serão introduzidos nas subseções a seguir.

2.4.1 SQLite

O SQLite é um banco de dados relacional que, diferentemente de outras ferramentas do tipo, não armazena informações em um servidor. Essa independência acontece porque ele consegue colocar os seus arquivos dentro de si próprio. Essa base de dados é de código aberto

e gratuita, sendo muito utilizada em aplicações mobile, com foco no sistema Android, que vem com implementação de banco de dados SQLite embutida (SOUZA, 2020).

Ao mesmo tempo em que dispensa um servidor, o SQLite também não demanda nenhum tipo de configuração. Assim, seu uso como base de dados para diferentes tipos de aplicações se torna, além de fácil, mais fluído, dinâmico e leve. Todo esse potencial tem colocado o SQLite como a 9^a opção de base de dados mais utilizada em todo o mundo. Ele é completo podendo ter várias tabelas, índices, gatilhos (triggers) e visões, e isso tudo estando contido em um único arquivo de disco, o que otimiza o gasto com espaço em memória, podendo facilmente ser embarcado em dispositivos como telefones celulares (SQLITE, 2017).

2.4.2 Firebase Realtime Database

O *Firebase* é uma plataforma do Sistema Google que contém várias ferramentas e uma infraestrutura para auxiliar desenvolvedores de aplicações web e de dispositivos móveis na construção de software de alta qualidade e performance (SMYTH, 2017).

No *Firebase Realtime Database* é possível criar aplicativos avançados e colaborativos, ao conceder acesso seguro ao banco de dados diretamente do código do cliente. Os dados são mantidos localmente e, mesmo off-line, os eventos em tempo real continuam sendo acionados, proporcionando uma experiência responsiva ao usuário final. Quando o dispositivo recupera a conexão, o *Realtime Database* sincroniza as alterações feitas nos dados locais com as atualizações remotas que ocorreram enquanto o cliente estava off-line, mesclando qualquer conflito automaticamente (FIREBASE, 2021).

O *Firebase Realtime Database* é um banco de dados hospedado na nuvem. Os dados são armazenados como JSON e sincronizados em tempo real com todos os clientes conectados. O *Realtime Database* é um banco de dados NoSQL e, por isso, tem otimizações e funcionalidades diferentes de um banco de dados relacional, ele possibilita uma ótima experiência em tempo real que atende a milhões de usuários sem comprometer a capacidade de resposta, ele oferece uma linguagem de regras flexíveis baseadas em expressão, denominadas regras de segurança, para definir como os dados são estruturados e quando podem ser lidos e gravados. Quando aplicativos são criados para Android utilizado este banco de dados em tempo real, todos os clientes compartilham uma instância do *Realtime Database* e recebem automaticamente atualizações com os dados mais recentes (FIREBASE, 2021).

3 METODOLOGIA

3.1 VISÃO GERAL DO PROTÓTIPO

O objetivo deste trabalho é desenvolver uma aplicação *mobile* para o autogerenciamento do diabetes mellitus, contemplando os principais recursos existentes nos aplicativos disponíveis no mercado, com uma abordagem simples e acessível, focada no autocuidado e na usabilidade.

O Rational Unified Process (RUP) foi escolhido como metodologia para implementar a solução, sendo assim, as atividades metodológicas foram organizadas dentro das fases do RUP: Concepção, Elaboração, Construção e Transição. O Quadro 3 exibe as atividades metodológicas dentro de cada fase do RUP, que por sua vez, são compostas pelas tarefas necessárias para o cumprimento daquela atividade. A coluna autores apresenta os autores dos trabalhos nos quais as tarefas foram baseadas.

Quadro 3: Atividades metodológicas e tarefas para o desenvolvimento do aplicativo.

Fase RUP	Atividades	Tarefas	Autor(es)
Iniciação/Concepção	Levantamento de Requisitos	Revisão de Literatura; Estudo de aplicações semelhantes; Levantamento de requisitos Funcionais e Não-Funcionais; Validação dos requisitos.	Cheah et al. (2017); Badham et al. (2018).
Elaboração	Análise e Projeto de Arquitetura	Gerar Diagrama de Atividades (UML); Gerar Diagrama de Use case (UML); Gerar Diagrama de Classe (UML).	Dinsoreanu, Salomie e Pusztai (2002); Guedes e Vicari (2010); Marco e Pace (2013).
Construção	Desenvolvimento e Testes	Desenvolver o aplicativo; Implementar casos de testes.	Marco e Pace (2013); Russell, Doyle e Collier (2018).
Transição	Implantação e Publicação	Gerenciar versionamento; Entregar o primeiro protótipo instalável; Receber <i>feedbacks</i> , implementar melhorias e entregar um novo protótipo; Publicar o aplicativo finalizado na <i>Play Store</i> .	Sommerville (2011); Pressman e Maxim (2016); Russell, Doyle e Collier (2018).

Fonte: Autor.

As fases do RUP, suas respectivas atividades e tarefas enumeradas no quadro 3 serão explicadas nas sessões a seguir.

3.2 LEVANTAMENTO DE REQUISITOS

Após revisão de literatura, apresentada na sessão 2, e estudo de aplicações semelhantes (Quadro 2), foi possível fazer o levantamento de requisitos funcionais e não-funcionais do aplicativo. O quadro 4 apresenta os requisitos funcionais elicitados durante a etapa de concepção do projeto.

Quadro 4: Requisitos funcionais do aplicativo.

Nome do requisito	Prioridade
Criar conta – Fazer Login	Essencial
Registrar glicemia	Essencial
Registrar medicação	Essencial
Registrar alimentação	Essencial
Registrar atividades físicas	Essencial
Gerar gráfico do nível de glicemia	Essencial
Criar lembretes e notificações	Essencial
Disponibilizar nível de glicose dos alimentos para consulta (Índice glicêmico e carga glicêmica)	Prioritário
Disponibilizar informações nutricionais e dietas para consulta	Prioritário
Realizar planejamento alimentar	Prioritário
Registrar outro tipo de informação/bloco de notas	Prioritário
Exportar relatórios	Prioritário
Registrar dados pessoais / "ficha de saúde"	Prioritário
Solicitar assistência profissional	Desejável

Fonte: Autor.

Os requisitos não-funcionais podem ser vistos como atributos de qualidade, de desempenho, de segurança ou como uma restrição geral de um sistema. Estes requisitos aumentam a qualidade do software e o torna utilizável para o usuário final. O Quadro 5 exibe todos os requisitos não-funcionais identificados.

Quadro 5: Requisitos não-funcionais do aplicativo.

Nome do requisito	Prioridade	Tipo
Interface simples e funcional	Essencial	Usabilidade
Sem conexão persistente com a internet	Prioritário	Usabilidade

Os dados serão sincronizados com um servidor de dados na nuvem.	Prioritário	Segurança
Login simplificado com <i>Single Sign On</i> .	Prioritário	Segurança
Disponibilizar para o maior número de versões do Android sem prejudicar o desempenho/layout.	Prioritário	Implantação
Publicar o aplicativo no Google Play Store	Desejável	Implantação
Uso de câmeras e GPS.	Desejável	Hardware

Fonte: Autor.

Após a identificação dos requisitos funcionais e não funcionais, seguiu-se para a fase de elaboração, onde foram definidos os casos de uso do app e a construção da sua arquitetura através dos diagramas UML, explicadas nas sessões abaixo.

3.3 ANÁLISE E PROJETO DE ARQUITETURA

Nessa atividade, primeiramente foi realizada a análise dos requisitos elencados, que serviram como alicerce para a elaboração do projeto de arquitetura. Em seguida, foram identificados os principais Atores e os casos de uso do aplicativo em uma visão de alto nível. O Quadro 6 mostra os principais Atores identificados durante a fase de análise dos requisitos.

Quadro 6: Principais atores do aplicativo.

Atores	Descrição
Paciente	Pessoa diagnosticada com Diabetes Mellitus.
Cuidador	Pessoa íntima do paciente que pode ser um familiar ou profissional de saúde, que cuida dele e está pronto para ajudá-lo em casos de emergência.
Profissional de saúde	Profissional de saúde responsável por acompanhar o tratamento do paciente ou central de atendimento responsável por fornecer o serviço de assistência.

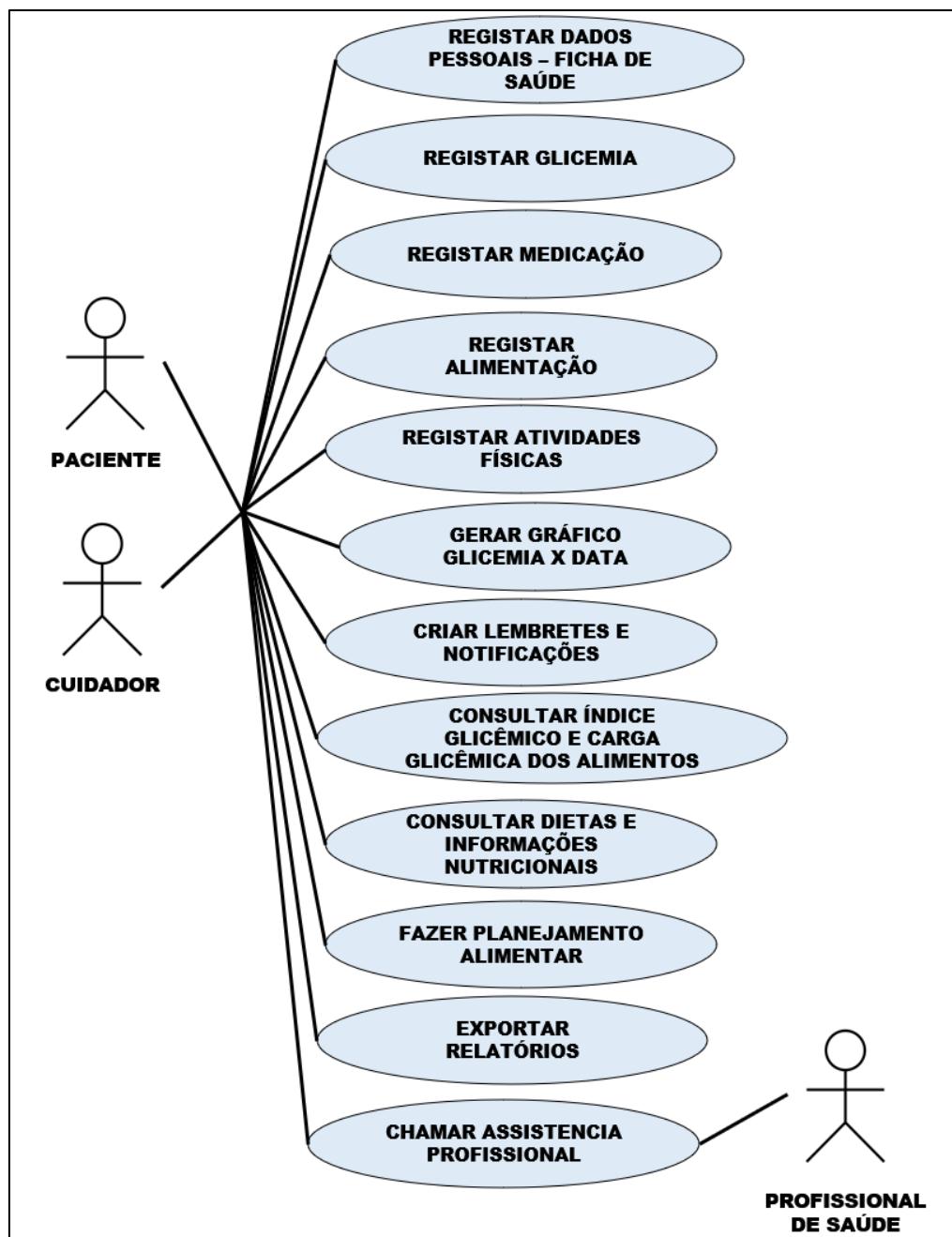
Fonte: Autor.

Os atores são a representação do principal perfil de usuário do aplicativo. O ator Profissional de saúde não representa um usuário em específico, mas sim, um provedor de

serviço ao qual tem uma função passiva na aplicação, pois ele apenas é contatado pelo paciente ou, por exemplo, apenas recebe os relatórios via e-mail caso o paciente os queira enviar.

Os Atores são utilizados para mostrar onde ocorrerá cada interação dos usuários com as funcionalidades e qual o tipo de usuário que irá interagir com ela, no diagrama de casos de uso. Eles podem ser qualquer tipo de pessoa ou sistema, que são representados por uma Figura "palito", esta relação é feita ligando uma linha reta entre o ator e o balão da funcionalidade. Estes Atores apresentados no Quadro 5, estão vinculados aos principais casos de uso do aplicativo que são representados pelo Diagrama de Casos de uso, na Figura 17.

Figura 17: Diagrama de casos de uso do aplicativo.



Fonte: Autor.

Dentro da metodologia RUP, cada fase do desenvolvimento do aplicativo pode ser dividida em iterações, que é um loop no desenvolvimento que resultará em uma entrega. Devido à grande quantidade e complexidade dos casos de uso em relação ao tempo e quantidade de desenvolvedores no projeto, o desenvolvimento foi segmentado em quatro iterações com o objetivo de poder criar ao final da iteração um protótipo instalável. O Quadro 7 mostra os casos de uso a serem desenvolvidos em cada iteração.

Quadro 7: Iterações do aplicativo desenvolvido

Iterações	Casos de uso
1º iteração	<ul style="list-style-type: none"> - Criar conta – Fazer Login; - Registrar glicemia; - Registrar medicação; - Registrar alimentação; - Gerar gráfico do nível de glicemia x data.
2º iteração	<ul style="list-style-type: none"> - Registrar atividades físicas; - Disponibilizar nível de glicose dos alimentos para consulta (Índice glicêmico e carga glicêmica); - Criar lembretes e notificações; - Registrar dados pessoais / "ficha de saúde".
3º iteração	<ul style="list-style-type: none"> - Registrar outro tipo de informação/bloco de notas; - Disponibilizar informações nutricionais e dietas para consulta; - Realizar planejamento alimentar.
4º iteração	<ul style="list-style-type: none"> - Exportar relatórios; - Solicitar assistência profissional.

Fonte: Autor.

Este trabalho focou em mostrar o desenvolvimento da 1º e da 2º Iteração, pois elas demonstram o fluxo completo do aplicativo e preparam a arquitetura para os outros casos de uso, de forma a diminuir o tempo de desenvolvimento deles.

Após definir os casos de uso para cada iteração, foi dada sequência a fase de Elaboração, onde as tarefas ‘gerar diagrama de atividades’ e ‘gerar diagrama de casos de uso’ foram desenvolvidas para as iterações 1 e 2, que podem ser visualizados na sessão de resultados.

Em seguida, iniciou-se as próximas fases do RUP: Construção e Transição, explicadas nos tópicos abaixo.

3.4 DESENVOLVIMENTO, TESTES, IMPLANTAÇÃO E PUBLICAÇÃO

Para a fase de construção (Desenvolvimento e Testes) foram utilizadas algumas ferramentas e tecnologias de desenvolvimento de software mobile. Antes de iniciar o desenvolvimento foi necessário organizar e configurar o projeto como um todo. O projeto foi criado com a estrutura padrão de um projeto Android através da IDE de desenvolvimento Android Studio, criado e distribuído de forma gratuita pela empresa Google. A linguagem de programação Java foi utilizada no *backend* da aplicação, enquanto a linguagem de marcação XML manipula seu *frontend* (DEITEL; DEITEL; DEITEL, 2015).

Para o armazenamento de dados, estabeleceu-se que todas as informações da aplicação seriam armazenadas utilizando-se o banco de dados SQLite, que seria responsável por manter o registro das informações localmente nos dispositivos móveis, e o *Firebase Realtime Database* seria necessário para o armazenamento das informações em nuvem. Desta forma, o SQLite é responsável por registrar os dados que serão processados durante a execução da aplicação e o *Realtime Database* foi utilizado para manter um backup caso haja necessidade de recuperação dos dados salvos no SQLite.

Após definir as ferramentas e especificações de arquitetura, foi construído o código do aplicativo Android, denominado MOGLIE. Neste projeto, a fase de transição (Implantação e Publicação) não foi realizada, pois o aplicativo não foi publicado na Google Play Store. Até a data de publicação deste trabalho, apenas as iterações 1 e 2 foram desenvolvidas. O protótipo da 1º iteração foi instalado e testado em um celular com o sistema operacional Android. O protótipo da 2º iteração foi instalado e testado no emulador do Android, dentro da IDE Android Studio. Os resultados do desenvolvimento serão mostrados no capítulo seguinte.

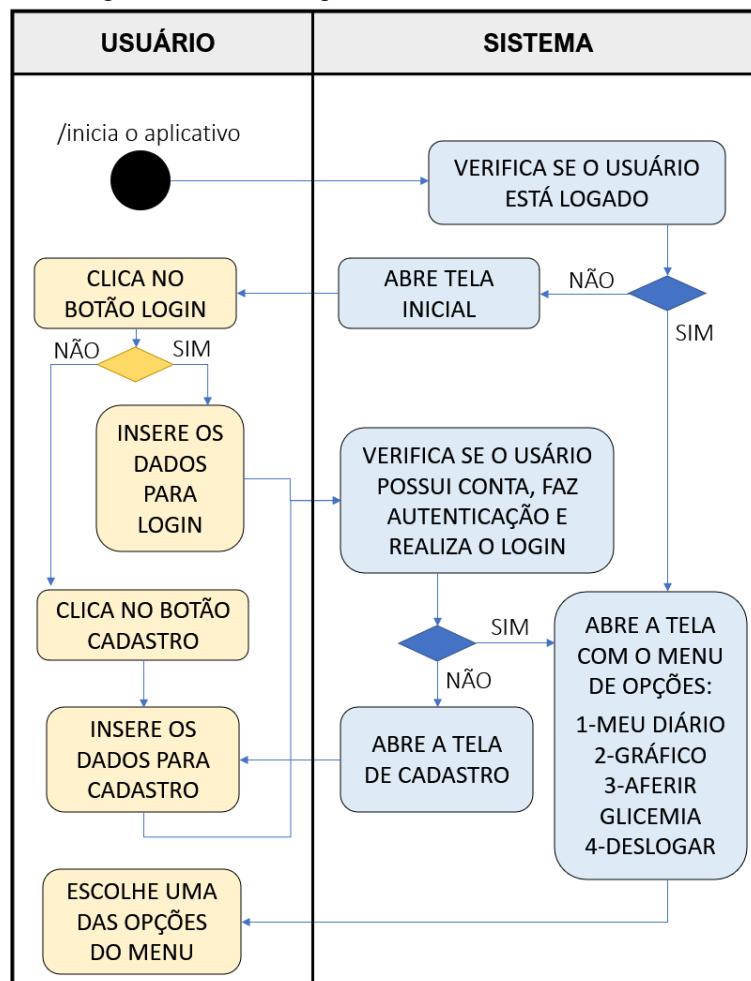
4 RESULTADOS

Nas subseções abaixo serão mostrados os resultados obtidos durante o desenvolvimento do aplicativo MOGLIE. O item 4.1 A mostra os resultados do desenvolvimento da 1º iteração do aplicativo, o item 4.2 mostra os resultados do desenvolvimento da 2º iteração, o item 4.3 mostra os resultados atingidos com os bancos de dados utilizados e o item 4.4 mostra o desempenho do aplicativo.

4.1 1º ITERAÇÃO

Tomando por base os casos de uso da 1º Iteração visto no Quadro 6, foi projetado um Diagrama de Atividades, para facilitar o desenvolvimento. A Figura 18 mostra o diagrama de atividades da 1º Iteração, para o caso de uso Criar Conta – Fazer Login.’

Figura 18: Diagrama de atividades para o caso de uso “Criar Conta – Fazer Login”.

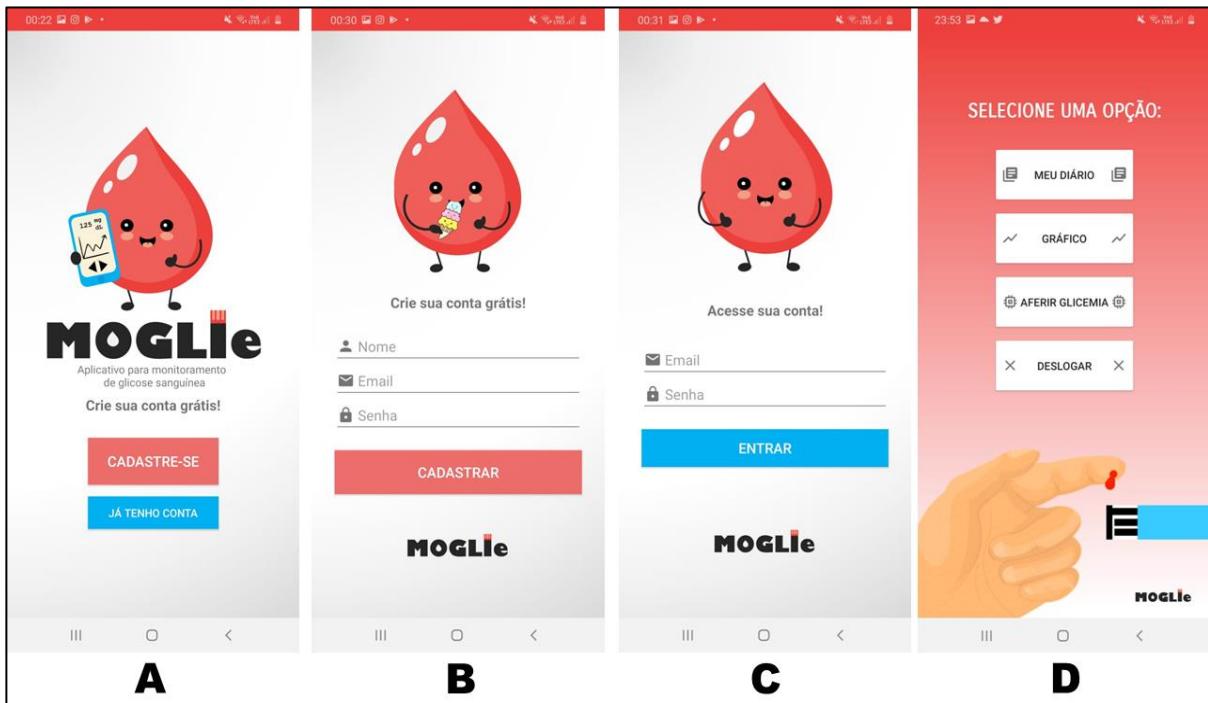


Fonte: Autor.

Como demonstrado na Figura 18, a última atividade realizada pelo sistema foi abrir o menu, onde o usuário pode escolher uma das opções disponíveis: “MEU DIÁRIO”, “GRÁFICO”

, “AFERIR GLICEMIA”, e “DESLOGAR”. Este menu, assim como as telas inicial, de cadastro e de login, estão representados pela Figura 19.

Figura 19: Tela inicial do aplicativo (A), tela de cadastro (B), tela de login (C) e menu de opções (D).



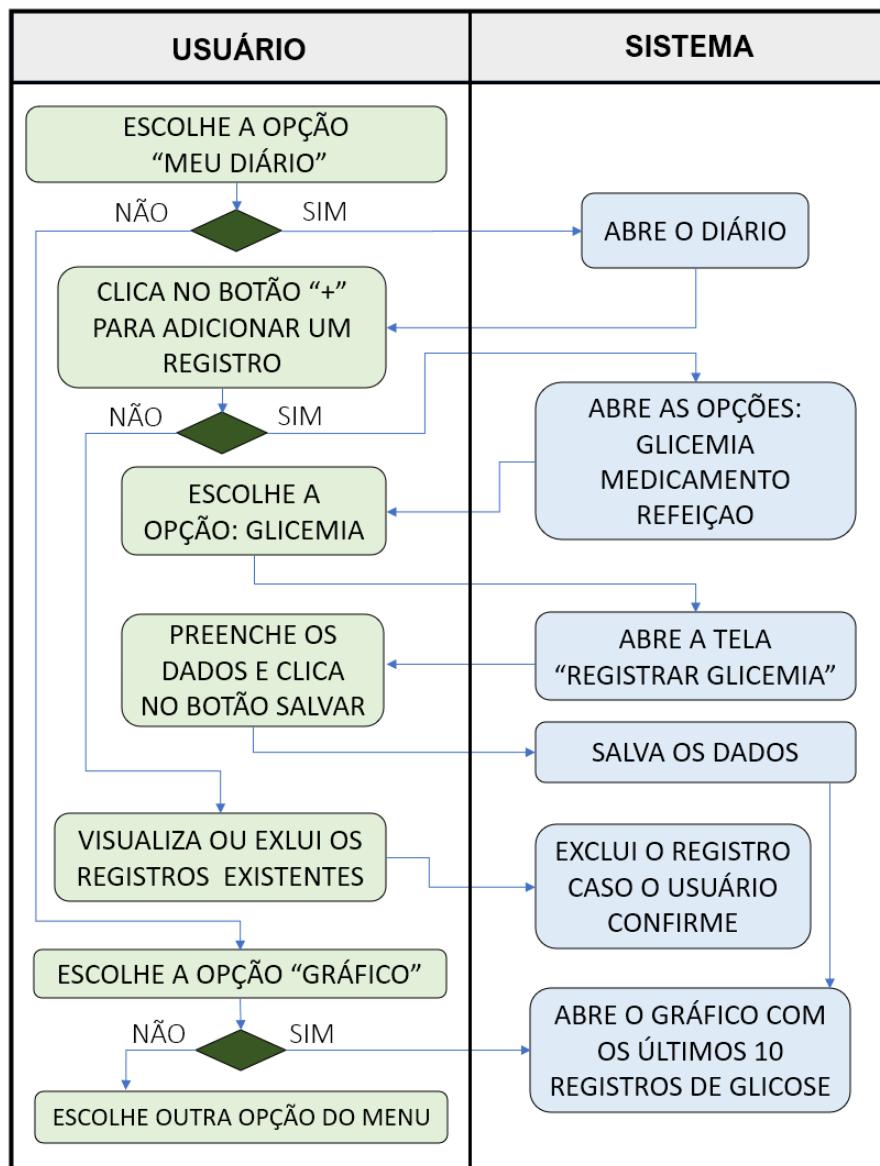
Fonte: Autor.

A Figura 19-A é a primeira tela exibida ao usuário após instalar o aplicativo MOGLIE em seu smartphone, ela contém dois botões, “Cadastre-se” e “Já tenho conta”, que direcionam o usuário para as telas de cadastro e login, exibidas na Figura 19-B e 19-C, respectivamente. Para efetuar o cadastro é necessário preencher as caixas de texto com o nome, e-mail e senha. Caso o usuário já tenha cadastro, ele pode efetuar o login preenchendo com seu e-mail e senha cadastrados. A Figura 19-D mostra a tela do menu, que contém 4 botões: “Meu Diário”, “Gráfico”, “Aferir Glicemia” e “Deslogar”.

Os casos de uso “Registrar medicação”, “Registrar refeição” e “Registrar Glicemia” podem ser acessados pelo usuário após clicar na opção “Meu Diário”. O caso de uso “Gerar gráfico do nível de glicemia x data” pode ser acessado ao clicar no botão “Gráfico” do menu. A opção “Deslogar” efetua o *logoff* do usuário e a opção “Aferir glicemia” é um atalho para a tela de registro de dados glicêmicos. Optou-se pela criação deste atalho para facilitar o tráfego dos usuários que querem um acesso rápido para registrar seu nível de glicose no aplicativo.

A Figura 20 demonstra o diagrama de atividades para as opções “Meu Diário” e “Gráfico”. As opções “Aferir Glicemia” e “Deslogar” do menu não foram representadas em um diagrama de atividades, devido a sua simplicidade.

Figura 20: Diagrama de atividades para os casos de uso "Registrar Glicemia" e "Gerar Gráfico do nível de glicemia x data".



Fonte: Autor.

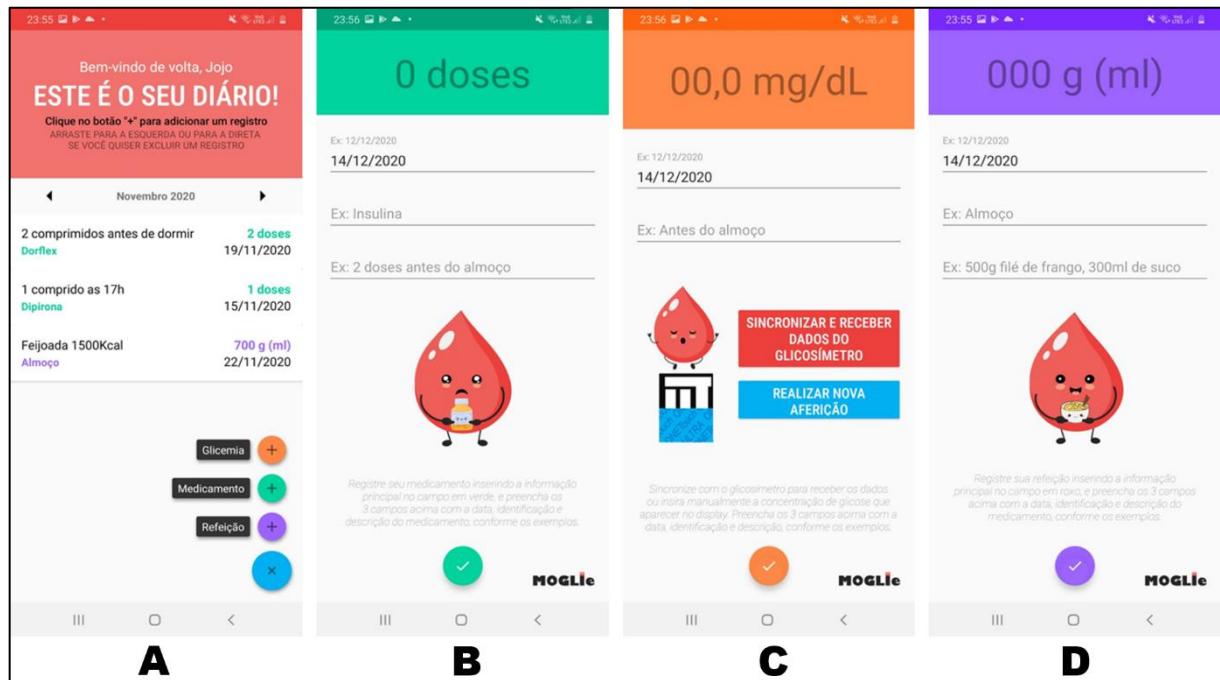
O diagrama de atividades acima mostra todas as tarefas que devem ser executadas pelo usuário e pelo sistema para que a pessoa consiga fazer um registro da sua glicemia e acessar o gráfico para visualizar os 10 últimos registros. Os casos de uso "Registrar medicação" e "Registrar refeição" não foram apresentados no diagrama de atividades pois são bem semelhantes ao do caso de uso "Registrar Glicemia", que está ilustrado no diagrama.

As Figuras 21 e 22 e mostram todas as telas desenvolvidas para os casos de uso da 1º iteração, exceto o caso "Criar conta – Fazer login", que foi demonstrado na Figura 20.

Após clicar no botão "Meu Diário", o usuário é direcionado para a tela exibida na Figura 21-A, e consegue acessar todas medidas registradas no mês que está selecionado. Para navegar entre os meses e acessar as anotações de outro mês específico, o usuário tem que clicar nas setas

esquerda ou direita, ao lado do nome do mês. Para excluir um registro, basta arrastar para esquerda ou para direta e confirmar a exclusão. Para acrescentar um novo registro no diário, o usuário precisa clicar no botão flutuante “+” no canto inferior direito e escolher uma das três opções: Glicemia, Medicamento ou Refeição (Figura 21-A).

Figura 21: tela “Meu Diário” (A), tela de registro de Medicamento (B), tela de registro de Glicemia (C) e tela de registro de Refeição (D)



Fonte: Autor

Ao selecionar a opção “Medicamento”, o usuário é direcionado para tela apresentada na Figura 21-B. Caso o usuário clique na opção “Glicemia”, é direcionado para a tela apresentada na Figura 21-C. E caso opte por “Refeição”, é direcionado para a tela apresentada na Figura 21-D. Uma vez dentro de uma dessas opções, o usuário pode preencher os dados solicitados: valor, data, palavra-chave (espaço sugerido para escrever uma descrição mais objetiva: nome do remédio, tipo de refeição, etc.) e descrição (espaço reservado para uma anotação, observação ou algum detalhe que o usuário deseje escrever sobre aquele item) e clicar no botão salvar, no centro inferior da tela.

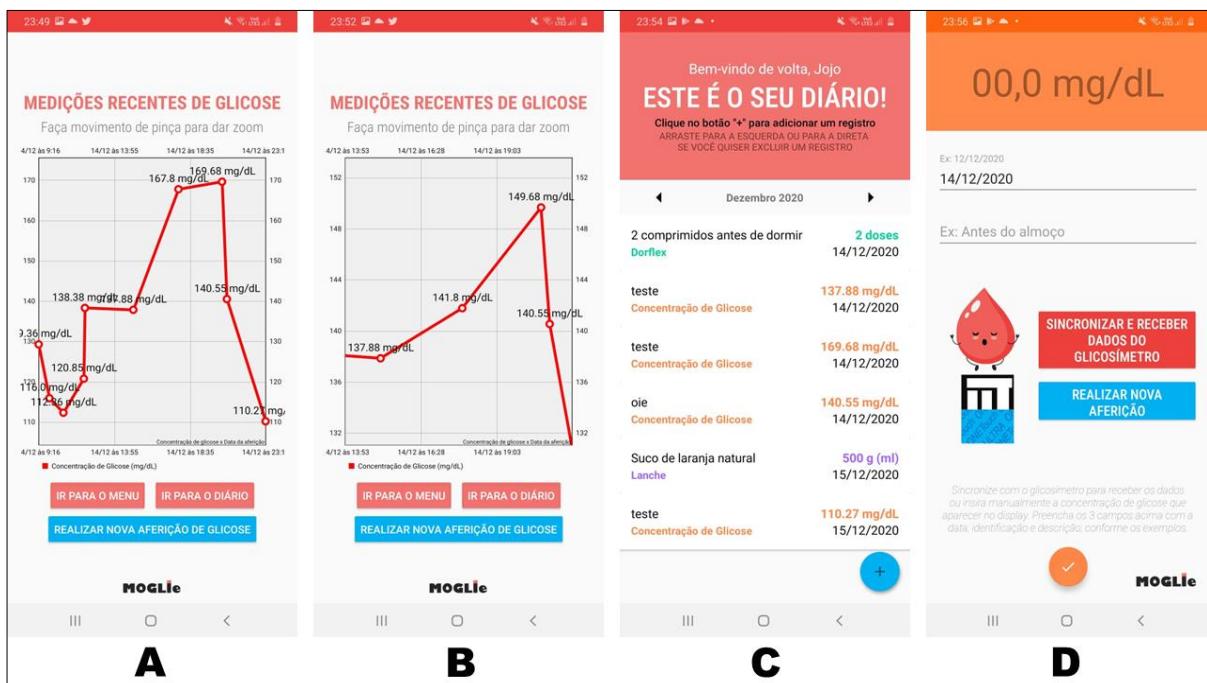
Cada caixa de texto do formulário possui um texto de exemplo, para simplificar o preenchimento do usuário, além disso, cada categoria é representada por uma cor diferente dentro do diário, para facilitar a visualização. Na Figura 21-A, podemos ver que o diário mostra o mês de novembro de 2020 com 3 registros, dois de medicamento (apresentados na cor verde) e um de refeição (apresentado na cor roxa).

A tela de registro de glicemia (Figura 21-C) não possui uma caixa de texto para palavra-chave. Como todos os registros dessa categoria serão obrigatoriamente de níveis de glicemia,

ao preencher com o valor, data e descrição, o registro é salvo com a palavra-chave “Concentração de Glicose”.

A Figura 22-C apresenta anotações do mês de dezembro de 2020, com vários registros de glicemia, onde é possível ver a palavra-chave “Concentração de glicose” destacada em laranja em todos os registros.

Figura 22: tela da opção “Gráfico” (A), gráfico com zoom (B), diário com registros de glicemia (C) tela de registrar glicemia (D)

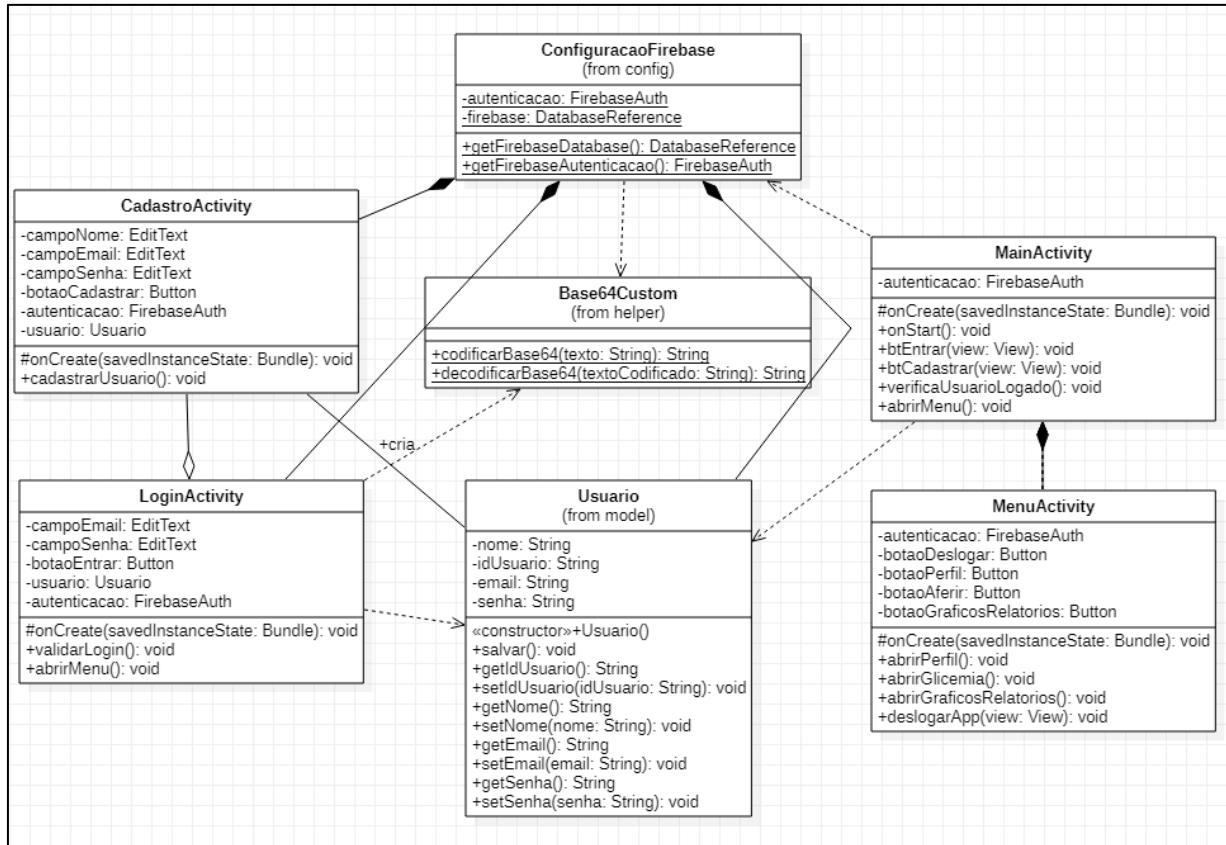


Fonte: Autor.

A Figura 22 também nos mostra o gráfico dos dados glicêmicos em relação a data, que pode ser acessado ao selecionar a opção “Gráfico” do menu. Além disso, após realizar um registro de glicemia, o usuário é automaticamente direcionado para esta tela. Uma vez dentro dessa funcionalidade, o usuário pode visualizar os últimos 10 registros realizados, podendo dar zoom e navegar pelo gráfico por meio do movimento de pinça e arraste com os dedos (Figura 22-B). A tela de gráficos também oferece três opções ao usuário: “Ir para o menu”, “Ir para o diário” e “Realizar uma nova aferição de glicose”.

As Figuras 20, 21 e 22 mostraram todas as telas da 1^a iteração do aplicativo MOGLIE. Para elaborar sua arquitetura e facilitar o desenvolvimento, utilizou-se o diagrama de classes para estruturar as classes, atributos e métodos da aplicação. A Figura 23 apresenta o Diagrama de Classes para a 1^o iteração, para as classes que envolvem cadastro de usuário, login e acesso ao menu inicial.

Figura 23: Diagrama de Classes das classes que envolvem cadastro de usuário, login e acesso ao menu do aplicativo.



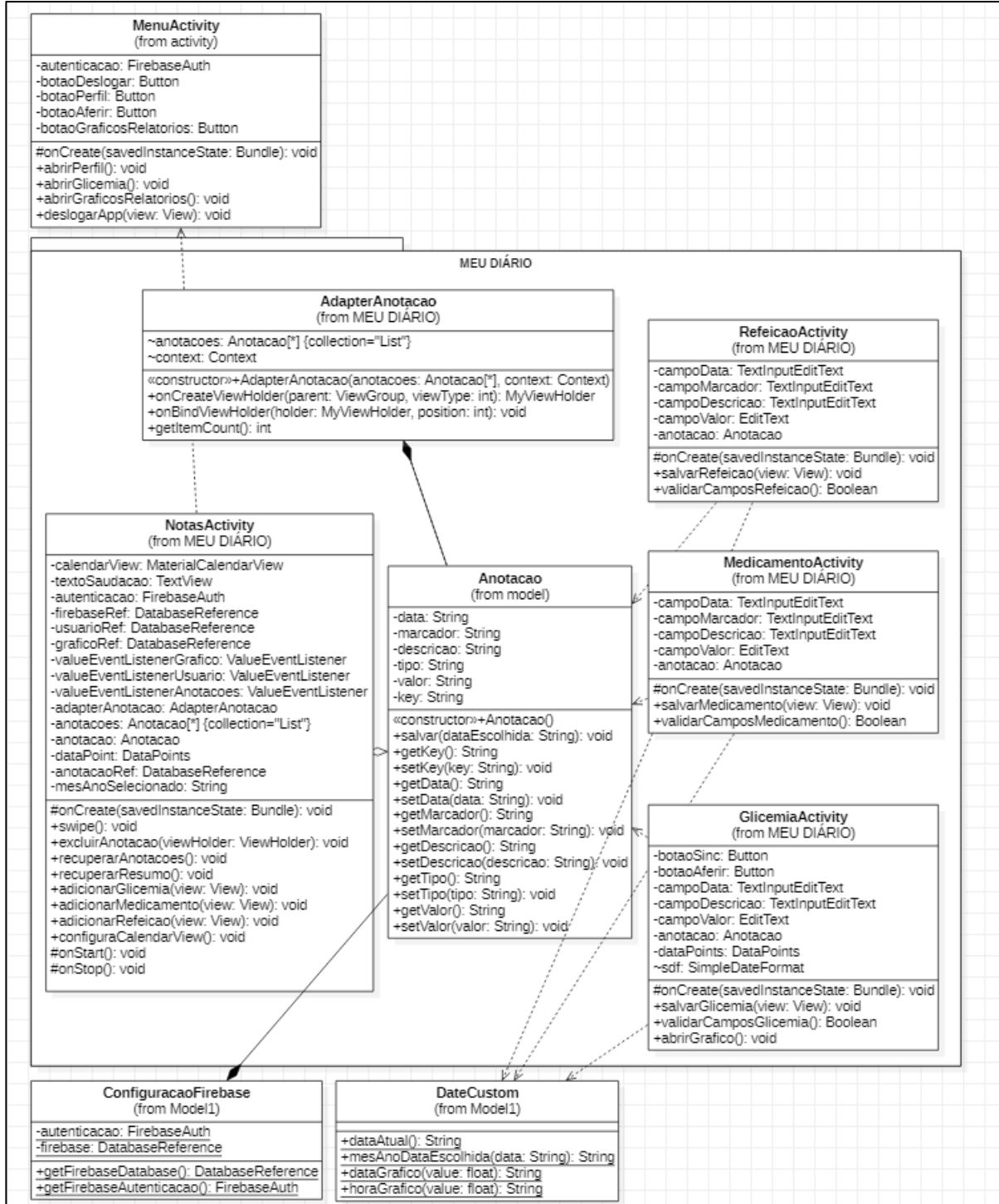
Fonte: Autor.

No diagrama da Figura 23 é possível visualizar as classes “CadastroActivity” e “LoginActivity”, que possuem os atributos e métodos necessários para o Cadastro e Login de usuários (Figura 19-B e C). A classe “Usuário” possui os atributos e métodos para a criação de novos usuários, as classes “Base64Custom” e “ConFiguraçãoFirebase” são classes de apoio que realizam a criptografia dos dados e o armazenamento deles no banco de dados *Firebase Realtime Database*, respectivamente. A classe *MenuActivity* foi criada para implementar o Menu, ela contém os métodos para abrir as telas do aplicativo, direcionando o usuário para a funcionalidade desejada ao clicar no seu respectivo botão (Figura 19 – D).

A classe “MainActivity” é a primeira classe que será executada após a instalação do protótipo, ela mostra a página inicial do aplicativo (Figura 19 – A) e possui os métodos que direcionam o usuário para o cadastro ou para login, após ele selecionar uma destas duas opções. Além disso, a classe “MainActitiviy” possui o método “verificaUsuarioLogado()”, que verifica se o usuário efetuou o *logoff* da última sessão, e, caso negativo, o direciona para o Menu, evitando que o usuário efetue o login toda vez que abrir o aplicativo. Essa funcionalidade é resultado da implementação do requisito não-funcional “Login simplificado com *Single Sign On*”.

As classes envolvidas no desenvolvimento da funcionalidade “Meu Diário” podem ser visualizadas na Figura 24, abaixo.

Figura 24: Diagrama de classes das classes que envolvem a funcionalidade “Meu Diário” do aplicativo



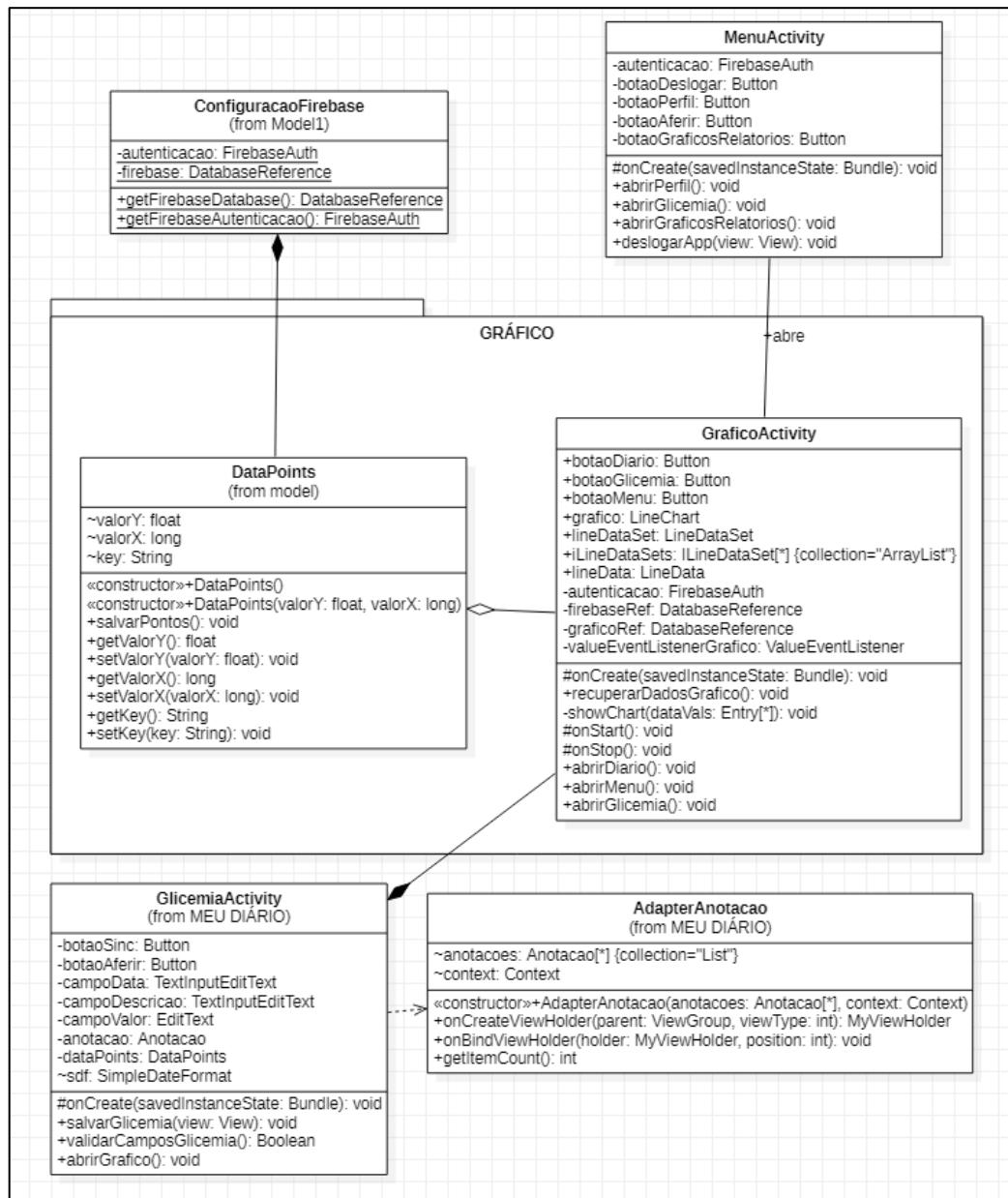
Fonte: Autor.

A classe Anotacao (Figura 24) possui os atributos e métodos necessários para salvar os dados que são mostrados dentro do diário, a classes RefeicaoActivity, MedicamentoActivity e GlicemiaActivity recebem os dados de refeição, medicamento e glicemia, respectivamente, e

salvam nos bancos de dados; também possuem métodos para validação dos campos e, no caso da glicemia, direcionar para a tela onde é mostrado o gráfico.

A classe AdapterAnotacao (Figura 24) possui o método que cria a lista de anotações dentro de cada mês (Figura 22-C). Ela representa uma funcionalidade importante dentro do aplicativo e foi criada separadamente por motivos de organização. A classe NotasActivity é a maior classe do software desenvolvido e possui todas as variáveis e métodos que são utilizados dentro do diário: método para excluir um registro; método para recuperar as anotações de um usuário assim que ele realizar *login* no aplicativo; método para abrir o menu flutuante e direcionar para as telas de Medicamento, Refeição ou Glicemia; entre outros métodos.

Figura 25: Diagrama de classes das classes que envolvem a funcionalidade “GRÁFICO” do aplicativo



Fonte: Autor

Além disso, a Figura 24 também mostra a classe de apoio ConFiguraçãoFirebase, usada para salvar os registros no banco de dados *Firebase Realtime Database*, e a classe DateCustom, utilizada para auxiliar na formatação de datas e identificação da data atual. A classe MenuActivity está incluída no diagrama pois ela possui o método que abre o Diário.

Já na Figura 25, podemos ver as duas principais classes utilizadas para plotar os dados glicêmicos no gráfico: DataPoints e GraficoActivity. A classe DataPoints recebe os dados glicêmicos e a data da medição da classe GlicemiaActivity e armazena no banco de dados. A classe GraficoActivity possui os atributos e métodos para gerar o gráfico de linha através dos dados fornecidos pela classe DataPoints, para recuperar os últimos dados que foram inseridos e também as funções que direcionam o usuário para uma das três opções disponíveis nos botões desta tela: “Ir para o menu”, “Ir para o diário” e “Realizar uma nova aferição de glicose” (Figura 22-A e B).

Além de enviar os dados para plotagem do gráfico, a classe GlicemiaActivity está no diagrama pois ela possui um método que direciona o usuário para a tela de gráfico logo após a inserção de um registro. A Figura 25 também mostra as classes AdapterAnotacao (responsável por listar os dados de glicemia no diário), MenuActivity (possui o método que abre o Gráfico) e a classe de apoio ConFiguraçãoFirebase (salva os dados na nuvem).

A 1^a iteração contou com outras classes de apoio que não foram ilustradas neste trabalho, mas foram essenciais para estruturar o código do aplicativo.

Após a entrega da 1^o iteração, os casos de uso da 2^o iteração começaram a ser desenvolvidos até resultarem em um novo protótipo.

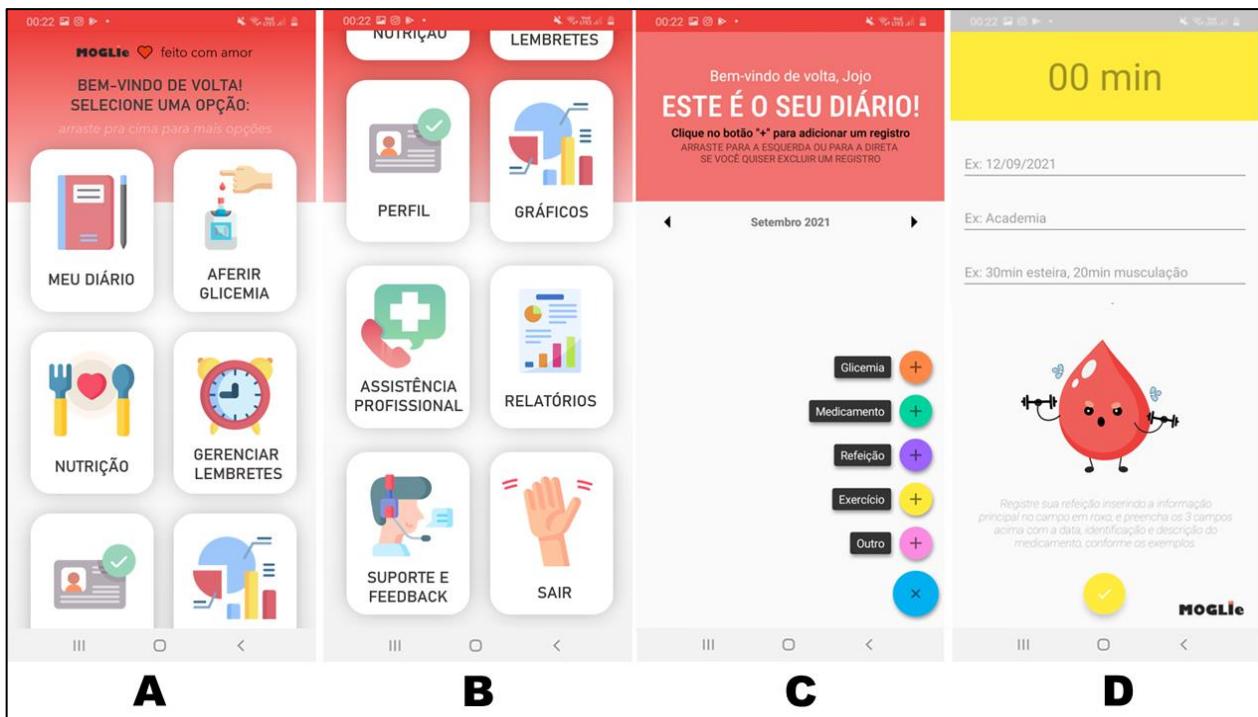
4.2 2^o ITERAÇÃO

A 2^o iteração do aplicativo MOGLIE envolveu a implementação dos casos de uso: registrar atividades físicas; disponibilizar nível de glicose dos alimentos para consulta (índice glicêmico e carga glicêmica) e registrar dados pessoais em ficha de saúde. Além disso, houveram mudanças em funcionalidades já existentes: o menu foi atualizado e novas telas foram acrescentadas na funcionalidade “Registrar Glicemia”.

A tela inicial, de cadastro e de login (Figura 19-A, B e C) não sofreram alterações visuais e nem de código. As telas de registro de glicemia, medicamento e refeição (Figura 21-B, C e D) também não foram alteradas. O atalho para a tela de registro de glicemia, “Aferir Glicemia” presente no primeiro protótipo (Figura 19-D) foi mantido no menu, e a opção “Deslogar” foi renomeada para “Sair”. O Gráfico (Figura 22-A e B) não sofreu alterações. O novo menu pode

ser visualizado na Figura 26-A e B, a nova tela de registro de exercícios (Figura 26-C) e o menu flutuante da tela “Meu Diário” (Figura 26-D) estão sendo mostradas.

Figura 26: Novo menu de opções (A), continuação do menu de opções (B), tela do “Meu Diário” com as novas opções “Exercício” e “Outros” (C) tela de registro de exercícios (D).



Fonte: Autor.

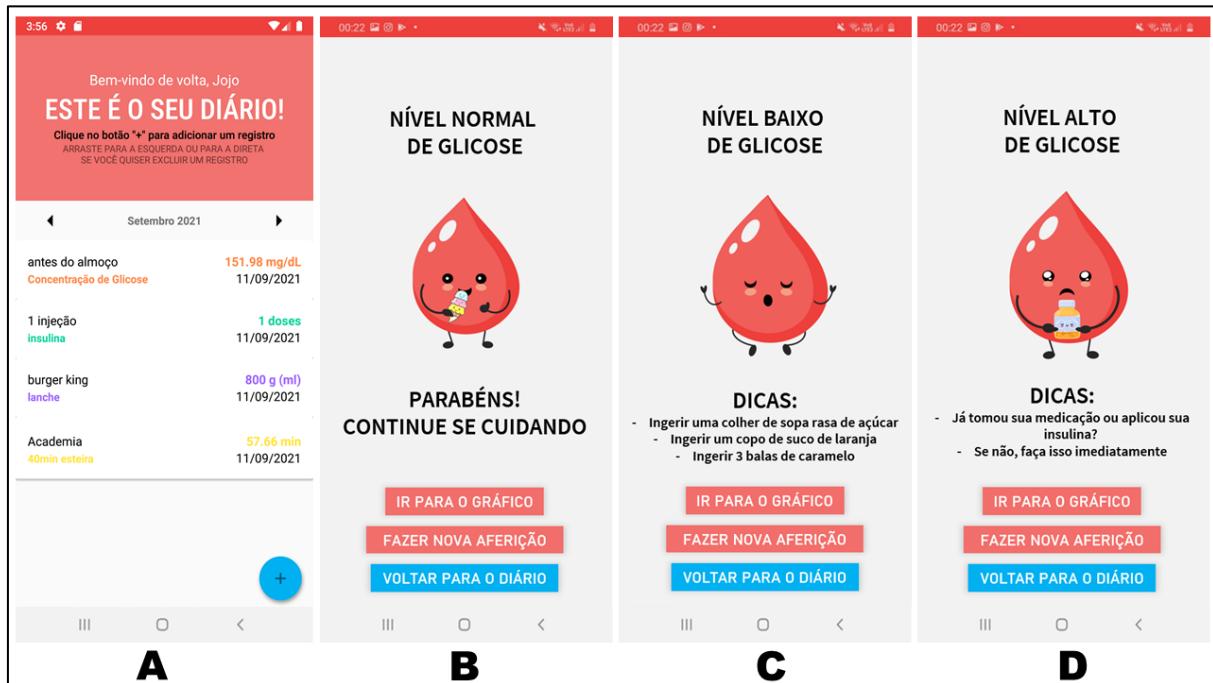
O novo menu contem mais opções, conta com botões grandes e ícones intuitivos. As opções “Assitência Profissional”, “Relatórios” e “Suporte e Feedback” não estão habilitadas, elas foram colocadas no menu a fim de preparar a arquiterura para as iterações seguintes (3 e 4) e proporcionar uma visão geral do produto.

Assim como na primeira iteração, ilustrado pelo diagrama de atividades da Figura 19, ao clicar em “Meu Diário” no menu (Figura 26-A), o usuário é direcionado para tela do diário (Figura 26-C), onde pode visualizar, excluir e adicionar um novo registro. Ao clicar no botão “+” ele pode escolher uma das 5 categorias (Glicemia, Medicamento, Refeição, Exercício ou Outro) para adicionar um novo registro.

A opção “Exercício” foi desenvolvida para implementar o caso de uso “Registrar atividades físicas” e direciona para a tela mostrada na Figura 26-D, onde o usuário pode inserir informações referentes a alguma atividade física realizada. A opção “Outro” não está habilitada, ela se refere ao caso de uso “Registrar outro tipo de informação/bloco de notas”.

A Figura 27-A, mostra um exemplo de cada categoria de anotação: Glicemia (laranja), Medicamento (verde), Refeição (roxo) e Exercício (amarelo). A telas representadas na Figura 27-B, C e D são uma melhoria no caso de uso “Registrar Glicemia”, elas representam os 3 possíveis feedbacks dados ao usuário após ele fazer um registro de glicemia.

Figura 27: Exemplo de anotações no diário (A), tela de feedback a um nível normal de glicose (B), tela de feedback a um nível baixo de glicose (C) e tela de feedback a um nível alto de glicose (D).



Fonte: Autor.

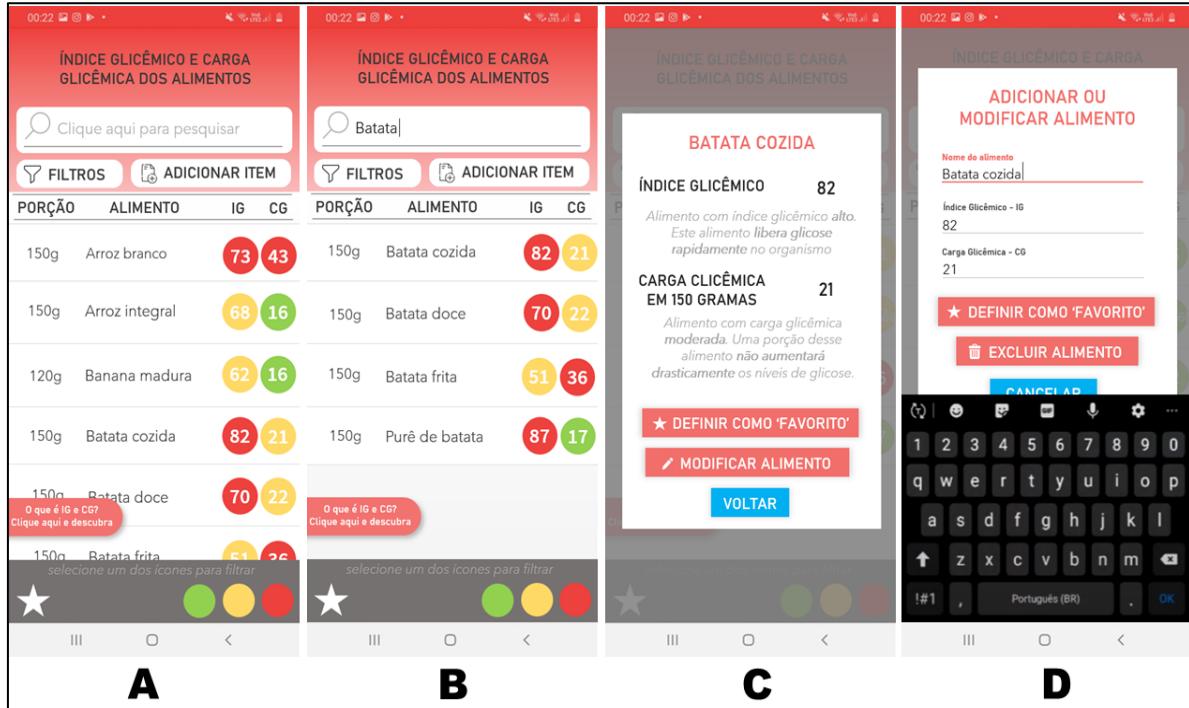
Se o valor da Glicemia inserido pelo usuário for menor ou igual a 70 ml/dl, então a Figura 27-C é exibida. Caso o valor da Glicemia seja maior ou igual a 180 ml/dl, a Figura 27-D é mostrada ao paciente. Caso o valor esteja no intervalo entre 70 ml/dl e 180 ml/dl, então a Figura 27-B é apresentada. O feedback vem acompanhado de dicas para regularizar o nível de glicose e também de 3 botões com as opções: “Ir para o gráfico”, “Fazer nova aferição” e “Voltar para o diário”. Os valores utilizados no intervalo são recomendados pela Sociedade Brasileira de Diabetes (2018), mas o nível de glicose considerado normal, alto e baixo pode variar para cada indivíduo.

O caso de uso “Disponibilizar nível de glicose dos alimentos para consulta (índice glicêmico e carga glicêmica)” foi desenvolvido e pode ser acessado através da opção “Nutrição” do menu. As telas desenvolvidas para esta funcionalidade podem ser visualizadas nas Figuras 28 e 29, a seguir.

Após clicar na opção “Nutrição” do menu, o usuário é redirecionado para tela apresentada na Figura 28-A, onde é possível visualizar os alimentos cadastrados em ordem alfabética. As

informações disponíveis na tabela são: porção, nome do alimento, índice glicêmico (IG) e carga glicêmica (CG).

Figura 28: tela da opção “Nutrição” (A), resultado da busca (B), detalhes do alimento (C) e alteração do alimento (D)



Fonte: Autor.

A Figura 28-B mostra o resultado da pesquisa pela palavra “Batata”, é possível perceber que o banco de dados retorna todos os alimentos que contem essa palavra ou uma parte dela, não sendo necessário escrever o nome completo do alimento para encontrá-lo. Além disso, os valores de IG e CG estão dentro de um círculo colorido, o círculo vermelho indica alto nível de IG e CG, o círculo amarelo indica um nível moderado de IG e CG e o círculo verde indica um nível baixo.

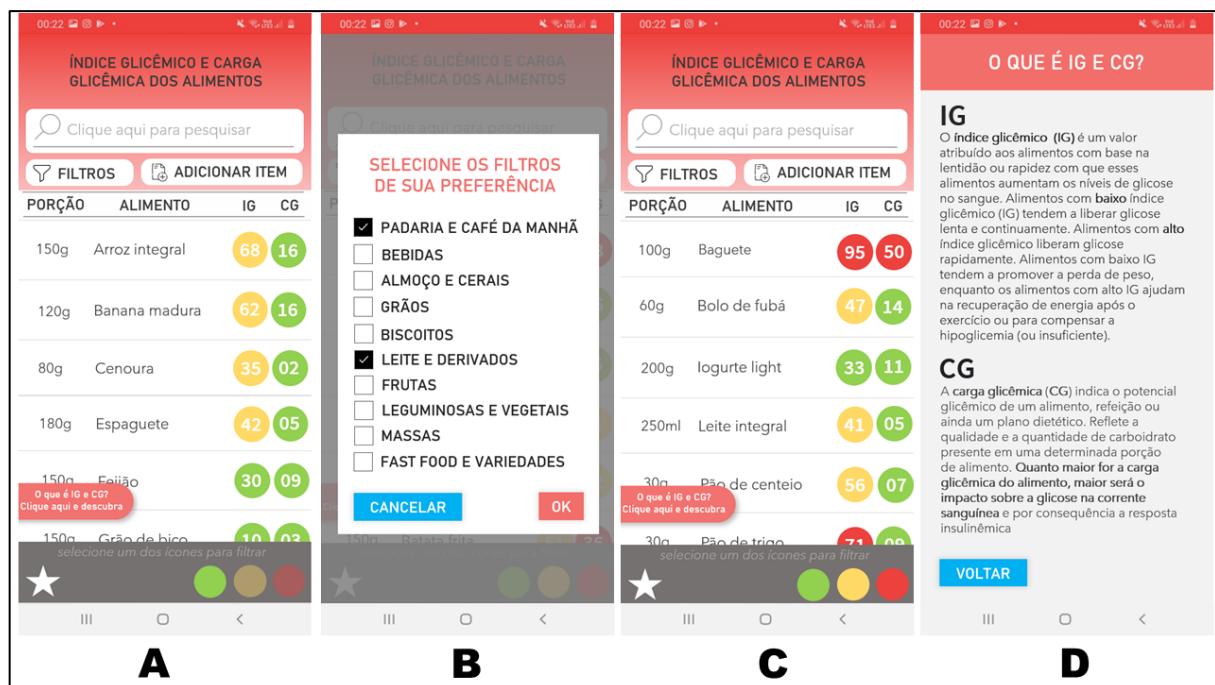
Ao clicar no alimento “Batata cozida”, o usuário é direcionado para a tela ilustrada na Figura 28-C, onde é possível ver os detalhes do alimento. Nesta tela, são apresentados os valores de IG e CG, acompanhado de dois parágrafos pequenos que informam o significado destes valores. No exemplo da Figura 28-C, o aplicativo está informando ao usuário que a batata cozida possui um Índice Glicêmico de 82, com a seguinte descrição: “Alimento com índice glicêmico alto. Este alimento libera glicose rapidamente no organismo.”, e uma Carga Glicêmica de 21 (em uma porção de 150 gramas), com o texto informativo: “Alimento com carga glicêmica moderada. Uma porção desse alimento não aumentará drasticamente os níveis de glicose.”. Dessa forma, os usuários mais leigos conseguem se manter informados em relação ao alimentos que serão ingeridos.

Ainda nesta tela (Figura 28-C), o usuário pode escolher uma das opções: “Definir como favorito” (que acrescenta o alimento a uma lista personalizada), “Modificar alimento” (que abre a tela mostrada na Figura 28-D) e “Voltar” (que redireciona o usuário para a tela anterior).

A tela apresentada na Figura 28-D permite que usuário modifique ou insira um alimento, o botão “Adicionar item” abaixo da barra de pesquisa (Figura 28-A e B) também direciona o usuário para essa tela. Além de alterar e/ou inserir novas informações, o usuário também pode excluir o registro de um alimento do banco de dados clicando no botão “Excluir alimento” (Figura 28-D).

Na extremidade inferior da tela, o usuário pode clicar em um dos ícones - círculos verde, amarelo e vermelho ou estrela branca - para filtrar os alimentos. Ao clicar no ícone verde, todos os alimentos com pelo menos um valor baixo de IG ou CG são filtrados, assim como demonstrado na Figura 29-A.

Figura 29: Alimentos filtrados pelo ícone verde (A), filtros disponibilizados ao clicar no botão “Filtro” (B), alimentos filtrados (C) e tela informativa sobre IG e CG (D).



Fonte: Autor.

Na Figura 29-A os alimentos estão filtrados pelo ícone verde, que indica um valor baixo de IG ou CG, eles também podem ser filtrados pelo ícone amarelo e vermelho caso o usuário deseja visualizar os alimentos com níveis moderados e altos de IG e CG, respectivamente. O ícone branco em formato de estrela, no canto inferior esquerdo, filtra os alimentos marcados como favoritos.

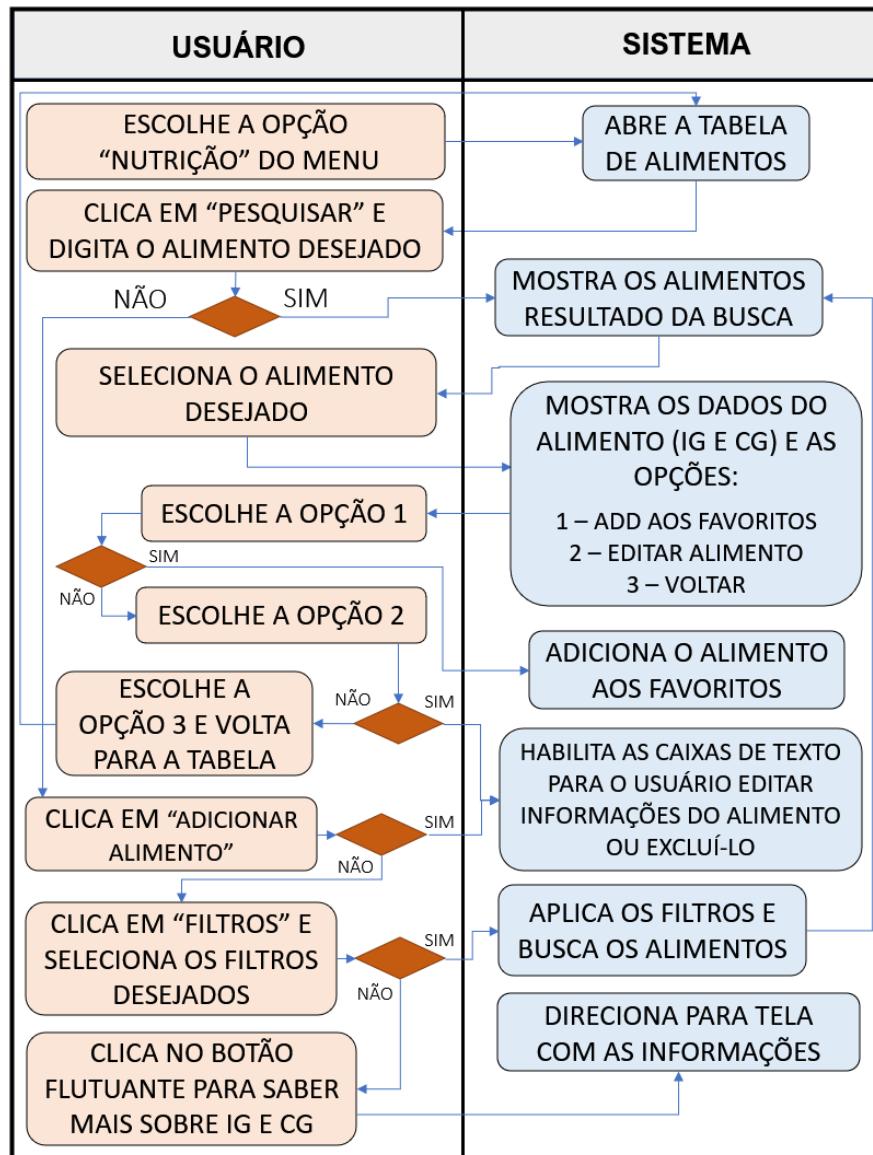
Ao clicar no botão “Filtros” abaixo da barra de pesquisa, o usuário é direcionado para a tela apresentada na Figura 29-B, onde ele pode selecionar os filtros que desejar. Os alimentos

foram categorizados de acordo com seu tipo, grupo nutricional e refeições onde geralmente são consumidos. No exemplo ilustrado, o usuário selecionou as opções “Padaria e café da manhã” e “Leite e derivados”, após confirmar a seleção clicando no botão “OK”, os filtros são aplicados. Os alimentos que fazem parte da categoria selecionada aparecem como resultado na tabela apresentada na Figura 29-C.

Por fim, o usuário também pode clicar no botão flutuante “O que é IG e CG? Clique aqui e descubra”, que aparece no canto inferior esquerdo da tela (Figura 29-A e C). Ao clicar nesse botão o usuário é direcionado para a Figura 29-D, uma tela com informações breves sobre os termos IG e CG, apresentando seu significado e importância.

O diagrama de atividades da Figura 30 resume o tráfego do usuário dentro da funcionalidade “Nutrição”, para facilitar o entendimento.

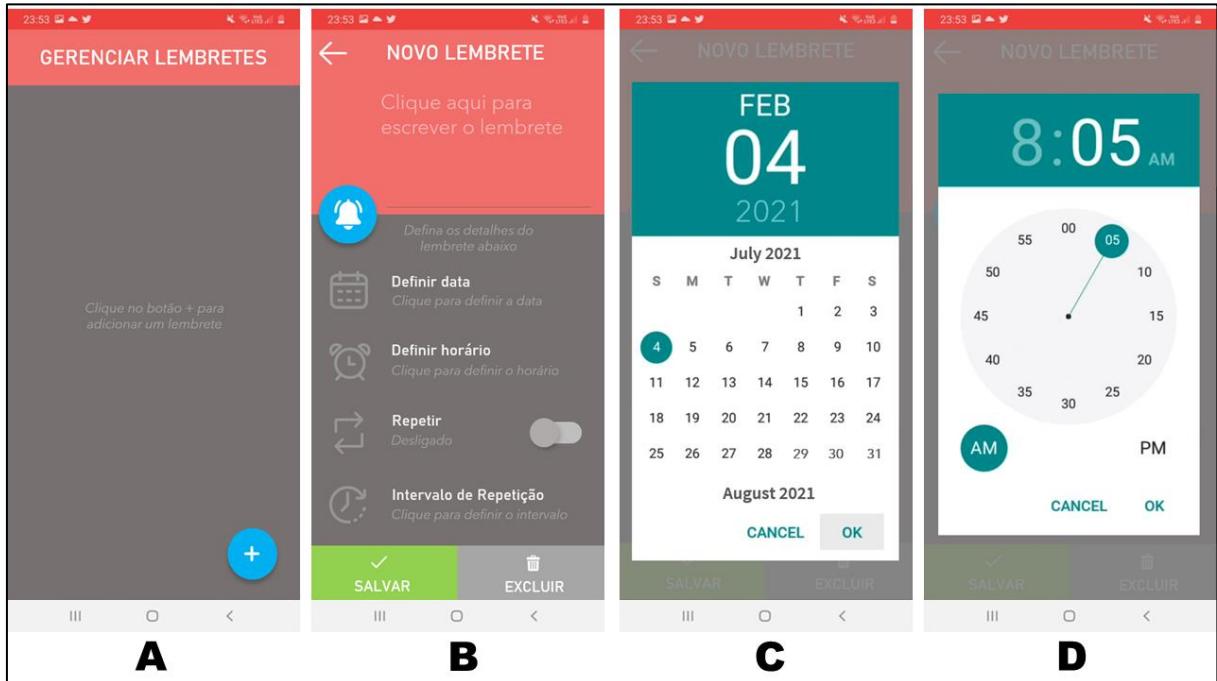
Figura 30: Diagrama de atividades para a opção “NUTRIÇÃO” do menu.



Fonte: Autor.

O caso de uso “Criar lembretes e notificações” pode ser acessado pelo usuário após ele clicar no botão “Gerenciar Lembretes” do menu, que o direciona para a tela apresentada na Figura 31-A abaixo.

Figura 31: tela “Gerenciar Lembretes” (A), tela “Novo lembrete” (B), tela “Definir data” (C) e tela Definir horário (D).



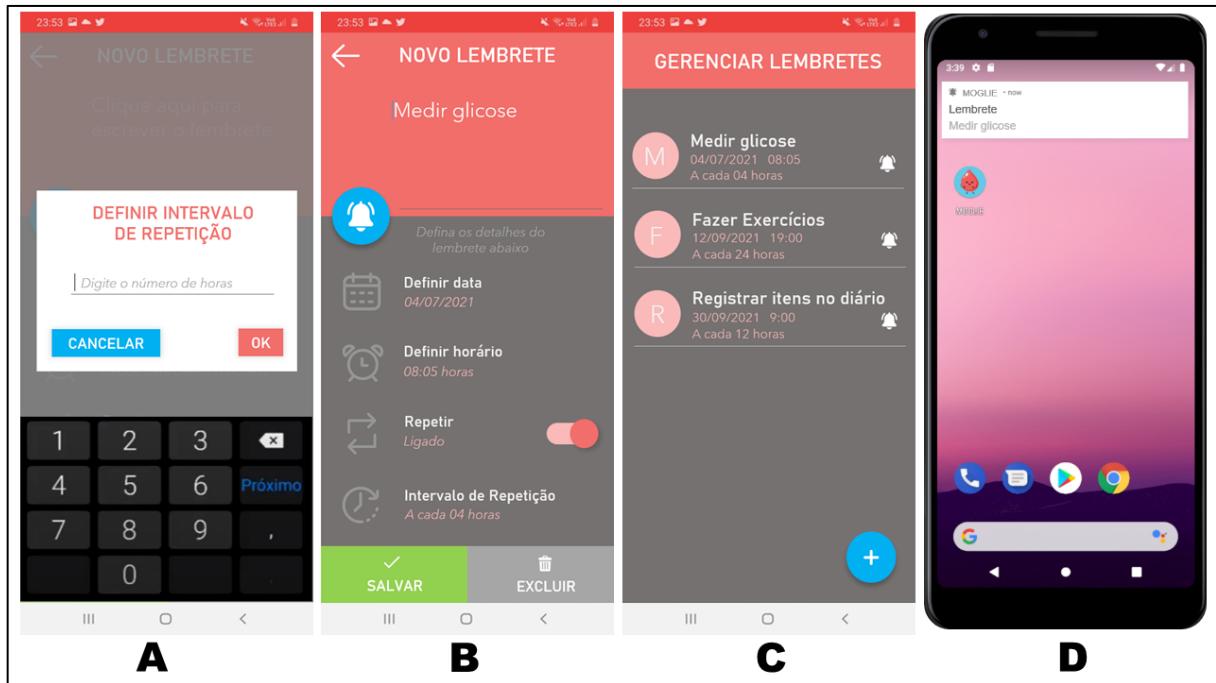
Fonte: Autor.

A Figura 31-A apresenta a lista de lembretes criados pelo usuário, que ainda não possui nenhum. Para acrescentar um novo lembrete, o usuário deve clicar no botão “+” no canto inferior direito da tela. Ao clicar nesse botão ele é direcionado para tela mostrada na Figura 31-B, onde deverá seguir todos os passos para a criação do lembrete.

O primeiro passo é escrever a descrição, título ou palavra-chave do lembrete, clicando em cima da caixa de texto sinalizada pela mensagem “Clique aqui para escrever o lembrete”. Em seguida o usuário deve definir a data do lembrete clicando na opção “Definir data”, que o direciona para a tela da Figura 31-C, onde ele pode selecionar o dia pelo calendário. Após definir a data, o usuário deve definir o horário do lembrete clicando na opção “Definir horário”, que o direciona para a Figura 31-D.

A opção “Repetir” permite que o usuário habilite ou desabilite a repetição do lembrete após um determinado número de horas, que é definido na opção “Intervalo de Repetição”. Para habilitar a repetição basta clicar no botão *switch* que o status muda de “Desabilitado” para “Habilitado”. Para definir o intervalo de repetição do alarme o usuário deve clicar nesta opção e inserir o número de horas que deseja estabelecer como intervalo, conforme mostra a Figura 32-A a seguir.

Figura 32: Definir intervalo de repetição do lembrete (A), visão geral do lembrete preenchido (B), lista de lembretes criados (C) e notificação do lembrete na tela do celular (D).



Fonte: Autor.

Após finalizar todas as etapas para criação do lembrete, é possível visualizar e conferir os dados, a Figura 32-B mostra o lembrete totalmente preenchido pelo usuário. Após clicar no botão “Salvar”, o lembrete é criado, e o usuário é direcionado para a tela anterior, onde é possível visualizar a lista dos lembretes criados. A Figura 32-C apresenta esta tela, com 3 lembretes criados pelo usuário. Para excluir um lembrete, o usuário seleciona o lembrete que deseja excluir, e depois o botão “Excluir”, no canto inferior direito da tela (Figura 32-B). A Figura 32-D mostra como a notificação do lembrete aparece no smartphone do usuário, é uma captura da tela do emulador do sistema operacional Android, dentro do Android Studio.

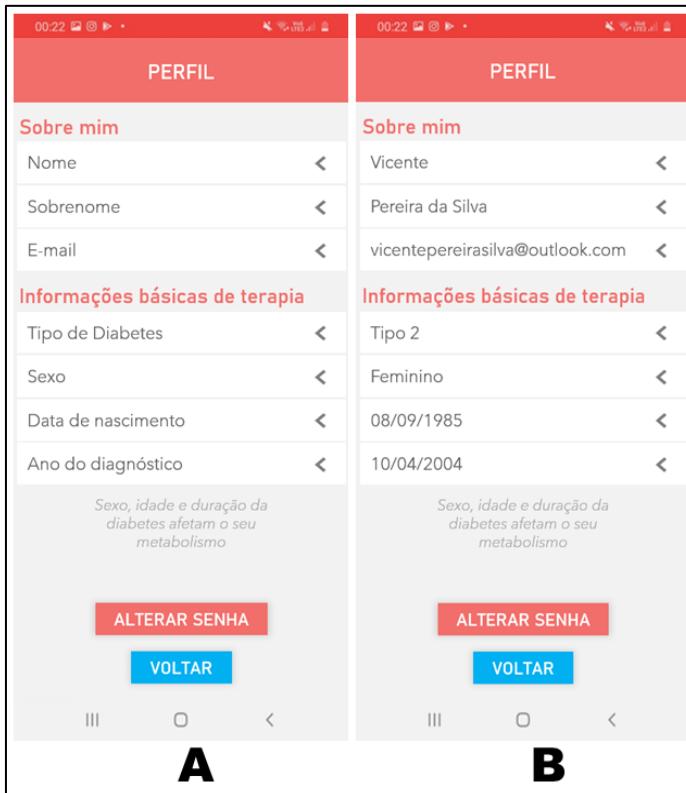
O último caso de uso implementado nesta iteração foi o “Registrar dados pessoais / ficha de saúde”. Optou-se por criar um formulário simples e de preenchimento opcional pelo usuário, ele pode ser acessado ao clicar na opção “Perfil” do menu e está ilustrado na Figura 33.

A Figura 33-A mostra o formulário desenvolvido para o perfil do usuário, que foi dividido em duas partes. A sessão “Sobre mim” permite que o usuário preencha com seu nome, sobrenome e e-mail, nas respectivas caixas de texto. A sessão “Informações básicas de terapia” contém os espaços para o usuário inserir os dados: tipo de diabetes, sexo, data de nascimento e ano do diagnóstico. A Figura 33-B mostra o formulário preenchido com dados fictícios.

Abaixo do último campo do formulário, existe um texto informativo: “Sexo, idade e duração da diabetes afetam o seu metabolismo”, com a finalidade de informar ao usuário o motivo pelo qual o formulário foi criado e incentivar o preenchimento desses dados. Na parte

inferior da tela, existe o botão “Alterar senha”, que não está habilitado. Ele foi planejado para permitir que o usuário altere sua senha de acesso, mas essa funcionalidade não foi implementada nessa iteração.

Figura 33: Formulário do perfil (A) e formulário do perfil preenchido (B)



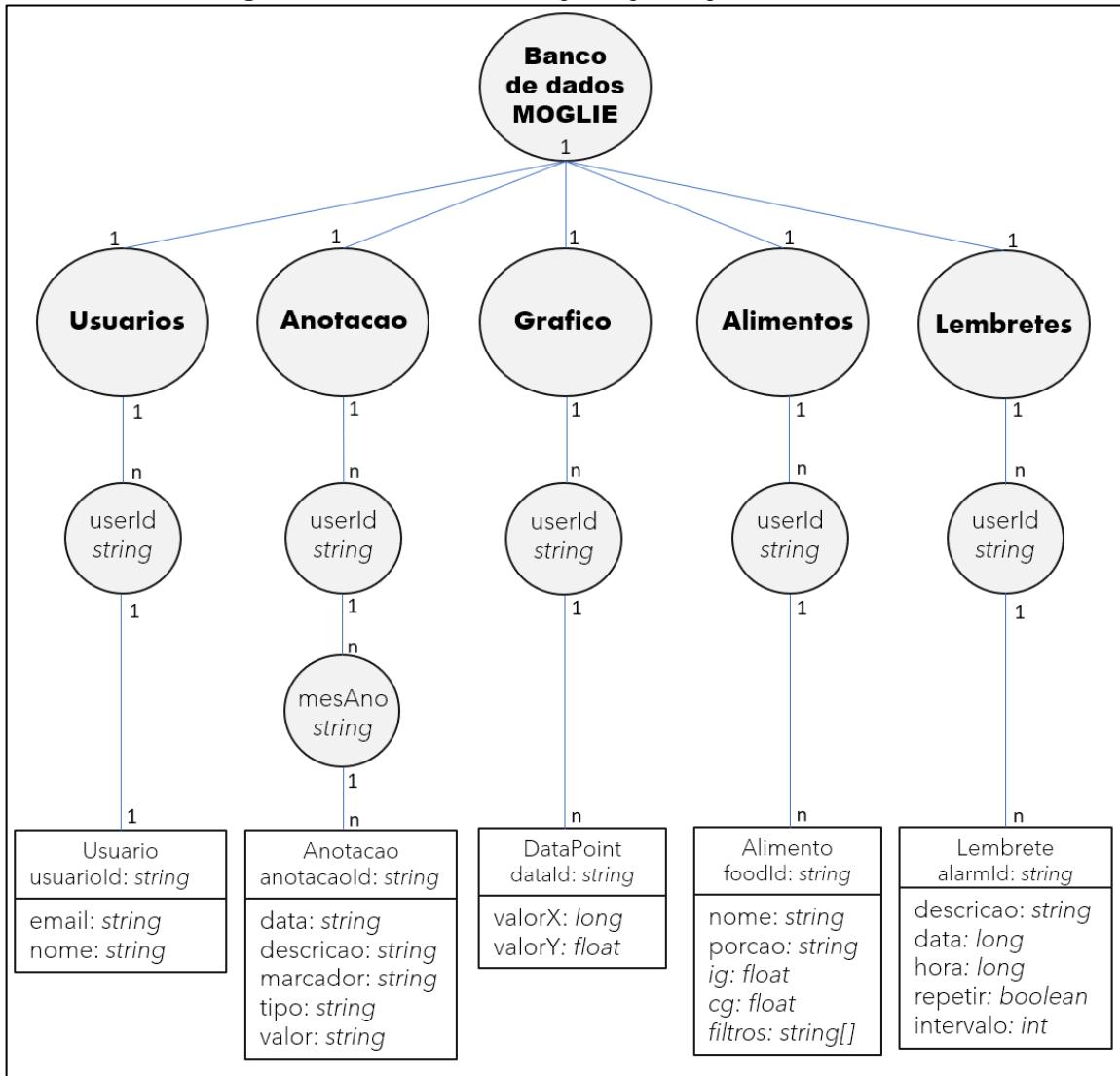
Fonte: Autor.

Após implementar as funcionalidades ilustradas pela Figura 33, finalizou-se o desenvolvimento da 2º iteração. Até a data de publicação deste trabalho, apenas as iterações 1 e 2 foram desenvolvidas.

4.3 PERSISTÊNCIA DE DADOS

Conforme definido na metodologia, foram utilizados dois bancos de dados no aplicativo, SQLite para armazenar os dados no dispositivo e o *Firebase Realtime Database* para armazenar um backup dos dados na nuvem. O banco de dados *Firebase* é um banco de dados NoSql, isso significa que os seus dados não se encaixam nos modelos de Entidade-Relacionamento. O *Firebase* representa os seus dados em forma de árvore, isto acontece por que eles são armazenados em objetos JSON, que é uma estrutura de dados hierárquica, muito usada na internet (FIREBASE, 2018). Portanto, os dados foram representados em uma estrutura de dados em árvore, apresentado na Figura 34.

Figura 34: Modelo de dados do protótipo do aplicativo MOGLIE.

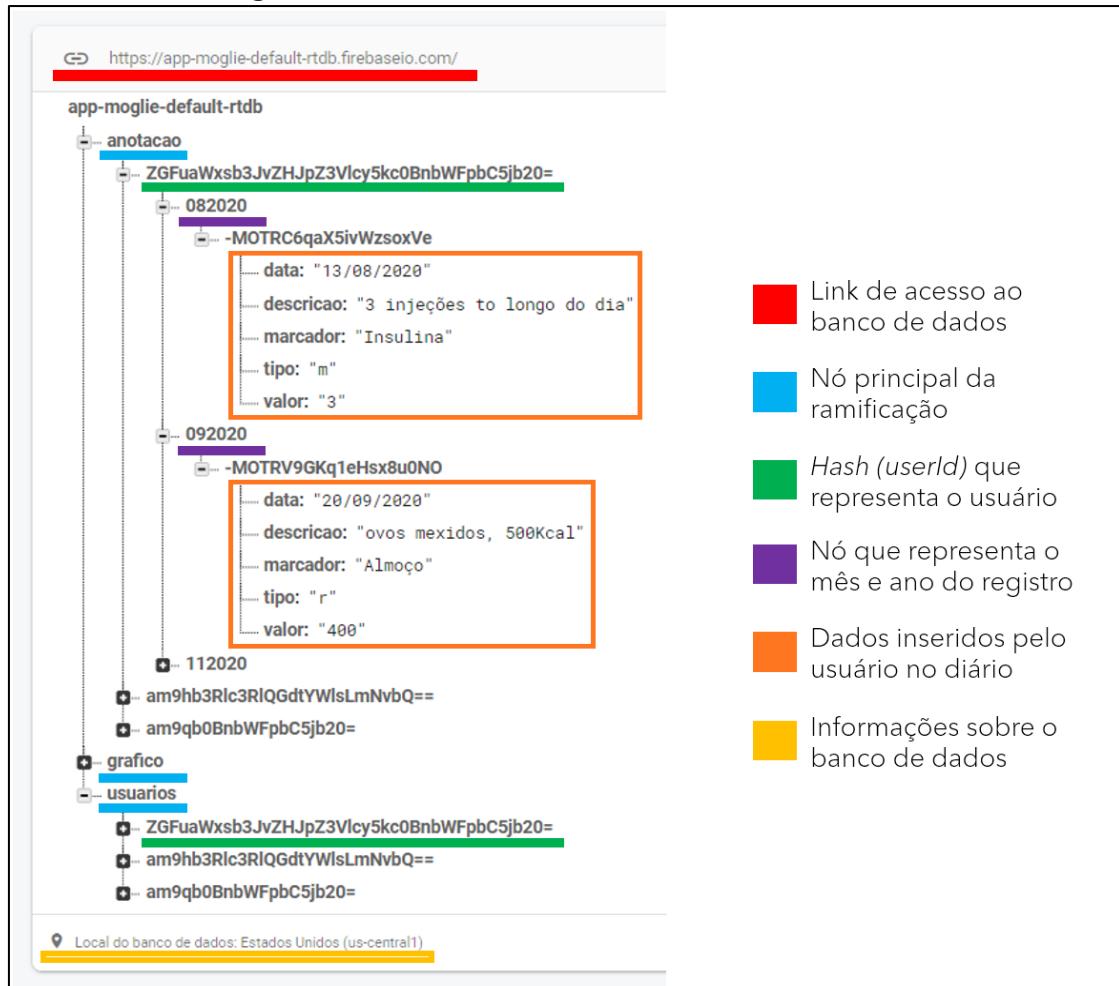


Fonte: Autor.

O modelo de dados se divide em 5 ramificações principais, cada uma delas é chamada de Nô e contém os dados de uma determinada funcionalidade do aplicativo: Usuarios, Anotacao, Grafico, Alimentos e Lembretes. A ramificação Anotacao armazena todos os dados do “Meu Diário”, que são os registros de glicemia, refeições, medicamentos e exercícios. Assim como é mostrado na Figura 28, um mesmo usuário pode armazenar “n” registros de Anotacao, ou seja, pode adicionar quantas anotações desejar ao seu diário. O nó Usuarios, por outro lado, permite que o usuário possua somente 1 registro contendo e-mail e senha, pois essas informações são únicas para cada cadastro.

A Figura 35 exemplifica como as informações ficam salvas dentro do *Firebase*, mostrando os dados armazenado dentro do nó Anotacao. Cada elemento está representado por uma cor, com sua respectiva descrição na legenda, ao lado direito da imagem.

Figura 35: Estrutura do armazenamento dos dados na nuvem.



Fonte: Autor.

Além disso, através do site do *Firebase* é possível acessar o serviço *Authentication*, que permite o administrador do sistema visualizar e gerenciar todos os usuários cadastrados no aplicativo, como mostra a Figura 36.

Figura 36: Usuários cadastrados no aplicativo MOGLIE

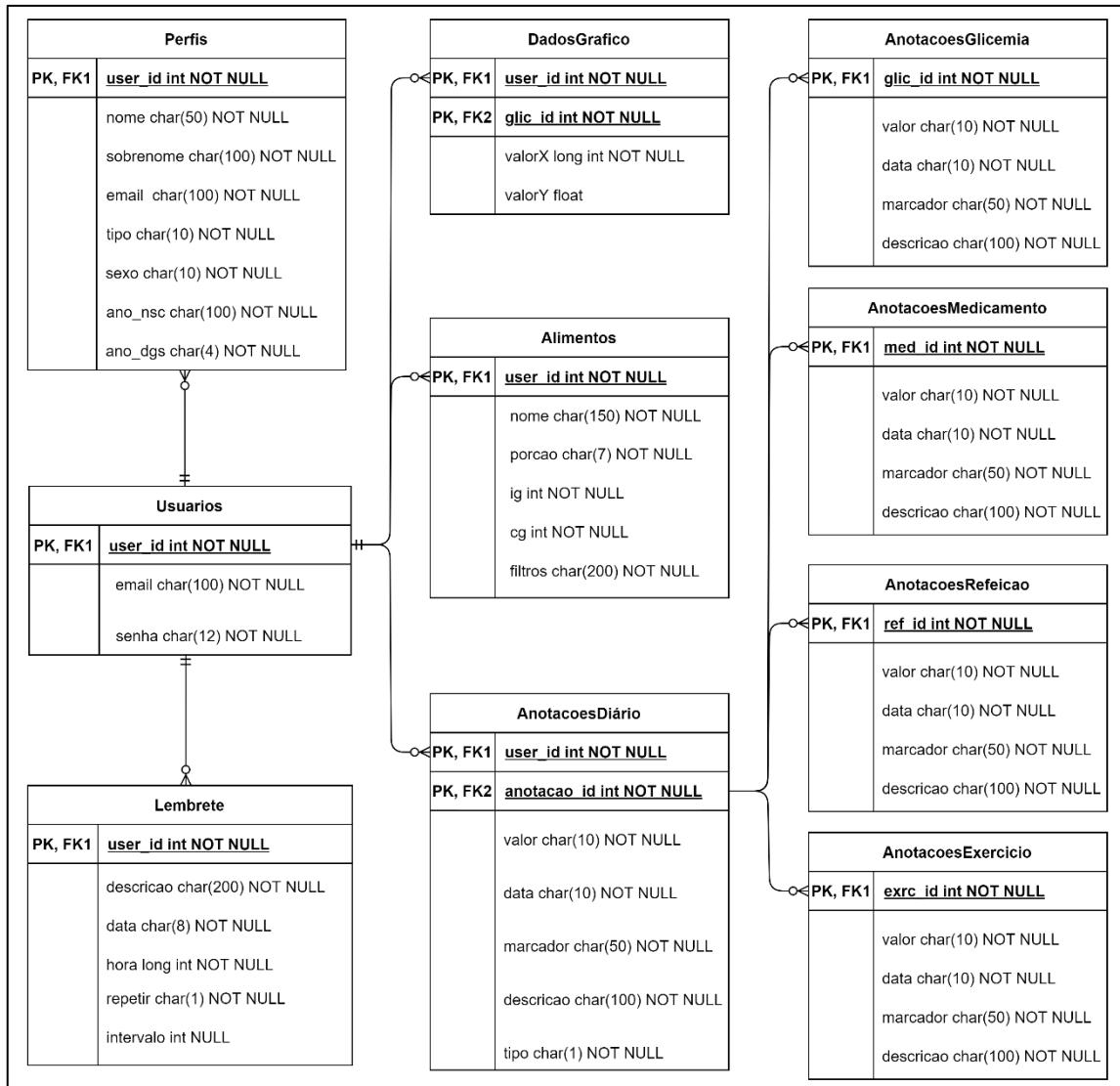
Pesquise por endereço de e-mail, número de telefone ou UID do usuário		Adicionar usuário		⋮	
Identificador	Provedores	Data de criação	Último login	UID do usuário	⋮
brunacampinos.g1998@gmail.com	✉	15 de dez. d...	15 de dez. d...	56d5Tk5Z2desnyJfT8ie67kuzz72	
joaoteste@gmail.com	✉	14 de dez. d...	14 de dez. d...	VifFXGzD4ld400msnTbhFH3z6Y93	
jojo@gmail.com	✉	13 de dez. d...	8 de jan. de ...	Jrungf28NRQo6Vr7HJ110mSeMR...	
danillorodrigues.ds@gmail.com	✉	13 de dez. d...	13 de dez. d...	fEh4StGZ1fsx1y03N7iTWvArFID2	
maisa123@gmail.com	✉	13 de dez. d...	13 de dez. d...	s4zwI8NJQdY7RE3MO8RC09mFL...	

Fonte: Autor.

Através dessa ferramenta, é possível visualizar a data que o usuário foi cadastrado (Data de criação), A data do último login realizado e o *userID* do usuário (utilizado dentro do *Realtime Database* – Figuras 34 e 35). Clicando sobre um usuário específico, é possível redefinir sua senha, desativar ou excluir sua conta.

A estrutura do banco de dados SQLite é similar ao *Firebase* e pode ser vista na Figura 37. Nela podem-se notar as tabelas definidas para o armazenamento dos dados pessoais do usuário, lembretes, alimentos, anotações no diário, dados do gráfico, entre outras tabelas.

Figura 37: Diagrama Entidade-Relacionamento do Banco de dados do aplicativo.



Fonte: Autor.

O diagrama ilustra todas as 10 tabelas desenvolvidas durante as iterações 1 e 2. Para implementar os casos de uso das iterações 3 e 4, será necessário adicionar mais tabelas ao banco. A tabela Alimentos é a única que necessita ser previamente preenchida pelo desenvolvedor,

para possibilitar a consulta pelas informações de IG e CG dos alimentos, conforme ilustrado no item 4.2.

Tabelas confiáveis de índices glicêmicos (IGs) e cargas glicêmicas (GLs) são essenciais para pesquisas que examinam a relação entre qualidades glicêmicas de carboidratos em alimentos, dietas e saúde. Os índices e cargas glicêmicas cadastrados para consulta no aplicativo foram retirados de um trabalho publicado no *The American Journal of Clinical Nutrition*, que contém informações sobre mais de quatro mil alimentos (Atkinson *et al*, 2021). Para testar a funcionalidade e desenvolver o protótipo, foram escolhidos 20 alimentos para serem cadastrados. As demais tabelas ilustradas pelo diagrama da Figura 37, são preenchidas pelo usuário a medida que ele vai utilizando o aplicativo.

Ao final das iterações 1 e 2, o desempenho do aplicativo foi analisado, para certificar que o software funciona conforme planejado.

4.4 DESEMPENHO DO APLICATIVO

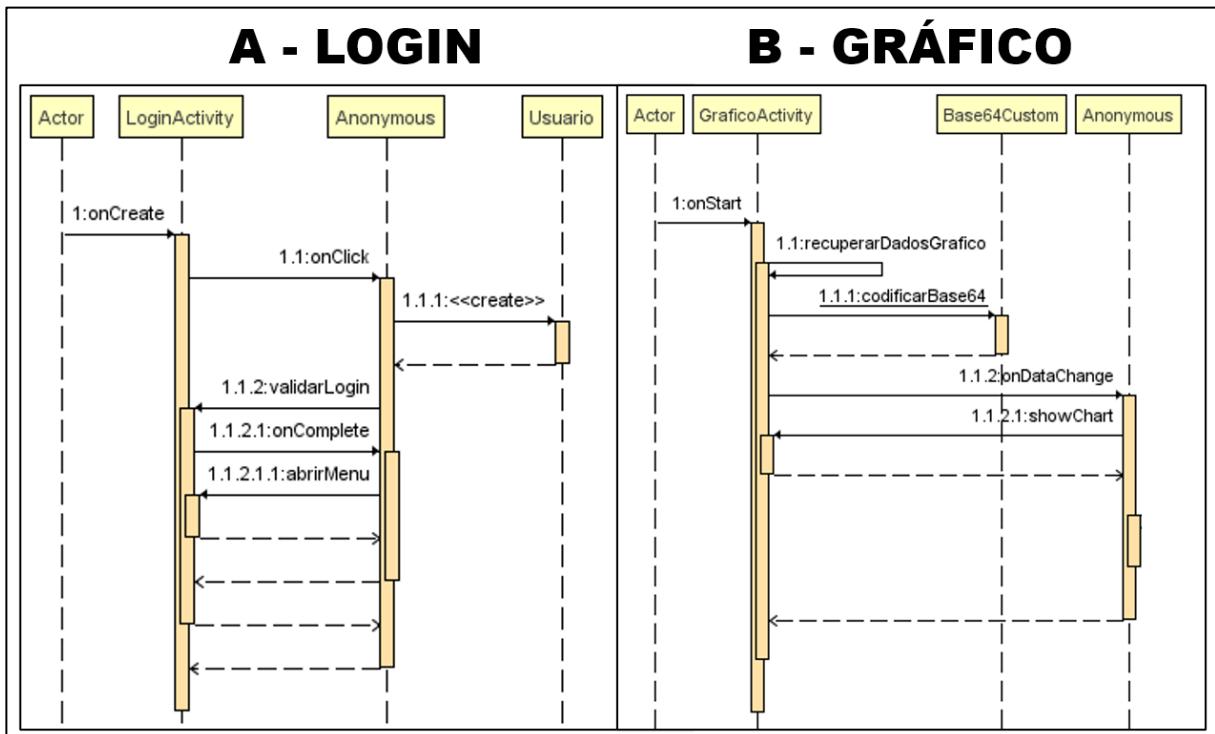
Para testar a aplicação, utilizou-se simuladores que fazem parte do Android Studio, ele simula o sistema operacional Android e permite realizar a instalação do aplicativo e utilizá-lo assim como em um smartphone.

A qualidade do aplicativo foi um ponto gerenciado desde o início do projeto. O código foi analisado usando a ferramenta *Analyze* do próprio Android Studio, que identifica várias possibilidades de falhas e indica boas práticas de programação. O uso da *Analyze* possibilita não deixar os testes de QA (*Quality Assurance* – Garantia de qualidade) somente para o final, visto que projetos ágeis são construídos de maneira iterativa e incremental. Os testes e a garantia da qualidade como um todo ocorreram em paralelo com o desenvolvimento do produto, sendo que desde a definição dos requisitos, os processos foram analisados e desenvolvidos com foco na qualidade do protótipo que foi entregue ao fim de cada iteração. Para monitoramento do banco de dados *Firebase*, utilizou-se o serviço de notificação de erros *Crash Reporting*, que não reportou nenhum erro ou problema de infraestrutura.

Para entender o comportamento do aplicativo em tempo de execução, não bastam apenas diagramas de código estático, mas sim de modelos que representem a comunicação dinâmica entre os objetos durante a sua execução, para isso utilizou-se os diagramas de sequência, gerados dentro da IDE Android Studio, que demonstram a execução dos objetos em relação ao tempo. A Figura 38 mostra o processo de funcionamento do Login (A) e do Gráfico (B). Os diagramas de sequência ilustram o comportamento dos objetos durante a execução. Na Figura

38-A, é possível ver que ao clicar no botão “Entrar” na tela de login (representada no diagrama de sequência pela ação “onClick”) os dados são autenticados pelo método “validarLogin”, e, após serem validados (“onComplete”), o software direciona o usuário ao menu pelo método “abrirMenu”, assim como planejado.

Figura 38: Diagramas de sequência das funcionalidades Login (A) e Gráfico (B).



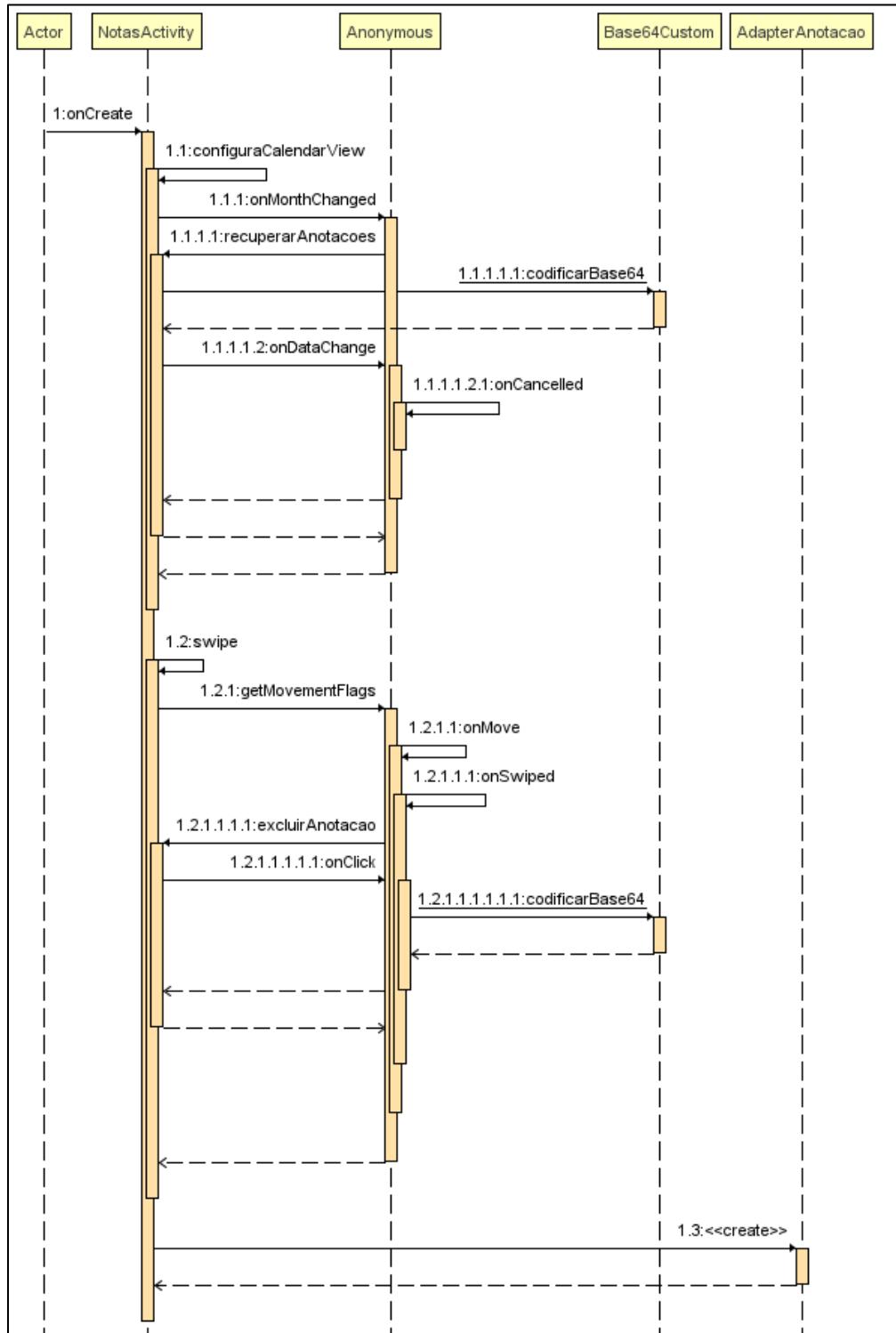
Fonte: Autor.

Na Figura 38-B, podemos acompanhar todos os métodos que são executados ao abrir a tela Gráfico, o método “recuperarDadosGrafico” recupera os dados do gráfico do usuário que está utilizando o aplicativo. Para encontrar esse usuário no banco de dados, o sistema utiliza o método “codificarBase64” para descriptografar o nome do usuário e localizar os dados a serem plotados. A ação “showChart” executa as funções necessárias para mostrar o gráfico ao usuário.

A Figura 39 mostra o diagrama de sequência da funcionalidade “Meu diário”, representado pela classe NotasActivity, a maior classe do aplicativo. Todas as ações, métodos e respostas do sistema estão dispostos no diagrama. Ele ilustra as principais ações que são reliazadas dentro do diário e a resposta do sistema à elas. Ao navegar entre os meses do diário (representado diagrama pela ação “onMonthChanged”) as anotações do usuário para um determinado mês são recuperadas pelo método “recuperarAnotacoes”, que utiliza o método “codificarBase64” para descriptografar e localizar os dados, assim como na Figura 38-B. A ação “onSwiped” representa o movimento de arraste realizado pelo usuário quando ele deseja excluir uma anotação do diário, que aciona método “excluirAnotacao”. Este método mostra

uma janela de confirmação ao usuário e, caso ele clique para confirmar a exclusão (ação “onClick” no diagrama), o dado é excluído do diário e também do banco de dados.

Figura 39: Diagrama de sequência da funcionalidade “Meu Diário”.



Fonte: Autor.

O processo de análise dos diagramas de sequência foi realizado para todo código do aplicativo, para garantir seu funcionamento e diagnosticar os problemas.

O monitoramento das transações realizadas dentro do banco de dados *Firebase Realtime Database* foi realizado dentro da plataforma, através do gráfico da Figura 40, que mostra o total de permissões, negações e erros dos últimos 30 dias (20 de agosto a 19 de setembro de 2021).

Figura 40: Gráfico de monitoramento de permissões, negações e erros do *Firebase Realtime Database*.



Fonte: Autor.

Na Figura 40 é possível visualizar que não houveram erros ou negações no período analisado. O número total de solicitações avaliadas como verdadeiras nas regras de segurança do banco de dados foi de 55 avaliações permitidas.

A fase de Transição do RUP não foi finalizada, pois até a data de publicação deste trabalho apenas as iterações 1 e 2 foram desenvolvidas. Sendo assim, o aplicativo não foi publicado na Google Play Store. Entretanto, como esta aplicação foi projetada com o objetivo de ser um protótipo executável, ela foi exportada e instalada em um celular, onde encontra-se em teste interno.

5 DISCUSSÃO

O autocuidado é a melhor forma de se ter uma vida saudável e com qualidade. Existem diversas tecnologias que facilitam o alto cuidado de pessoas com diabetes, e dentre elas está o uso de aplicativos móveis para o gerenciamento da glicemias.

A metodologia RUP foi escolhida para ser utilizada no desenvolvimento do protótipo, através dela foi possível planejar os incrementos e iterações que foram implementados. Devido a quantidade e complexidade dos casos de uso em relação ao tempo e quantidade de desenvolvedores no projeto, o projeto foi segmentado em quatro iterações com o objetivo de poder criar, ao final de cada iteração, um protótipo. Este trabalho focou em mostrar o desenvolvimento da 1º e da 2º Iteração, pois elas demonstram o fluxo completo do aplicativo e prepara a arquitetura para os demais casos de uso.

No estudo de aplicações semelhantes realizado na revisão de literatura, foram elencados os principais aplicativos para autogerenciamento da diabetes, disponíveis no mercado e suas principais funcionalidades foram comparadas, conforme o Quadro 2. Dessa lista, as funcionalidades implementadas no protótipo foram:

- cadastro de dados pessoais;
- atribuir senha para acesso ao app;
- registro e gerenciamento do nível de glicemia;
- registro e gerenciamento de medicamentos orais/insulina;
- registro e gerenciamento de informações alimentares;
- registro e gerenciamento de exercícios físicos; consultar informações nutricionais, dieta para diabéticos ou nível de glicose dos alimentos (funcionalidade parcialmente implementada);
- sistemas de lembretes, alarmes ou notificações;
- gerar gráficos dos dados glicêmicos.

Do total de funcionalidades elencadas no quadro 2, a única funcionalidade não implementada no protótipo foi a ‘exportar relatórios’, prevista para ser desenvolvida na 4º iteração, conforme o Quadro 7.

Conforme mencionado, a funcionalidade “consultar informações nutricionais, dieta para diabéticos ou nível de glicose dos alimentos” foi parcialmente implementada. Devido sua complexidade, durante o levantamento de requisitos, análise e projeto de arquitetura ela foi dividida em 3 casos de uso: consultar nível de glicose dos alimentos (índice glicêmico e carga

glicêmica); consultar informações nutricionais e dietas; realizar planejamento alimentar. Esses dois últimos foram planejados para serem implementados na 3º iteração, conforme Quadro 7, portanto não foram desenvolvidos até a data de publicação deste trabalho.

Os principais objetivos do aplicativo MOGLIE são promover o automonitoramento e a mudança de comportamento através de um design centrado no usuário. O sistema operacional Android permite usar um conjunto de pacotes, contendo classes pré-definidas, que possibilitam o acesso aos recursos do sistema operacional e a incorporação dos mesmos nos aplicativos em desenvolvimento (DEITEL; DEITEL; DEITEL, 2015). Estes pacotes auxiliaram no tratamento e uso das convenções de aparência, comportamento e diretrizes de estilo específicas de aplicativos do Android e permitiram desenvolver um layout de acordo com os requisitos não-funcionais estabelecidos para a interface.

Diversos modelos de interfaces foram estudados durante o período de desenvolvimento do aplicativo MOGLIE. Alguns critérios foram priorizados para se definir o layout final, como a facilidade para acessar cada funcionalidade da aplicação, a rapidez na navegação entre telas e uma identidade visual agradável para melhor apreciação do conteúdo. Todos estes critérios são boas práticas de usabilidade para melhorar a experiência de uso e retenção de usuários em aplicativos móveis (GRIFFITHS, 2015).

Por mais interessados que os pacientes possam estar com a utilização da tecnologia na autogestão de seus próprios cuidados, inicialmente pode ocorrer frustração com as características de design inerentes a usabilidade e falta de confiança com a tecnologia (SARKAR et al., 2016). Os aplicativos precisam ser simples, com botões grandes e com instruções de navegações fáceis de seguir, o que aumenta a possibilidade de engajar uma ampla faixa etária e tornar relevantes para aqueles com menor nível de alfabetização e com idade avançada (SARKAR et al., 2016; CECHETTI et al., 2019). No tocante a esse aspecto, o aplicativo proposto nesse trabalho priorizou o uso de botões grandes e elementos visuais intuitivos, acompanhados por dicas e instruções de uso.

Dentre os recursos encontrados nos aplicativos que foram estudados durante a revisão de literatura, o alarme sonoro, o registro de medidas glicêmicas e de ingestão de medicamentos estiveram mais presentes (FROISLAND D, ARSAND E, SKARDERUD, F., 2012; ÅRSAND et al, 2015). Estes três recursos foram implementados no protótipo desenvolvido, Lyons (2013) afirma que o monitoramento constante dos níveis glicêmicos tem sido essencial para o usuário alcançar o controle glicêmico e reduzir os riscos de complicações (TIEFENGRABNER, 2014).

O recurso de gerenciar lembretes foi implementado ao protótipo para que o usuário crie alarmes personalizados, para lembrá-lo de realizar alguma tarefa. O recurso é integrado aos

sistemas de alarmes do sistema operacional Android e emite um alarme sonoro e notificação ao ser ativado. De acordo com o estudo realizado por Mulvaney et al. (2012) e com o trabalho publicado por Pagoto et al. (2013), uma das vantagens dos recursos de alarmes sonoros nos aplicativos, em comparação com um relógio com alarme e anotações em registros de papel, é a praticidade de o usuário poder ajustar os horários de aplicação da insulina de acordo com as refeições, e lembrar facilmente quando ele mediu a glicemia atual e anterior. Isso pode ser útil após um evento de hipoglicemia, permitindo ao usuário ver e entender o efeito do carboidrato na refeição sobre os índices glicêmicos.

Aliado a ferramenta de criar lembretes, o registro de medicamentos também pode ser uma importante ferramenta no combate a falta de adesão a medicamentos a longo prazo. No ensaio clínico randomizado conduzido por Ashoorkhani et al. (2016), realizado com 132 pacientes hipertensos primários com uso de um aplicativo pra monitoramento de pressão arterial, identificou-se que a adesão aos medicamentos a longo prazo depende do comportamento e do autogerenciamento.

Outra importante ferramenta do aplicativo é a visualização dos dados glicêmicos ao longo do tempo através de um gráfico. Vários estudos mostram os efeitos, eficiência e benefícios de gráficos e visualização de dados para Diabetes Mellitus (KIM, 2006; WONG et al, 2017; CHEN et al, 2018). A visualização de dados do MOGLIE visa facilitar a identificação de padrões, falhas de tratamento, e apoia a compreensão do efeito da insulina e das atividades do dia-a-dia de forma ativa e ao longo do tempo.

Apesar da visualização de dados desenvolvida no protótipo ser uma ferramenta útil ao usuário, ela apresenta limitações. Um gráfico temporal integrando todos os tipos de registros, variação da glicemia, refeições, medicamentos e exercícios, permitiria a inspeção e interpretação lado a lado da ligação, interação, causalidade e interdependência entre os fatores de tratamento da diabetes e suas circunstâncias, e promoveria a compreensão do impacto que os alimentos têm sobre a glicemia e como os alimentos interagem com o exercício e a insulina para prevenir a hipoglicemia e a hiperglicemia. Além disso, não é possível selecionar o período para visualização, o protótipo do aplicativo informa apenas as medições mais recentes de glicemia. Essas melhorias podem ser incluídas nas próximas iterações do projeto, afim de atingir o maior potencial da ferramenta e proporcionar uma melhor experiência ao usuário.

As abordagens de visualização têm um valor significativo para os profissionais de saúde, uma vez que lhes permite encontrar informações ocultas nos dados, obter insights e evidências e tomar decisões mais assertivas (KO, CHANG, 2018). Ferramentas gráficas podem permitir que o médico visualize rapidamente o tempo, nome e dosagens da medicação de um paciente

(GORDON, BHAN, 2017). Nessa perspectiva, uma versão melhorada do gráfico, aliada a implementação dos casos de uso previstos para a 4^a iteração (exportar relatórios e contactar assistência profissional) podem ser um instrumento útil para os profissionais de saúde monitorarem o tratamento do paciente, visualizando dados e exportando relatórios.

Estudo aponta que a escassez de tempo dos profissionais da saúde para educar os usuários pode ser compensada através da implementação de autocuidado via uso de aplicativo, que pode fornecer apoio de decisão para o autocuidado e otimizar as condutas de tratamento para cada indivíduo (KUMAH-CRYSTAL et al., 2015). Em vista desse contexto, o aplicativo desenvolvido pode auxiliar seus usuários a entenderem os objetivos do tratamento, desenvolver hábitos e a estabelecerem metas para o autocuidado, levando em consideração a motivação para as práticas de autocuidado e o tempo de diagnóstico.

Uma pesquisa com pacientes diabéticos determinou que os recursos do aplicativo mais importantes para eles incluíam, além da facilidade de uso, a comunicação direta com o glicosímetro (SKORVSETH et al, 2015). A arquitetura do aplicativo MOGLIE foi planejada para que, no futuro, essa melhoria possa ser implementada, a tela de registro de glicose (Figura 21-C) possui um botão chamado “conectar e receber dados do glicosímetro”, ele foi implementado para simular a conexão com um dispositivo medidor de glicose e promover uma visão geral de como seria essa funcionalidade. Ao clicar nesse botão, um valor aleatório de glicose aparece na tela, tal qual um glicosímetro quando está sincronizado com o dispositivo móvel e envia os dados para a aplicação. Caso o protótipo fosse lançado como um aplicativo finalizado e esse recurso não fosse implementado, o botão seria excluído. Dessa forma, o usuário só poderia inserir os valores glicose manualmente, não havendo a opção de conectar com outros equipamentos.

O auto monitoramento da ingestão alimentar pode ser um aliado para a manutenção efetiva dos níveis de glicose. A funcionalidade “consultar nível de glicose dos alimentos (índice glicêmico e carga glicêmica)” foi implementada no protótipo na 2^o iteração, através dela o usuário tem acesso a uma tabela com vários alimentos e pode consultar o índice glicêmico e carga glicêmica dos alimentos, esses valores ajudam a identificar quando a ingestão excessiva de glicose pode estar ocorrendo e quais os principais alimentos que estão relacionados à hipoglicemia e hiperglicemia.

A maior meta da terapia nutricional é alcançar o controle glicêmico, evitando os picos de glicemia (SACHS, et al., 2009); assim, a disponibilização dos valores de glicose de cada alimento se torna uma ferramenta de importância significativa dentro das tecnologias eHealth. Além disso, houve a preocupação de tornar essas informações acessíveis a usuários mais leigos.

Acompanhado dos valores de carga glicêmica e índice glicêmico, a telas desenvolvidas para o protótipo informam o que significa aquele valor e qual o impacto dele na alimentação de um indivíduo com diabetes, como demonstrado na Figura 28-C.

A partir das descobertas deste estudo, ficou claro que, para obter uma entrada completa e confiável de informações dietéticas e terapia nutricional, as aplicações de dieta existentes precisam de melhorias substanciais. Para desenvolver um recurso que disponibilize informações nutricionais, dietas, e permita realizar um planejamento alimentar focado em pacientes diabéticos, são necessárias habilidades que vão além dos conhecimentos de desenvolvimento de software.

Para implementar tal recurso no protótipo, seria necessário investir mais tempo e procurar informações além das que estão disponíveis na literatura, entrar em contato com profissionais de nutrição e com pacientes diabéticos que fazem acompanhamento nutricional. Nesse sentido, a terceira iteração, que engloba os casos de uso relacionados ao planejamento alimentar, não foi implementada principalmente devido a necessidade de maior aprofundamento no assunto.

Entretanto, as informações de índice glicêmico e carga glicêmica dos alimentos oferecidas para consulta no protótipo desenvolvido são um avanço diante das demais soluções do mercado. Poucos aplicativos possuem essa funcionalidade, e, dentro desse montante, a maioria não a oferece de forma gratuita ao usuário ou não estão integradas a outros recursos igualmente necessários.

Por fim, para validar a usabilidade e coletar *feedbacks*, faz-se necessário a avaliação da solução por usuários e profissionais da saúde.

6 CONCLUSÃO

O presente trabalho apresentou o desenvolvimento do protótipo de uma aplicação mobile para o autogerenciamento do diabetes mellitus, contemplando os principais recursos presentes nos aplicativos existentes no mercado, com uma abordagem simples e acessível, focada no autocuidado e na usabilidade. A revisão de literatura permitiu mapear o estado da arte em tecnologias m-health para Diabetes Mellitus e foi apresentado um comparativo entre os principais aplicativos usados no gerenciamento e monitoramento à Diabetes Mellitus e uma solução que engloba o maior numero de funcionalides foi desenvolvida.

Os conceitos de engenharia de software e o paradigma de orientação a objetos foram explorados e utilizados na implementação do aplicativo MOGLIE, nativo para dispositivos moveis com o sistema operacional Android. A construção do protótipo traz um conjunto de experiências aprendidas durante o desenvolvimento de sua arquitetura que facilitara inclusão de novos componentes e utilização de sensores.

Como trabalhos futuros, sugere-se a execução das iterações restantes do aplicativo, bem como a implantação, testes e avaliação com usuários e profissionais de saúde. Assim, será possível receber feedbacks, implementar melhorias e publicar o aplicativo finalizado na Google Play Store.

7 REFERÊNCIAS

- ALOTAIBI, M. M.; ISTEPLANIAN, R.; PHILIP, N. **A mobile diabetes management and educational system for type-2 diabetics in Saudi Arabia (SAED)**. mHealth, AME Publishing Company, v. 2, p. 33, aug 2016. ISSN 2306-9740.
- AMERICAN DIABETES ASSOCIATION (ADA). **Standards of Medical Care in Diabetes2015**. Diabetes Care, v. 38, n. January, p. S1–S2, 2015. ISSN 19355548.
- APPANNIE. **Global App Downloads Surpassed 175 Billion in 2017**. 2018. 163 p. Disponível
- ÅRSAND, E., MUZNY, M., BRADWAY, M., et al. **Performance of the First Combined Smartwatch and Smartphone Diabetes Diary Application Study**. J Diabetes Sci Technol. 2015; 9(3): 556-63.
- ARYANA, B. et al. **Application of Object Oriented Thinking in Product Design: Design Process of Personal Digital Partner**. In: International Association Of Societies Of Design Research – IASDR, 3., 2007, Hong Kong. Proceedings...Hong Kong: The Hong Kong Polytechnic University School of Design, 2007.
- ASHOORKHANI, M. et al. **Comparing the effectiveness of the BPMAP (Blood Pressure Management Application) and usual care in self-management of primary hypertension and adherence to treatment in patients aged 30-60 years: study protocol for a randomized controlled trial**. Trials, v. 17, n. 1, p. 511, 2016.
- ATKINSON, F. S., et al. **International tables of glycemic index and glycemic load values 2021: a systematic review**. The American Journal of Clinical Nutrition, 2021. Disponível em: <<https://doi.org/10.1093/ajcn/nqab233>>. Acesso em: 2 ago. 2021.
- AVAZU. **Global Internet Industry Research**. 2016. Disponível em: <[http://avazuinc.com/20Brazil\(Full%20Edition\).pdf](http://avazuinc.com/20Brazil(Full%20Edition).pdf)>. Acesso em: 28 mar. 2021.
- AZEVEDO, P. R. et al. **Health education shares in the context of chronic diseases: integrative review / Ações de educação em saúde no contexto das doenças crônicas: revisão integrativa**. Revista de Pesquisa: Cuidado é Fundamental Online, v. 10, n. 1, p. 260, jan. 2018.
- BADHAM, J. et al. **Developing agent-based models of complex health behaviour**. Health and Place, Elsevier Ltd, v. 54, n. August, p. 170–177, 2018. ISSN 18732054.
- BIØRN-HANSEN, A.; MAJCHRZAK, T. A.; GRØNLI, T.-M. **Progressive Web Apps: The Possible Web-native Unifier for Mobile Development**. Proceedings of the 13th International Conference on Web Information Systems and Technologies, n. Webist, p. 344–351, 2017.
- BOOCH, G. **Object-oriented Analysis and Design with Applications**. 3. ed. Boston: Addison-Wesley Professional, 2007. 720p. ISBN 9780201895513.
- BRANDÃO, D. S.; PINHEIRO, M. C. (Org.). **Estratégias para o cuidado da pessoa com doença crônica: diabetes mellitus**. 1. ed. Brasília - DF: [s.n.], 2016. v. 35. 709–17; discussion 717 p. ISSN 1530-891X. ISBN 9788533420595.

CAVALCANTE, R. B., FERREIRA, M. N., MAIA, L. L., et al. **Use of information technologies and communication in health education for adolescent students.** J Health Inform. 2012; 4(4):182-6.

CECHETTI, N. P. et. al. **Developing and implementing a gamification method to improve user engagement: A case study with an m-Health application for hypertension monitoring.** Telematics and Informatics, p. 1-33, abr. 2019.

CHEAH, W. et al. **Agent-oriented requirement engineering for mobile application development.** International Journal of Interactive Mobile Technologies, v. 11, n. 6, p. 32–48, 2017. ISSN 18657923.

CHOMUTARE, T., FERNANDEZ-LUQUE, L., ÅRSAND, E. AND HARTVIGSEN, G. (2011). **Features of Mobile Diabetes Applications: Review of the Literature and Analysis of Current Applications Compared Against Evidence-Based Guidelines.** J Med Internet Res, v. 13, n. 3.

COELHO, Camila Arnellas; SARTORELLI, Reinaldo Coelho. **Persistência de Objetos via Mapeamento Objeto-Relacional.** 2004. 110 f. TCC (Graduação) - Curso de Sistemas de Informação, Computação e Informática, Universidade Presbiteriana Mackenzie, São Paulo, 2004. Cap. 1. Disponível em: <<http://www.dominiopublico.gov.br/download/texto/ea000253.pdf>>. Acesso em: 01 jun. 2021

COSTA, R. E. **Bancos de Dados Relacionais.** 2011. TCC (Tecnólogo) – Curso de Tecnologia em Processamento de Dados. Faculdade de Tecnologia de São Paulo. São Paulo. 2011. Disponível em: <<http://www.fatecsp.br/dti/tcc/tcc0025>>. Acesso em: 01 jun. 2021.

DATE, C.J.; Int. a **Sistemas de Bancos de Dados**, tradução da 4a.edição norte-americana, Editora Campus, 1991.

DEBON, R., COLEONE, J. D., BELLEI, E. A., AND DE MARCHI, A. C. B. **Mobile health applications for chronic diseases: A systematic review of features for lifestyle improvement.** 2019. Diabetes Metab Syndr Clin Res Rev, v. 13, n. 4

DEBON, R., COLEONE, J. D., BELLEI, E. A., et al. **Mobile health applications for chronic diseases: A systematic review of features for lifestyle improvement.** 2019. Diabetes Metab Syndr Clin Res Rev, v. 13, n. 4

DEITEL, H.; DEITEL, P.; DEITEL, A. **Android: Como programar - 2ed.** Bookman Editora, 2015. ISBN 9788582603482. Disponível em: <<https://books.google.com.br/books?id=aGMfCgAAQBAJ>>. Acesso em: 15 nov. 2020. em: <<https://www.appannie.com/en/insights/market-data/global-app-downloads-2017/>>. Acesso em: 29 mar. 2021.

DINSOREANU, M.; SALOMIE, I.; PUSZTAI, K. **On the design of agent-based systems using UML and extensions.** Proceedings of the International Conference on Information Technology Interfaces, ITI, p. 205–210, 2002. ISSN 13301012.

FERGUSON C., JACKSON D. **Selecting, appraising, recommending and using mobile applications (apps) in nursing.** J Clin Nurs. 2017 [acesso em: 03 dez. 2018];26(21-22):3253-5. Disponível em: <<https://doi.org/10.1111/jocn.13834>>. Acesso em: 16 jun. 2021.

FERNANDES, H. M. **Banco de Dados Relacional (SQL) e Não Relacional (NoSQL) – O que são, para que servem e qual a diferença?** 2020. Disponível em: <<https://marquesfernandes.com/tecnologia/banco-de-dados-relacional-sql-e-nao-relacional-nosql-o-que-sao-para-que-servem-e-qual-a-diferenca/>>. Acesso em: 10 set. 2021.

FIREBASE. **Firebase Realtime Database.** 2021. Disponível em: <<https://Firebase.google.com/docs/database>>. Acesso em: 01 set. de 2021.

FONSECA DE OLIVEIRA, A. R.; ALENCAR, M. S. DE M. **O uso de aplicativos de saúde para dispositivos móveis como fontes de informação e educação em saúde.** RDBCi: Revista Digital de Biblioteconomia e Ciência da Informação, v. 15, n. 1, p. 234, 31 jan. 2017

FROISLAND, D.; ARSAND E., SKARDERUD, F. **Improving diabetes care for young people with type 1 diabetes through visual learning on mobile phones: Mixed-methods study.** J Med Internet Res. 2012; 14(4):e111

GAMA, M. P. R. **Do milagre canadense do século xx às esperanças de cura do século xxi (editorial).** Endocrinologia & Diabetes Clínica e Experimental, 2 (2), p. 3–5, 2002.

GOLBERT, A., VASQUES, A. C. J., FARIA A. C. R. A. et al **Diretrizes da Sociedade Brasileira de Diabetes 2019-2020.** São Paulo: Clannad, 2019.

GOOGLE PLAY, 2016. **Aplicativos para diabetes.** Disponível em: <<https://play.google.com/store/search?q=diabetes&c=apps&docType=1&sp=CAFiCgoIZGlhYmV0ZXN6BRgAwAECigECCAE%3D:S:ANO1ljL7uGQ>>. Acesso em: 04 jan. 2021.

GORDON, W. J., BHAN, I. **Graphical timeline software for inpatient medication review.** Health Informatics J 2017. [Epub ahead of print]; DOI: 10.1177/1460458216682355.

GRIFFITHS, S. **Google's Mobile App UX Principles.** 2015. Disponível em: <https://d3atsf3fgek2rw.cloudfront.net/content/uploads/2015/04/Mobile-App-UX-principles_Full-Report_Final.pdf>. Acesso em: 20 abr. 2021.

GUDWIN, R. R., **Diagramas de Atividade e Diagramas de Estado,** 2021 DCA-FEEC-UNICAMP. Disponível em: <<https://www.dca.fee.unicamp.br/~gudwin/ftp/ea976/AtEst.pdf>>. Acesso em: 25 jul. 2021

GUEDES, T. A. **UML 2: uma abordagem prática.** São Paulo: Novatec, 2011.

GUEDES, G. T. A.; VICARI, R. M. **Applying a UML profile in the requirements modeling to multi-agents systems.** Proceedings - 7th International Conference on the Quality of Information and Communications Technology, QUATIC 2010, p. 216–221, 2010. ISSN 17519918.

HSU, W. C. et al. **Utilization of a Cloud-Based Diabetes Management Program for Insulin Initiation and Titration Enables Collaborative Decision Making Between Healthcare**

Providers and Patients. Diabetes Technology & Therapeutics, Mary Ann Liebert, Inc., 140 Huguenot Street, 3rd FloorNew Rochelle, NY 10801USA, v. 18, n. 2, p. 59–67, feb 2016. ISSN 1520-9156.

IBGE. Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal. Rio de Janeiro: IBGE, 2016. ISBN 9788524044458.

INTERNATIONAL DIABETES FEDERATION (IDF). **Diabetes Atlas.** 8th editio. ed. [s.n.], 2017. ISBN 9782930229874. Disponível em: <<https://www.idf.org/e-library/welcome.html>>. Acesso em: 14 abr. 2021.

JOBE, W. **Native Apps Vs. Mobile Web Apps.** International Journal of Interactive Mobile Technologies (iJIM), v. 7, n. 4, p. 27, 2013. ISSN 1865-7923.

JONGH, D. **Cochrane Database of Systematic Reviews Mobile phone messaging for facilitating self-management of long-term illnesses (Review).** n. 12, 2012.

KIM, M. I. **An integrated graphic scheme for the display of insulin prescription and blood glucose information.** Diabetes Technol Ther 2006;8:505–512.

KO, I., CHANG, H. **Interactive data visualization based on conventional statistical findings for antihypertensive prescriptions using National Health Insurance claims data.** Int J Med Inform 2018;116:1–8

KORTH, H.F. e SILBERSCHATZ, A.; **Sistemas de Bancos de Dados.** Makron Books, 2a. edição revisada, 1994.

KUMAH-CRYSTAL, Y. A., HOOD, K. K., HO, Y. X., et al. **Technology use for diabetes problem solving in adolescents with type 1 diabetes: relationship to glycemic control.** Diabetes Technol Ther. 2015; 17(7): 449-54.

LECHETA, R. R. **Google Android-3^aEdição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK.** [S.l.]: Novatec Editora, 2013.

LISANGELA, R. **O Diabetes Mellitus E a Hemoglobina Glicada E Sua Correlação Com a Glicemia Diabetes Mellitus and the Glycated Hemoglobin and Its Correlation.** Revista Saúde e Desenvolvimento, vol. 3, p. 18, 2013.

LATIF, M. et al. **Review of mobile cross platform and research orientations.** 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2017, p. 0–3, 2017.

MARCO, A. D.; PACE, S. **Model-driven approach to Agilla Agent generation.** 2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013, p. 1482–1487, 2013. ISSN 2376-6492.

MENDES, A. B. V., FITTIPALDI, J. A. S., NEVES, R. C. S. et al. **Prevalence and correlates of inadequate glycaemic control: results from a nationwide survey in 6,671 adults with**

diabetes in Brazil. 2009. Acta Diabetologica, 47(2), 137–145. doi:10.1007/s00592-009-0138-z

MIDDELWEERD A., MOLLEE J. S., BRUG J., et al. **Apps to promote physical activity among adults: a review and content analysis.** Int J Behav Nutr Phys Act. 2014. Disponível em: <<https://doi.org/10.1186/s12966-014-0097-9>>. Acesso em: 15 fev. 2021.

MULVANEY, S. A., ROTHMAN, R. L., DIETRICH, M. S., et al. **Using mobile phones to measure adolescent diabetes adherence.** Health Psychol. 2012; 31(1):43-50.

mysugr GMBH. **mySugr: Diário da Diabetes.** Google Play, 2018. Disponível em: <https://play.google.com/store/apps/details?id=com.mysugr.android.companion&hl=pt_BR&gl=US>. Acesso em: 11 out. 2020.

NAHAR, P. et al. **mHealth and the management of chronic conditions in rural areas: a note of caution from southern India.** Anthropology & Medicine, v. 24, n. 1, p. 1–16, 2 jan. 2017

NUNDY, S. et al. **Developing a behavioral model for mobile phone-based diabetes interventions.** Patient Education and Counseling, Elsevier Ireland Ltd, n. 1, p. 125–132, 2013. ISSN 07383991.

OLIVEIRA, J. E. P. de (Org.). **Diretrizes da Sociedade Brasileira de Diabetes 2017-2018.** São Paulo: Clannad, 2017. 383 p. ISBN 9788593746024.

PAGLIALONGA, A.; LUGO, A.; SANTORO, E. **An overview on the emerging area of identification, characterization, and assessment of health apps.** Journal of Biomedical Informatics, v. 83, p. 97–102, jul. 2018.

PAGOTO, S., SCHNEIDER, K., JOJIC, M., et al.. **Evidence-Based Strategies in Weight-Loss Mobile Apps.** Am J Prevent Med. 2013; 45 (5):576-82.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição.** [S.l.]: McGraw Hill Brasil, 2016.

QUE, P.; GUO, X.; ZHU, M. **A Comprehensive Comparison between Hybrid and Native App Paradigms.** Proceedings - 2016 8th International Conference on Computational Intelligence and Communication Networks, CICN 2016, p. 611–614, 2017.

REYES A., QIN P., BROWN C. A. **A standardized review of smartphone applications to promote balance for older adults.** Disabil Rehabil. 2018. Disponível em: <<https://doi.org/10.1080/09638288.2016.1250124>> Acesso em: 13 jun. 2021.

RUSSELL, S.; DOYLE, O.; COLLIER, R. W. **Developing Android applications using agent-oriented programming.** Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017, v. 2018-January, p. 1–7, 2018.

RZEWUSKA M., AZEVEDO-MARQUES, J. M., COXON, D. et al. **Epidemiology of multimorbidity within the Brazilian adult general population: evidence from the 2013 National Health Survey (PNS 2013).** PLoS One. 2017;12(2):e0171813

SARKAR, U. et al. **Usability of Commercially Available Mobile Applications for Diverse Patients.** Journal of General Internal Medicine, v. 31, n. 12, p. 1417–1426, Acesso em: 18 ago. 2021.

SACHS, A. et al. **Manual oficial de contagem de carboidratos para profissionais da saúde /Sociedade Brasileira de Diabetes.** Rio de Janeiro: Diagraphic, 2009. Disponível em: <<http://crn5.org.br/wpcontent/uploads/2013/05/Manual-de-Contagem-deCarboidratos.pdf>>. Acesso em: 12 out 2021.

SCHAAN, B. D., HARZHEIM, E., GUS, I. **Perfil de risco cardíaco no diabetes mellitus e na glicemia de jejum alterada.** Rev Saúde Pública. 2014. Disponível em: <<https://www.scielo.br/j/rsp/a/HNd393MQmZTrRC3SWMZ6vmb/?lang=pt>>. Acesso em: 18 ago. 2021

SIRMA MEDICAL SYSTEMS. **Diabetes:M.** Google Play, 2018. Disponível em: <https://play.google.com/store/apps/details?id=com.mydiabetes&hl=pt_BR&gl=US>. Acesso em: 11 out 2020.

SKRØVSETH, S. O., A°RSAND, E., GODTLIEBSEN, F., et al. **Data-driven personalized feedback to patients with type 1 diabetes: A randomized trial.** Diabetes Technol Ther 2015;17:482–489.

SMYTH, N. **Firebase Essentials - Android Edition:**. CreateSpace Independent Publishing Platform, 2017. ISBN 9781546660330.

SOCIAL DIABETES. **SocialDiabetes.** Google Play, 2018. Disponível em: <https://play.google.com/store/apps/details?id=com.socialdiabetes.android&hl=pt_BR&gl=US>. Acesso em: 11 out. 2020.

SOCIEDADE BRASILEIRA DE DIABETES. **Diabetes - Diagnóstico e Tratamento.** 2018. Disponível em: <<http://www.diabetes.org.br/publico/diabetes/diagnostico-e-tratamento>>. Acesso em: 14 abr. 2021.

SOUZA, I. **O que é SQLite, por que ele é usado, e o que o diferencia do MySQL?.** 2020. Disponível em: <<https://rockcontent.com/br/blog/sqlite/>> Acesso em: 01 set. de 2021.

SPICHLER, E. R., SPICHLER, D., LESSA I., et al. **Capture-recapture method to estimate lower extremity amputation rates in Rio de Janeiro, Brazil.** 2001 .Pan Am J Public Health. 2001;10(5):334-40.

SQLITE. **About SQLite.** 2017. Disponível em: <<https://www.sqlite.org/about.html>>. Acesso em: 01 set. de 2021.

TIBES C. M. S., DIAS J. D., ZEM-MASCARENHAS S. H. **Aplicativos móveis desenvolvidos para a área da saúde no Brasil: revisão integrativa da literatura.** REME Rev Min Enferm. 2014. Disponível em: <<https://doi.org/10.5935/1415-2762.20140035>>. Acesso em: 15 fev. 2021.

TIEFENGRABNER, M., DOMHARDT, M., OOSTINGH, G. J., et al. **Can smartphone-based logging support diabetologists in solving glycemic control problems?** Clifton: IOS Press Book; 2014 v.198. p. 188-95. [Studies in Health Technology and Informatics].

USE MOBILE. **Aplicativo nativo, web App ou aplicativo híbrido?**. 2020. Disponível em: <<https://usemobile.com.br/aplicativo-nativo-web-hibrido/>>. Acesso em: 26 mai. 2021.

WHO. **Global Observatory for eHealth. MHealth: new horizons for health through mobile technologies.** Geneva, SW: World Health Organization; 2011 .Disponível em: <http://apps.who.int/iris/bitstream/handle/10665/44607/9789241564250_eng.pdf>. Acesso em: 03 dez. 2020.

WHO. **Global Report on Diabetes.** 2016. Disponível em: <<http://www.diabetes.org.br/publico/diabetes/diagnostico-e-tratamento>>. Acesso em: 21 mar. 2020.

WONG, J. C., NEINSTEIN, A. B., LOOK, H., et al. **Pilot study of a novel application for data visualization in type 1 diabetes.** J Diabetes Sci Technol 2017;11:800–807

WU, Y., YAO, X., VESPASIANI, G., et al. **Mobile App-Based Interventions to Support Diabetes Self-Management: A Systematic Review of Randomized Controlled Trials to Identify Functions Associated with Glycemic Efficacy.** 2017. JMIR mHealth and uHealth, v. 5, n. 3.