

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

PROTÓTIPO DE APLICATIVO PARA ACOMPANHAMENTO
E CONTROLE DA GLICEMIA

TIAGO DIONESTO WILLRICH DA SILVA

BLUMENAU
2015

2015/1-14

TIAGO DIONESTO WILLRICH DA SILVA

**PROTÓTIPO DE APLICATIVO PARA ACOMPANHAMENTO
E CONTROLE DA GLICEMIA**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Sistemas
de Informação — Bacharelado.

Prof. Mauro Marcelo Mattos, Doutor – Orientador

**BLUMENAU
2015**

2015/1-14

PROTÓTIPO DE APLICATIVO PARA ACOMPANHAMENTO E CONTROLE DA GLICEMIA

Por

TIAGO DIONESTO WILLRICH DA SILVA

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Mauro Marcelo Mattos, Doutor - Orientador, FURB

Membro: _____
Prof. Aurélio Faustino Hoppe, Mestre - FURB

Membro: _____
Prof. Roberto Heinzle, Doutor - FURB

Blumenau, 07 de julho de 2015.

Dedico este trabalho a todos os amigos e familiares, especialmente aqueles que me ajudaram diretamente na realização deste.

Uma viva inteligência de nada serve se não estiver ao serviço de um carácter justo; um relógio não é perfeito quando trabalha rápido, mas sim quando trabalha certo.

Luc de Clapiers Vauvenargues

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça.

Aos meus pais Dion e Janine, pelo incentivo, apoio e amor.

Aos meus amigos Diogo Lenzi, Jonathan Silva Santos, Paulo Roberto, Paula Petrizi e muitos outros, pela ajuda e pelos momentos de descontração que me mantiveram animado.

Ao meu orientador, professor Mauro Marcelo Mattos, por ter contribuído com suas sugestões e incentivado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

RESUMO

A diabetes é uma doença metabólica que se não tratada corretamente pode acarretar em complicações sérias à saúde, exigindo em alguns casos o monitoramento diário dos níveis de glicose no sangue. Este trabalho consiste no desenvolvimento de um protótipo de aplicativo para auxiliar os diabéticos na manutenção de informações referentes ao tratamento de sua doença e possibilitar o acompanhamento de seu desempenho por um profissional de saúde. O principal objetivo do aplicativo é permitir que o diabético possa salvar e disponibilizar para seu endocrinologista as medições de glicemias, usos de medicamentos e rotina que deseja seguir, juntamente com suas anotações, para que o endocrinologista possa ter uma noção aprimorada do desempenho diário do seu paciente. O aplicativo foi desenvolvido na linguagem de programação Java para o sistema operacional Android, sincronizando as informações do usuário com um *web service*, também desenvolvido em Java, e utilizando o sistema de gerenciamento de banco de dados MySQL. O protótipo construído validou o conjunto de requisitos estabelecidos, indicando que a sua aplicação prática pode ser um instrumento de melhoria da relação paciente-endocrinologista.

Palavras chaves: Aplicativo. Controle Glicêmico. Diabetes.

ABSTRACT

Diabetes is a metabolic disease that if not treated properly can result in serious health complications, requiring in some cases the daily monitoring of glucose levels in the blood. This work consists in developing a prototype application to help diabetics in maintaining information regarding the treatment of their disease and allow monitoring of their performance by a health professional. The main objective is to allow the diabetic person save and provide for his endocrinologist the measurements of blood glucose, medication use and other personal routine follows along with his own notes, in such a way that the endocrinologist may have a broader overview of the daily performance of his patient. The application was developed in Java programming language to the Android operating system, synchronizing user information with a web service, also developed in Java, and using the database management system MySQL. The prototype built validated the set of established requirements, indicating that its practical application can be a tool for improving patient-endocrinologist relationship.

Key-Words: Application. Glycemic Control. Diabetes.

LISTA DE FIGURAS

Figura 1 - Glicosímetro e lancetador para AMGC	20
Figura 2 - Iniciativas de mHealth no Brasil.....	21
Figura 3 - Provedores de mHealth.....	22
Figura 4 - Tipos de pesquisas sobre mHealth.....	23
Figura 5 - Tela principal e gráfico gerado pelo aplicativo GlicoCare.....	24
Figura 6 - Tela principal do aplicativo OnTrack	25
Figura 7 - Opções de gráficos do aplicativo OnTrack.....	25
Figura 8 - Tela principal do aplicativo DiabetesControl	26
Figura 9 - Exemplo de gráfico gerado pelo aplicativo DiabetesControl	27
Figura 10 - Diagrama de Casos de Uso	31
Figura 11 - Diagrama de Atividades do processo de abertura do aplicativo.....	32
Figura 12 - Diagrama de Atividades do processo de cadastro de glicemia.....	33
Figura 13 - Diagrama de Atividades do processo de configuração de um endocrinologista ...	34
Figura 14 - Diagrama de classes do servidor.....	35
Figura 15 - Diagrama de classes do aplicativo	36
Figura 16 - Arquitetura do sistema.....	37
Figura 17 - Classe modelo da entidade <code>User</code>	38
Figura 18 - Classe responsável pelas operações no banco de dados da entidade <code>User</code>	39
Figura 19 - Classe onde são definidas as propriedades da tabela para a entidade <code>User</code>	39
Figura 20 - Parte do arquivo XML referente à tela de cadastro	40
Figura 21 - Classe responsável pela tela de cadastro.....	41
Figura 22 - Declaração da <i>activity</i> da tela de cadastro no arquivo <code>AndroidManifest.xml</code>	41
Figura 23 - Classe do <i>web service</i> responsável pelas requisições referentes às glicemias.....	42
Figura 24 - Código responsável pelas requisições HTTP através do método POST	43
Figura 25 - Método responsável pela conversão de código JSON em um único objeto	43
Figura 26 - Código responsável pela sincronização de glicemias	44
Figura 27 - Tela de <i>login</i> e tela de cadastro.....	45
Figura 28 - Menu principal do paciente.....	46
Figura 29 - Telas de histórico e de cadastro de glicemia.....	46

Figura 30 - Telas de lista e cadastro de medicamentos	47
Figura 31 - Telas de histórico e de cadastro de uso de medicamento	48
Figura 32 - Tela de rotina e tela de cadastro de item de rotina.....	48
Figura 33 - Tela de opções de gráficos.....	49
Figura 34 - Gráfico gerado pelo aplicativo e <i>e-mail</i> contendo-o.....	49
Figura 35 - Tela de <i>chat</i> com endocrinologista	50
Figura 36 - Tela de configurações do paciente.....	51
Figura 37 - Menu principal do endocrinologista	51
Figura 38 - Tela com a lista de pacientes e tela com detalhes do paciente	52
Figura 39 - Visualização do histórico de glicemias do paciente	52
Figura 40 - Modelo Entidade Relacionamento do banco de dados do servidor	70
Figura 41 - Modelo Entidade Relacionamento do banco de dados do aplicativo	71

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais	29
Quadro 2 - Requisitos Não Funcionais.....	30
Quadro 3 - Comparativo com trabalhos correlatos.....	53
Quadro 4 - Caso de Uso 01 - "Efetuar cadastro"	61
Quadro 5 - Caso de Uso 02 - "Efetuar <i>login</i> "	61
Quadro 6 - Caso de Uso 03 - "Efetuar <i>logoff</i> "	62
Quadro 7 - Caso de Uso 04 - "Configurar endocrinologista"	62
Quadro 8 - Caso de Uso 05 - "Gerenciar histórico de glicemias"	63
Quadro 9 - Caso de Uso 06 - "Gerenciar o histórico de uso de medicamentos"	64
Quadro 10 - Caso de Uso 07 - "Configurar itens de rotina"	65
Quadro 11 - Caso de Uso 08 - "Parametrizar intervalo de glicemia desejado"	66
Quadro 12 - Caso de Uso 09 - "Conversar com chat por histórico"	66
Quadro 13 - Caso de Uso 10 - "Visualizar gráficos de performance"	67
Quadro 14 - Caso de Uso 11 - "Gerar e-mail contendo gráfico"	67
Quadro 15 - Caso de Uso 12 - "Visualizar lista de pacientes"	67
Quadro 16 - Caso de Uso 13 - "Visualizar informações do paciente"	67
Quadro 17 - Caso de Uso 14 - "Remover paciente"	68
Quadro 18 - Caso de Uso 15 - "Visualizar históricos do paciente"	68
Quadro 19 - Caso de Uso 16 - "Visualizar rotina do paciente"	69
Quadro 20 - Tabela "User"	72
Quadro 21 - Tabela "Glycemia"	73
Quadro 22 - Tabela "MedicationUse"	74
Quadro 23 - Tabela "Medication"	75
Quadro 24 - Tabela "RoutineItem"	75
Quadro 25 - Tabela "Message"	76
Quadro 26 - Tabela "User"	78
Quadro 27 - Tabela "Glycemia"	79
Quadro 28 - Tabela "MedicationUse"	80
Quadro 29 - Tabela "Medication"	81
Quadro 30 - Tabela "RoutineItem"	82

Quadro 31 - Tabela "Message"	83
------------------------------------	----

LISTA DE SIGLAS

AMGC - Auto monitoramento da Glicemia Capilar

ADT - *Android Developer Tools*

DG - Diabetes Gestacional

DM - *Diabetes Mellitus*

DM1 - *Diabetes Mellitus* tipo 1

DM2 - *Diabetes Mellitus* tipo 2

EA - *Enterprise Architect*

HTTP - *Hypertext Transfer Protocol*

IDE - *Integrated Development Environment*

JSON - *JavaScript Object Notation*

MER - Modelo de Entidade e Relacionamento

OMS - Organização Mundial de Saúde

REST - *Representational State Transfer*

SGBD - Sistema de Gerenciamento de Banco de Dados

SQL - *Structured Query Language*

URL - *Uniform Resource Locator*

XML - *Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVOS DO TRABALHO	16
1.2 ESTRUTURA DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 DIABETES.....	18
2.2 CONTROLE GLICÊMICO.....	19
2.3 MOBILE HEALTH.....	20
2.4 TRABALHOS CORRELATOS	23
2.4.1 GlicoCare	23
2.4.2 OnTrack.....	24
2.4.3 DiabetesControl.....	26
3 DESENVOLVIMENTO DO PROTÓTIPO	28
3.1 LEVANTAMENTO DE INFORMAÇÕES	28
3.2 ESPECIFICAÇÃO	29
3.2.1 Requisitos Funcionais	29
3.2.2 Requisitos Não-Funcionais	30
3.2.3 Diagrama de Casos de Uso	30
3.2.4 Diagramas de Atividades	31
3.2.5 Diagramas de Classes.....	34
3.3 IMPLEMENTAÇÃO	36
3.3.1 Ferramentas e Técnicas Utilizadas.....	36
3.3.1.1 Ferramentas.....	37
3.3.1.2 Arquitetura do Sistema	37
3.3.1.3 Persistência	38
3.3.1.4 Interface	40
3.3.1.5 Comunicação com servidor	41
3.3.1.6 Sincronização de dados.....	44
3.3.2 Operacionalidade da Implementação	45
3.3.2.1 Paciente.....	45
3.3.2.2 Endocrinologista	51

3.4 RESULTADOS E DISCUSSÃO	53
3.4.1 Comparativo com Trabalhos Correlatos	53
4 CONCLUSÕES.....	56
4.1 EXTENSÕES	56
REFERÊNCIAS	58
APÊNDICE A – Descrição dos Casos de Uso	61
APÊNDICE B – Modelos de Entidade e Relacionamento	70
APÊNDICE C – Dicionário de Dados do Servidor.....	72
APÊNDICE D – Dicionário de Dados do Aplicativo	78

1 INTRODUÇÃO

Atualmente, existem 382 milhões de pessoas diabéticas no mundo, cerca de 8,3% da população adulta. Sem ações em escala global para a prevenção da doença, esse número deve subir para 592 milhões em menos de 25 anos. No Brasil, cerca de 11,9 milhões de pessoas entre 20 e 79 anos são diabéticos (INTERNATIONAL DIABETES FEDERATION, 2013).

Mesmo sendo uma doença com sérias implicações quando não tratada corretamente, muitos diabéticos não procuram acompanhamento adequado ou não seguem o tratamento proposto a eles até que as consequências da patologia se tornem claras, como a falência de órgãos ou a necessidade de amputação de membros (PÉRES et al., 2007).

Como as consequências da diabetes estão diretamente relacionadas ao quadro de hiperglicemia gerado pela doença, é necessário que os níveis glicêmicos de seus portadores sejam controlados constantemente, através do automonitoramento da glicemia capilar e da utilização de medicamentos (INTERNATIONAL DIABETES FEDERATION, 2013).

De acordo com Brasil (2007), portadores da diabetes do tipo 1¹ tem maior necessidade de acompanhar seus níveis de glicemia várias vezes ao dia do que portadores da diabetes do tipo 2² ou diabetes gestacional³. Porém, a exemplo de estudos como o de Costa (2010), todos podem se beneficiar da utilização de dispositivos móveis no seu tratamento, inclusive os não insulino-dependentes.

O mercado de dispositivos móveis, principalmente os celulares e *tablets*, está em evidência e conquistando um público cada vez maior, devido a diversidade de aparelhos e facilidade de aquisição. Segundo a Agência Nacional de Telecomunicações (2012), o Brasil encerrou o ano de 2011 com mais de 242 milhões de aparelhos celulares em operação, representando um crescimento de 19,4% em relação ao ano anterior, que contava com cerca de 202 milhões de aparelhos. Ou seja, com o passar dos anos, a procura por este mercado está cada vez maior, tornando-o muito atrativo para todas as áreas.

Segundo Kuszka (2014),

¹ Diabetes resultante da diminuição ou ausência da produção de insulina, devido à destruição imunológica das células beta no pâncreas.

² Diabetes resultante da resistência das células à insulina produzida.

³ Diabetes temporária resultante da resistência das células à insulina produzida, devido aos hormônios da gravidez.

“A utilização de dispositivos móveis acessando dados na nuvem não é uma tendência, já é um fato. Veremos cada vez mais diferentes usos para esse potente computador conectado na rede, cheio de sensores que fica no seu bolso. Estamos cada vez mais utilizando nossos dispositivos móveis para que possamos trabalhar, nos comunicar, buscar informações, nos divertir.”

Essa gama de utilizações disponíveis para dispositivos móveis se dá, em grande parte, pela possibilidade de instalação de aplicativos desenvolvidos por terceiros. Ainda que já existam no mercado muitas ferramentas para auxiliar o controle glicêmico, a maioria delas tem como foco apenas o papel do paciente no tratamento. Assim, não facilitam efetivamente a interação com o endocrinologista, que possui papel fundamental na construção de um plano de vida saudável para o diabético.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é disponibilizar um aplicativo Android que auxilie o usuário e o endocrinologista no acompanhamento e controle da glicemia.

Os objetivos específicos deste trabalho são:

- a) disponibilizar ao paciente recurso para registro de glicemias e de administração de medicamentos periódicos;
- b) disponibilizar ao endocrinologista recurso para visualização das informações fornecidas por seus pacientes;
- c) disponibilizar um servidor para sincronia de dados entre paciente e endocrinologista.

1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica sobre diabetes, controle glicêmico, *mobile health* e trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento da aplicação iniciando-se com o levantamento de informações, tendo na sequência a especificação de seus requisitos, contendo os requisitos funcionais e não-funcionais do sistema, diagramas de caso de uso, diagramas de atividades e diagramas de classes. Em seguida, a descrição da implementação, com técnicas e

ferramentas utilizadas e a operacionalidade da ferramenta. Por fim, são apresentados os resultados e discussão do trabalho.

No quarto capítulo tem-se as conclusões e as sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como aplicativos móveis, diabetes, controle glicêmico e os trabalhos correlatos.

2.1 DIABETES

Diabetes Mellitus (DM), ou simplesmente diabetes, é o nome dado ao grupo de doenças metabólicas que afetam negativamente a forma como as células do corpo humano absorvem a glicose presente no sangue, resultando num quadro de hiperglicemia. São três os principais tipos de diabetes mellitus, a diabetes mellitus tipo 1 (DM1), a diabetes mellitus tipo 2 (DM2) e a diabetes gestacional (DG). A DM1 é resultante da diminuição ou ausência da produção de insulina, devido à destruição imunológica das células beta no pâncreas. A DM1 ocorre mais frequentemente em crianças e jovens, e sua principal forma de tratamento consiste na aplicação de insulina em diferentes horas do dia. A DM2 é resultante da resistência das células à insulina produzida, ocorrendo mais frequentemente após os 40 anos de idade, sendo que normalmente a prática de exercícios físicos e uma dieta controlada são o suficiente para o controle adequado da doença. Assim como a DM2, a DG também é resultante da resistência das células à insulina produzida, porém desencadeada pelos hormônios da gravidez (INTERNATIONAL DIABETES FEDERATION, 2013, p. 22).

De acordo com World Health Organization (2011), a hiperglicemia tem impactos também a longo prazo, tais como a fragilidade vascular periférica, a lesão dos rins (nefropatia) ou a lesão dos nervos periféricos (neuropatia), podendo levar a doenças específicas da diabetes, como a perda de discriminação cromática, a retinopatia diabética, o pé diabético ou a fragilidade cardiovascular, com aumento do risco de enfarte do miocárdio e de acidentes vasculares cerebrais.

Conforme Gonçalves (2014),

A doença da diabetes tipo 2 assume a dimensão de uma epidemia global de difícil controle, com consequências na perda de capacidades, dependência e morte. Os reflexos econômicos e sociais são marcantes, pelo que são diversos os esforços na implementação de medidas preventivas e de tratamento. O carácter comportamental da doença ligada a hábitos de alimentação e sedentarismo da vida moderna, acentuado em idade mais avançada, torna mais complexa a aplicação de medidas de controle da doença. As tecnologias de informação e os dispositivos móveis permitem novas soluções para monitorização da doença, através de aplicações

interactivas para ajuda e motivação personalizada, adaptada às condições pessoais do doente.

A principal forma de diagnosticar a DM é através da medição da glicemia. Comumente, essa medição é efetuada em um exame laboratorial que coleta e analisa o sangue do paciente em jejum e/ou após a ingestão de uma quantidade substancial de glicose (MURAKAMI, 2007).

O tratamento da DM consiste em várias etapas, envolvendo o diabético, sua família e o profissional de saúde para máxima eficiência. O controle glicêmico com acompanhamento periódico, a adoção de hábitos saudáveis, o uso de medicamentos de forma correta e o tratamento de eventuais complicações cooperam para a melhoria da qualidade de vida do diabético (PÉRES et al., 2007).

2.2 CONTROLE GLICÊMICO

As complicações da DM estão diretamente ligadas aos efeitos tóxicos dos altos níveis de glicose no sangue. Assim, o controle glicêmico é a principal responsabilidade do diabético. O controle glicêmico inadequado resulta em complicações que degradam sua qualidade de vida e, conseqüentemente, reduzem sua expectativa de vida (INTERNATIONAL DIABETES FEDERATION, 2013, p. 24).

Segundo Guariente et al. (2002, p. 53),

“A automonitorização glicêmica é parte fundamental no tratamento do Diabetes Mellitus, uma vez que tal procedimento permite detectar e prevenir quadros hipoglicêmicos conferindo ao diabético a capacidade de reajustar as doses e tipo de insulina a fim de manter os níveis de glicose sanguíneo o mais próximo possível do normal, evitando ou retardando o aparecimento de complicações agudas e crônicas.”

O automonitoramento do nível de glicose do sangue por intermédio da medida da glicemia capilar é parte integrante do autocuidado das pessoas com diabetes mellitus insulino dependentes. A frequência do automonitoramento da glicemia capilar (AMGC) deve ser determinada individualmente, sendo que a frequência recomendada para quem possui a DM1 é de 3 a 4 vezes ao dia (BRASIL, 2007).

O AMGC é feito através de glicosímetros portáteis (Figura 1), em horários definidos pelo paciente junto ao seu profissional de saúde responsável. O AMGC antes e depois das refeições e atividades físicas pode auxiliar no entendimento do metabolismo do diabético,

contribuindo para estabelecer uma estimativa do nível de resposta à atividade física, insulinas e alimentos ingeridos (BRASIL, 2007).

Figura 1 - Glicosímetro e lancetador para AMGC



Fonte: Accu Chek (2015).

O exame do nível de glicemia capilar normalmente é feito através de uma pequena quantidade de sangue colhida na ponta de um dos dedos das mãos ou dos pés, que é perfurado por um lancetador. Essa quantidade de sangue é aplicada em uma tira reagente, que por sua vez é inserida no glicosímetro, que analisa e determina a quantidade de glicose no sangue (MURAKAMI, 2007).

2.3 MOBILE HEALTH

Mobile health, ou mHealth, é uma designação para a área que trata do uso de dispositivos móveis na saúde, envolvendo a aplicação da computação móvel, comunicações sem fio e tecnologias em rede para melhorar os diversos serviços e funções em que o paciente tem uma liberdade de mobilidade (PAWAR et al., 2012).

O crescimento de aplicativos mHealth tem sido extremamente rápido. Em 2011 nos EUA as estimativas eram de um mercado de US\$ 718 milhões e em 2012 algo em torno de US\$ 1,3 bilhões. Além disso, o número de aplicativos mHealth disponíveis na App Store da Apple chegou a 15.000 em setembro de 2011, um aumento significativo em relação aos 4.000 identificados em fevereiro de 2010 (WORLD BANK, 2012).

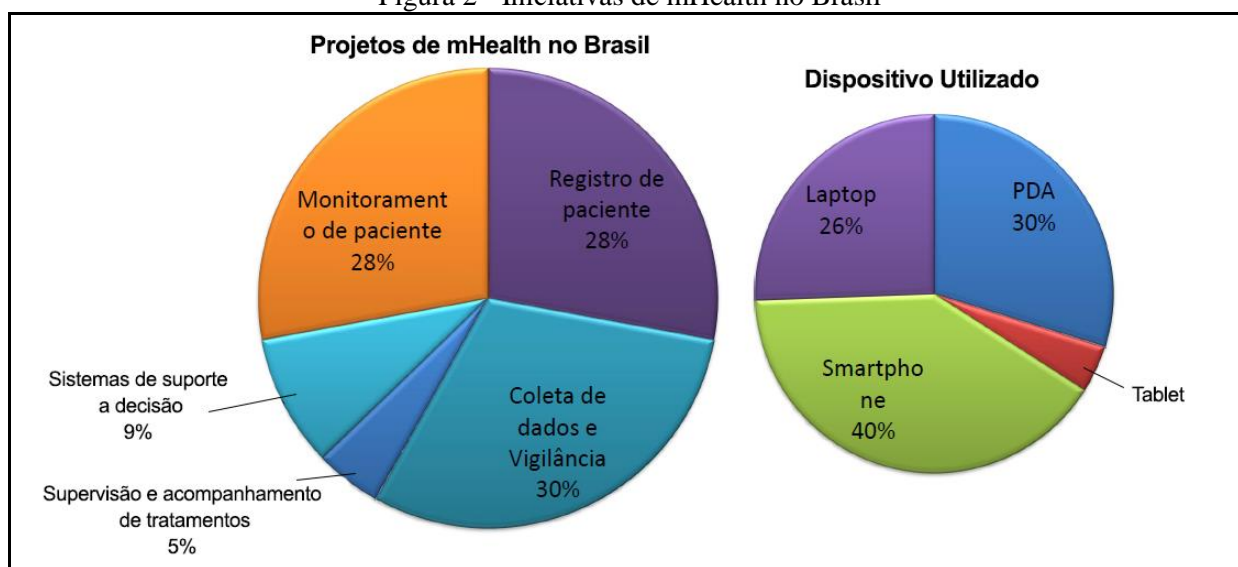
Conforme pesquisa efetuada pela Consumer Health Information Corporation (2011), foi revelado que "26% dos aplicativos mHealth são baixados e usados apenas uma vez. Das pessoas que confirmam estarem usando os aplicativos, 74% desistem até o dia 10 de uso". As razões para parar de usar foram identificadas como: "impreciso (10,2%)", "não se engajar (15,8%)", "interface com usuário não amigável (32,6%)", e "encontrado um melhor (34,4%)".

O Brasil é o país que possui o terceiro maior sistema de saúde do mundo, com cerca de 7 mil hospitais, 25 mil laboratórios, 17 mil clínicas e 125 mil consultórios médicos, a tecnologia da informação é geralmente utilizada apenas em atividades de rotina, como o agendamento de consultas e registros gerais (MARIN, 2010).

Os dispositivos móveis na área de saúde foram introduzidos com o uso dos *paggers*, na década de 70, que permitiam alertar o usuário através de *bips*. A tecnologia evoluiu para suportar envio de mensagens do servidor para os usuários e posteriormente para troca de mensagens bidirecionais e até correio de voz. Em 1996, aconteceu o ano de ouro dos *paggers* no Brasil, chegando à marca de 800 mil usuários (BARRETO, 2011).

Em uma pesquisa realizada por Iwaya et al. (2013), no período de junho a novembro de 2011, foram analisados através de buscas em publicações científicas, relatórios técnicos e publicações comerciais de produtos, 42 projetos envolvendo iniciativas de mHealth no Brasil. A Figura 2 apresenta a estratificação dos projetos em termos de área de abrangência e em termos de dispositivos utilizados.

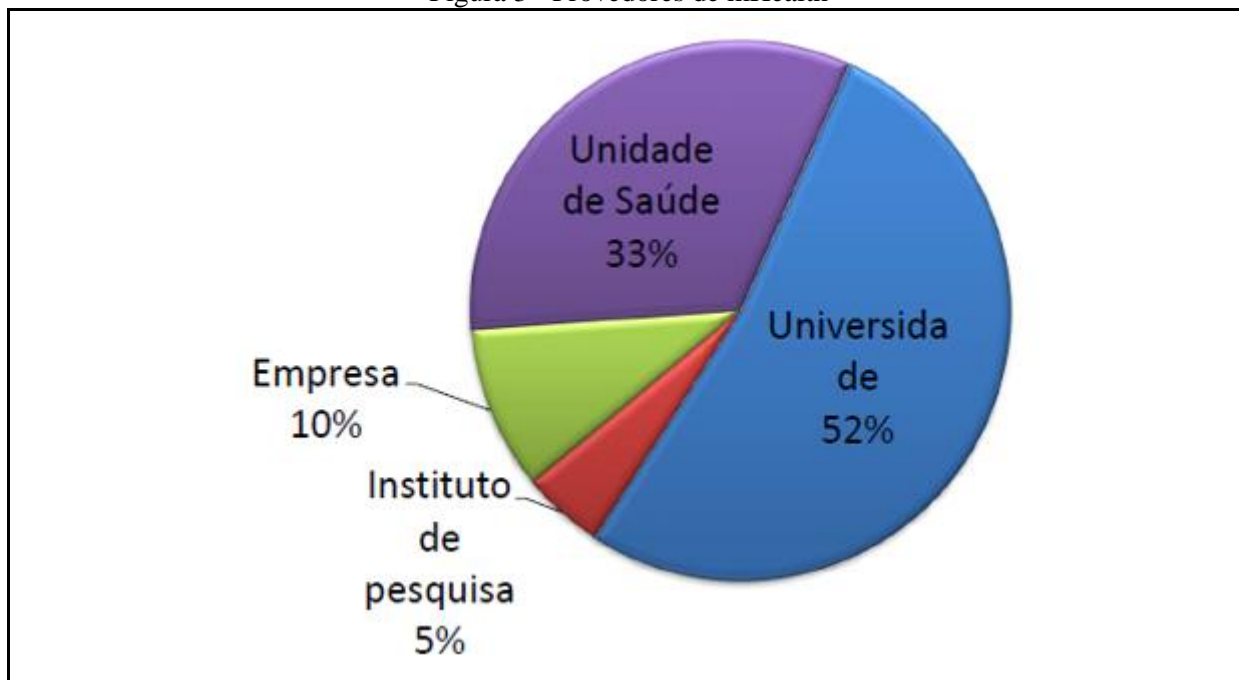
Figura 2 - Iniciativas de mHealth no Brasil



Fonte: adaptado de Iwaya et al. (2013).

Esta mesma pesquisa identificou que 52% dos projetos são oriundos de Universidades e apenas 10% desenvolvidos por empresas, como se observa na Figura 3.

Figura 3 - Provedores de mHealth

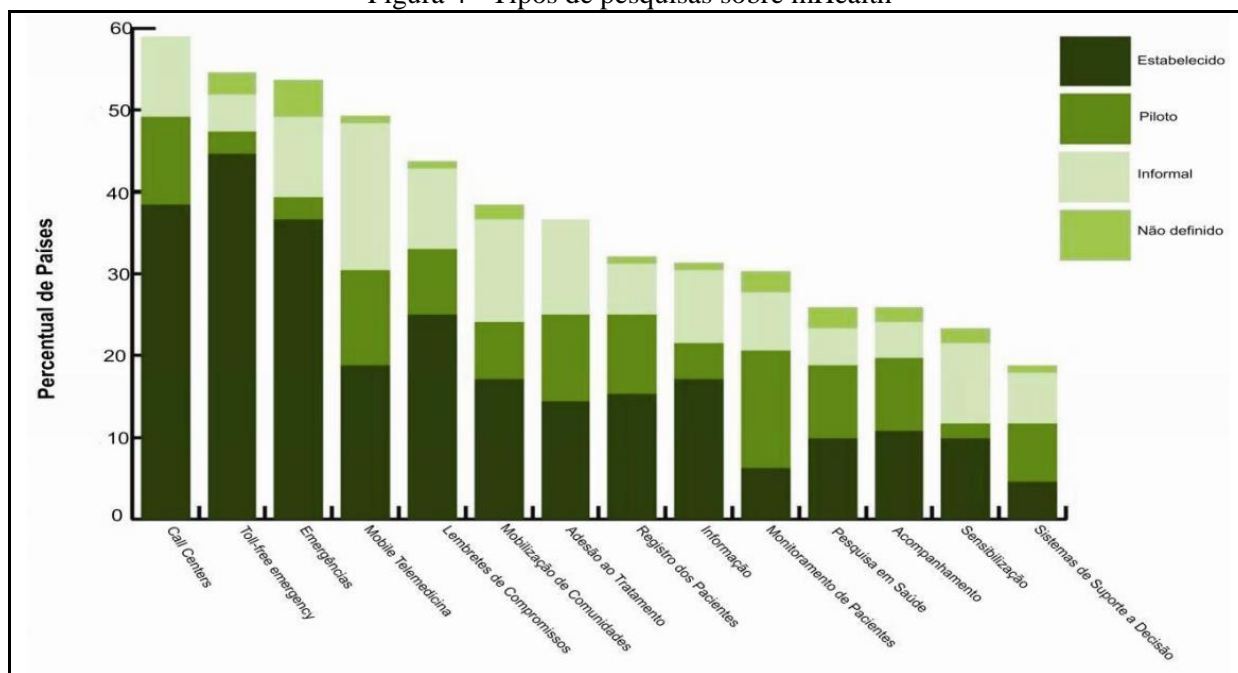


Fonte: adaptado de Iwaya et al. (2013).

De acordo com Tan (2014), o número de pacientes com diabetes está crescendo globalmente, o que acarreta em aumento de custos para o tratamento. Neste contexto, a mudança no estilo de vida através da introdução de tecnologia que permita o automonitoramento (PACAUD et al., 2012) pode contribuir para a diminuição nesta tendência de crescimento da doença. Na proposta do autor, tanto as tecnologias de eHealth como de mHealth podem ser utilizadas para introduzir o conceito de automonitoramento.

Conforme Costa (2013), em 2011, a Organização Mundial de Saúde (OMS) realizou uma pesquisa sobre *mobile health* entre seus 112 países membros, onde verificou-se que 83% deles possuíam ao menos uma iniciativa em mHealth. Na pesquisa, foram apresentados dados que apontam os vários níveis de desenvolvimento de pesquisa em mHealth, indicando a preocupação recorrente entre os países em buscar a tecnologia para melhorar os serviços de saúde, como também levar atendimento para as localidades geograficamente distantes dos grandes centros. Na Figura 4 é apresentado a porcentagem de países membros da OMS que possuem pesquisas em alguma categoria de mHealth.

Figura 4 - Tipos de pesquisas sobre mHealth



Fonte: Costa (2013) apud World Health Organization (2011).

2.4 TRABALHOS CORRELATOS

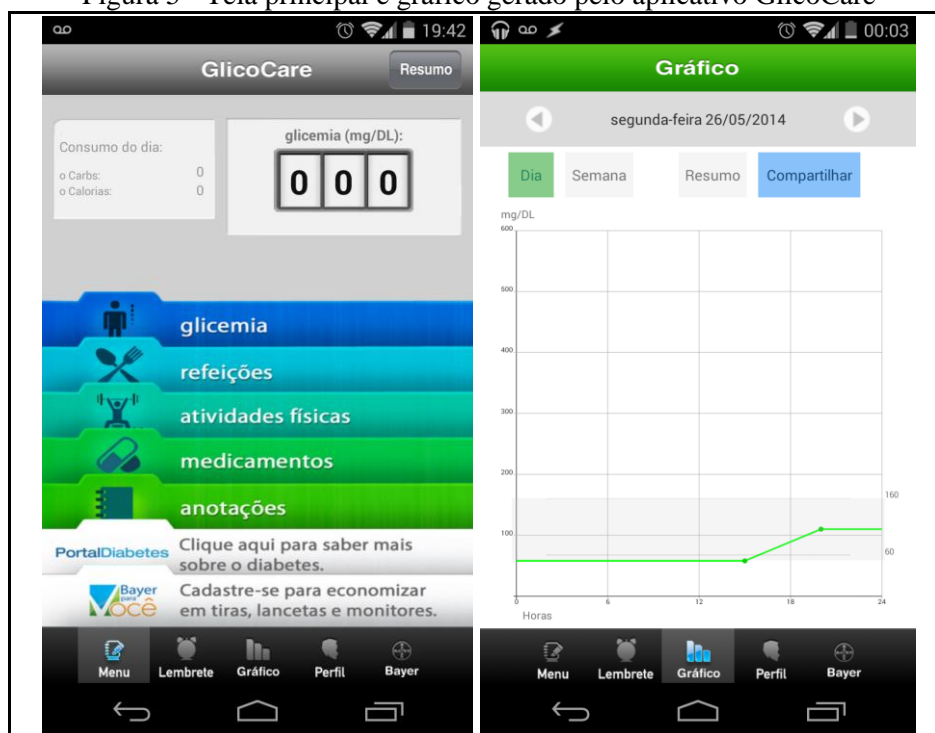
Pode-se citar como trabalhos correlatos os sistemas GlicoCare (BAYER AG, 2013), OnTrack (MEDIVO, 2014) e DiabetesControl (WIDÉN, 2010). Os três sistemas foram desenvolvidos com o intuito de auxiliar o diabético no seu tratamento através da manutenção de informações obtidas no seu dia-a-dia.

2.4.1 GLICOCARE

O aplicativo GlicoCare (Figura 5) foi desenvolvido pela empresa Bayer, uma multinacional alemã focada na produção de químicos e farmacêuticos. Disponível para download nas plataformas Android e iOS, tem como principal objetivo auxiliar o diabético a manter um diário de seu tratamento. Além do gerenciamento do histórico de glicemias, o aplicativo também permite que o usuário gere seu histórico de refeições, exercícios físicos e medicamentos, possibilitando ainda o cadastramento de lembretes (BAYER AG, 2013).

O aplicativo também possibilita a geração e compartilhamento de gráficos referentes aos níveis de glicemia diários ou semanais (Figura 5). Além disso, exibe dicas sobre o tratamento da diabetes para o usuário, caso o mesmo deseje.

Figura 5 - Tela principal e gráfico gerado pelo aplicativo GlicoCare

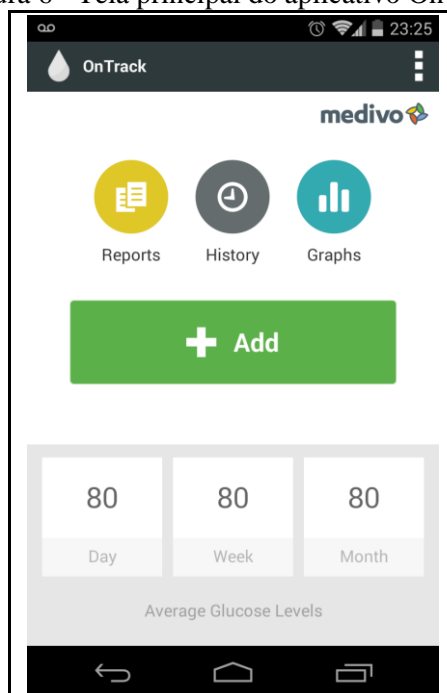


A navegação no aplicativo é feita através do menu inferior e dos atalhos para os principais cadastros que estão disponíveis na tela inicial. Nela, também são visíveis os carboidratos e calorias ingeridos no dia, além da última glicemia cadastrada.

2.4.2 ONTRACK

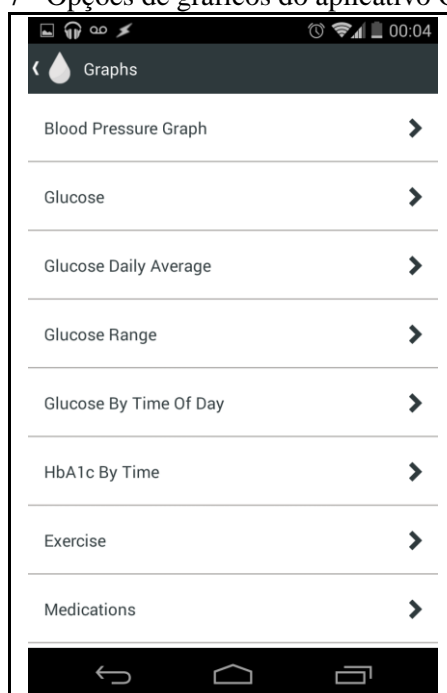
O aplicativo OnTrack (Figura 6), adquirido pela empresa Medivo em 2013, é um *software* cujo principal objetivo também é fornecer ao usuário a possibilidade de manter um diário de seu tratamento. Disponível apenas em inglês e para a plataforma Android, ele permite que o usuário gere suas informações dentro de uma gama de categorias pré-cadastradas (glicemias, medicamentos, pressão sanguínea, entre outras), possibilitando também a adição de novas categorias (MEDIVO, 2014).

Figura 6 - Tela principal do aplicativo OnTrack



O OnTrack oferece ao usuário a visualização de gráficos de todos os dados pertencentes às categorias pré-cadastradas (Figura 7). Além disso, é possível exportar todo o conteúdo da sua base de dados nos formatos CSV, XML e HTML, podendo-se filtrar o período desejado. Na inserção de novos registros, o OnTrack permite que o usuário cadastre diferentes informações (glicemia, pressão arterial, etc.) de uma só vez, sem precisar abrir mais de uma tela de cadastro.

Figura 7 - Opções de gráficos do aplicativo OnTrack



A navegação entre as principais funcionalidades do aplicativo é feita através dos atalhos presentes na tela inicial. Nela também são exibidas as médias diárias, semanais e mensais das glicemias cadastradas.

2.4.3 DIABETESCONTROL

O aplicativo DiabetesControl (Figura 8) foi desenvolvido como tese de mestrado por Widén (2010), sendo compatível com as versões 1.5 em diante da plataforma Android. Seu foco é no suporte de inserção de glicemias através de dispositivos *bluetooth* compatíveis, possibilitando também, manualmente, o gerenciamento de glicemias, medicamentos, exercícios físicos e refeições.

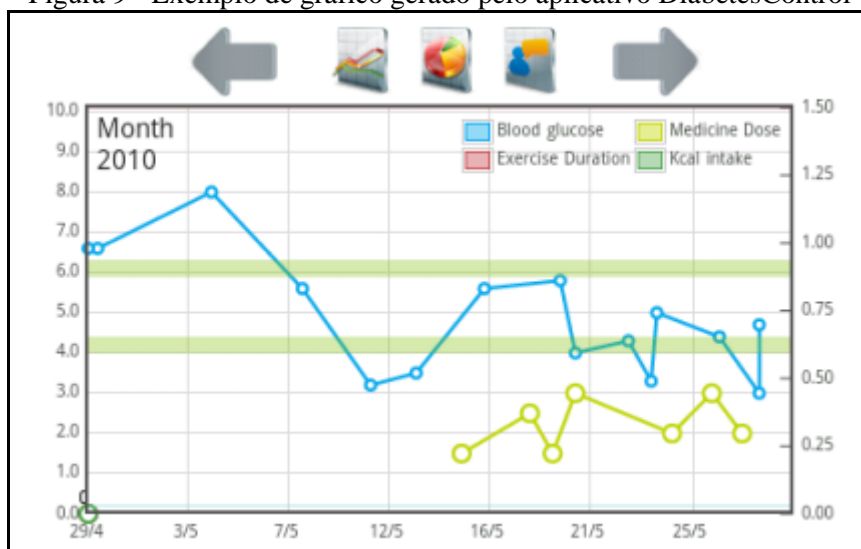
Figura 8 - Tela principal do aplicativo DiabetesControl



Fonte: Widén (2010).

O aplicativo também permite que o usuário configure lembretes para suas atividades diárias relacionadas ao tratamento da diabetes e exporte seus dados em formato XML. Além disso, o aplicativo é capaz de gerar gráficos de linha e de pizza com base nos registros do usuário (Figura 9). Esses gráficos também podem ser enviados por *e-mail* através da própria aplicação.

Figura 9 - Exemplo de gráfico gerado pelo aplicativo DiabetesControl



Fonte: Widén (2010).

A navegação é feita exclusivamente pela tela inicial, onde estão dispostos os atalhos para todas as funcionalidades do sistema. Os atalhos são compostos apenas por imagens, sem descrição.

3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo são descritos os aspectos técnicos do desenvolvimento do sistema, compostos pelo levantamento de informações e pela sua especificação, que contém seus requisitos funcionais e não funcionais, o diagrama de casos de uso, diagramas de atividades e diagramas de classes. Na sequência, apresenta-se a implementação com as técnicas e ferramentas utilizadas e a operacionalidade do sistema, e, por fim, os resultados obtidos e discussões pertinentes.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O aplicativo GluControl foi desenvolvido com o propósito de oferecer aos diabéticos uma maneira de manter um diário do seu tratamento, possibilitando o acompanhamento por um endocrinologista, configuração de rotina e geração de gráficos. Dentre as diversas informações que um diabético pode registrar uma ou mais vezes ao dia para auxiliar no seu tratamento, o sistema desenvolvido foca no essencial, os níveis de glicose no sangue (glicemias) e o uso de medicamentos.

Existem dois tipos de usuário na aplicação, o paciente e o endocrinologista, e cada um tem acesso a diferentes interfaces e funcionalidades. O usuário do tipo paciente pode cadastrar suas glicemias, seus medicamentos e a utilização dos mesmos. Ademais, pode configurar itens de rotina para lembrá-lo dos horários em que deve verificar sua glicemia ou tomar algum medicamento, configurar um endocrinologista para acompanhá-lo e definir um intervalo ideal para suas glicemias. O usuário do tipo endocrinologista pode visualizar as informações e históricos dos seus pacientes, no entanto, não pode editá-los. Tanto o paciente quando o endocrinologista tem acesso aos gráficos de performance do paciente e podem trocar mensagens entre si, através do aplicativo.

Como o aplicativo possui uma base de dados local que é sincronizada com a base de dados presente no servidor, o paciente pode utilizar as principais funcionalidades do aplicativo mesmo sem acesso à internet. O endocrinologista também pode, após uma sincronia, visualizar por tempo indeterminado as informações de seus pacientes sem a necessidade de estar conectado, porém, logicamente, os dados podem se tornar obsoletos.

3.2 ESPECIFICAÇÃO

Nesta seção são apresentados os requisitos funcionais e não funcionais do sistema, o diagrama de casos de uso, diagramas de atividades e diagramas de classes. Para a criação do diagrama de casos de uso, diagramas de atividades e diagramas de classes foi utilizada a ferramenta Enterprise Architect (EA).

3.2.1 REQUISITOS FUNCIONAIS

Nesta subseção são apresentadas as principais funcionalidades do sistema. O Quadro 1 apresenta os requisitos funcionais definidos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Quadro 1 - Requisitos Funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir ao visitante se cadastrar no sistema.	UC01
RF02: O sistema deverá permitir ao usuário efetuar o <i>login</i> .	UC02
RF03: O sistema deverá permitir ao usuário efetuar o <i>logoff</i> .	UC03
RF04: O sistema deverá permitir ao paciente configurar um endocrinologista para acompanhar o seu tratamento.	UC04
RF05: O sistema deverá permitir ao paciente manter seu histórico de glicemias.	UC05
RF06: O sistema deverá permitir ao paciente manter seu histórico de uso de medicamentos.	UC06
RF07: O sistema deverá permitir ao paciente configurar itens de rotina.	UC07
RF08: O sistema deverá permitir ao paciente a parametrização do intervalo de glicemia desejado para suas medidas.	UC08
RF09: O sistema deverá permitir ao paciente e ao endocrinologista conversarem entre si por <i>chat</i> com histórico.	UC09
RF10: O sistema deverá permitir a visualização de gráficos de performance do paciente.	UC10
RF11: O sistema deverá permitir a geração de <i>e-mails</i> contendo os	UC11

gráficos de performance.	
RF12: O sistema deverá permitir ao endocrinologista visualizar a lista de seus pacientes.	UC12
RF13: O sistema deverá permitir ao endocrinologista visualizar as informações de um paciente.	UC13
RF14: O sistema deverá permitir ao endocrinologista a remoção de um paciente da sua lista de acompanhamento.	UC14
RF15: O sistema deverá permitir ao endocrinologista visualizar os históricos de seus pacientes.	UC15
RF16: O sistema deverá permitir ao endocrinologista visualizar a rotina de seus pacientes.	UC16

3.2.2 REQUISITOS NÃO-FUNCIONAIS

No Quadro 2 são apresentados os requisitos não funcionais definidos para o sistema.

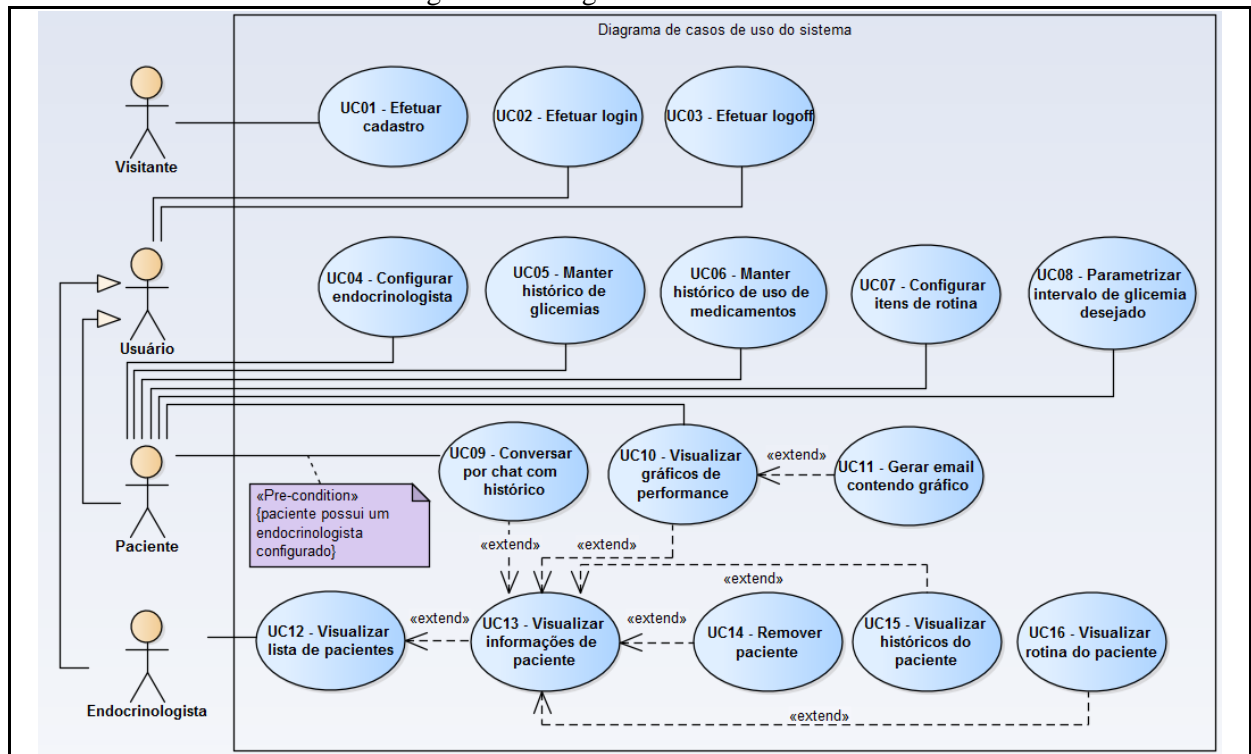
Quadro 2 - Requisitos Não Funcionais

Requisitos Não Funcionais
RNF01: O aplicativo deverá ser desenvolvido em Java, para a plataforma Android.
RNF02: O aplicativo deverá utilizar o banco de dados SQLite.
RNF03: O servidor deverá utilizar o banco de dados MySQL.

3.2.3 DIAGRAMA DE CASOS DE USO

Nesta subseção é apresentado, através da Figura 10, o diagrama de casos de uso da aplicação, sendo que o detalhamento de cada caso pode ser encontrado a partir do Apêndice A. O ator *Visitante* é todo aquele que não possui cadastro no sistema, portanto não tem acesso aos seus principais recursos e pode apenas efetuar o cadastro (UC01). Após cadastrar-se, o *Visitante* se torna um *Usuário*, podendo ser do tipo *Paciente* ou *Endocrinologista* e ganhando acesso a diferentes funcionalidades.

Figura 10 - Diagrama de Casos de Uso



O Usuário, por possuir cadastro, pode efetuar *login* (UC02) ou *logout* (UC03) no aplicativo.

O Paciente pode configurar um endocrinologista (UC04), manter seus históricos de glicemias e medicamentos (UC05 e UC06), configurar itens de rotina (UC07), parametrizar o intervalo desejado para suas glicemias (UC08), conversar com seu endocrinologista (UC09), visualizar seus gráficos de performance (UC10) e enviá-los por *e-mail* (UC11).

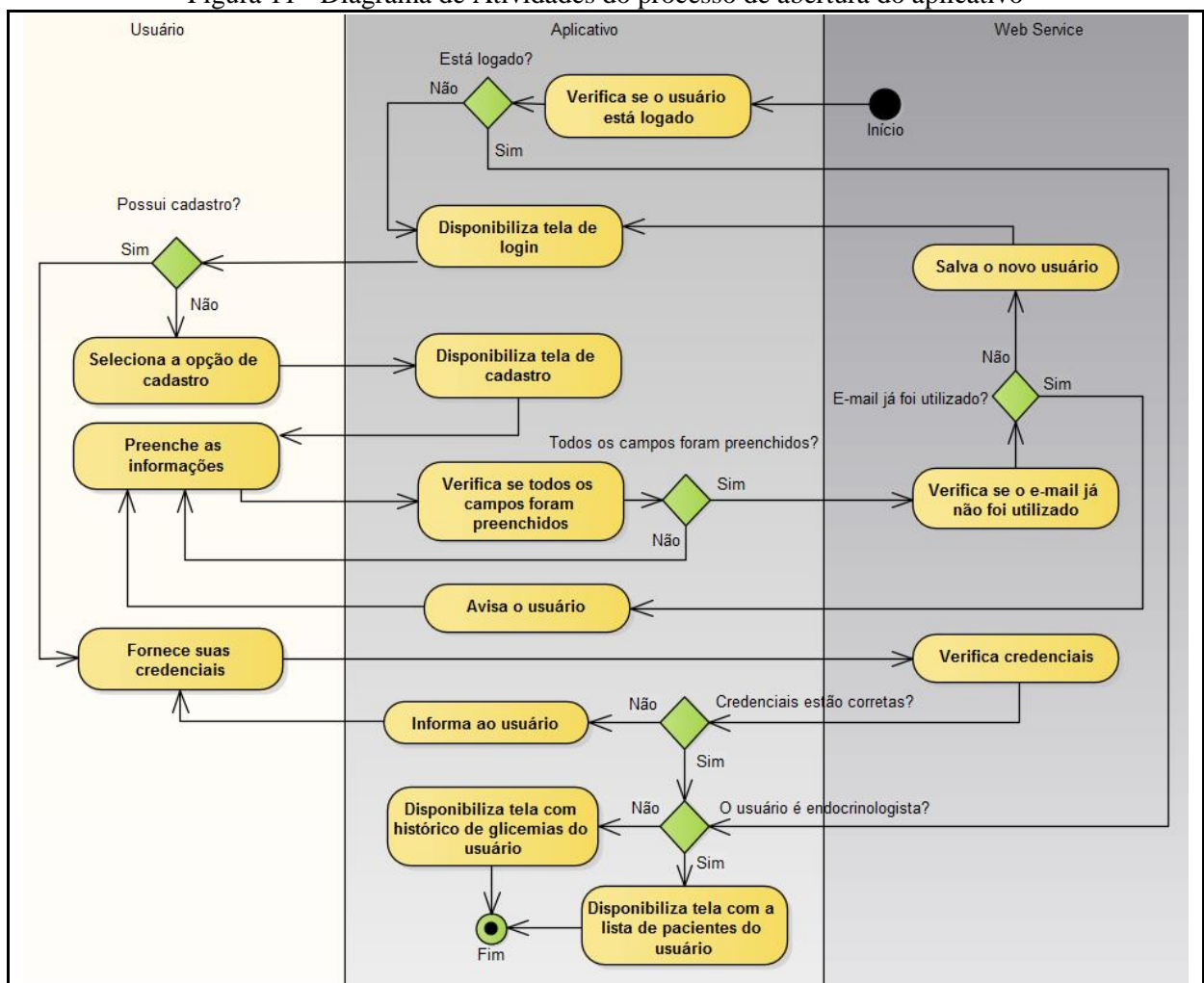
O Endocrinologista pode visualizar sua lista de pacientes (UC12), onde pode selecionar um paciente para maiores informações (UC13) e ter acesso aos seus históricos (UC15), rotina (UC16), *chat* (UC09) e gráficos de performance (UC10), que também podem ser enviados por *e-mail* (UC11). Além disso, o Endocrinologista também pode remover um paciente da sua lista de acompanhamento (UC14).

3.2.4 DIAGRAMAS DE ATIVIDADES

Nesta subseção apresenta-se os diagramas de atividades necessários para entendimento das principais funcionalidades do sistema. Na Figura 11 estão representadas as atividades executadas pelo usuário, pelo aplicativo e pelo *web service* no processo de abertura do

aplicativo. Inicialmente, o aplicativo verifica se o usuário já está logado no sistema. Caso não esteja, o usuário pode fornecer suas credenciais para *login* ou cadastrar-se. Ao cadastrar-se, o usuário fornece suas informações e o sistema verifica junto ao *web service* se o *e-mail* fornecido já não foi utilizado. Caso todas as informações estejam devidamente preenchidas, o usuário é cadastrado com sucesso e pode efetuar *login* no aplicativo. Ao efetuar *login*, o usuário fornece suas credenciais, que são então verificadas junto ao *web service*. Se as credenciais estiverem incorretas o aplicativo exibe um aviso para o usuário, caso contrário o aplicativo disponibiliza a tela inicial apropriada para o tipo de usuário que efetuou o *login*.

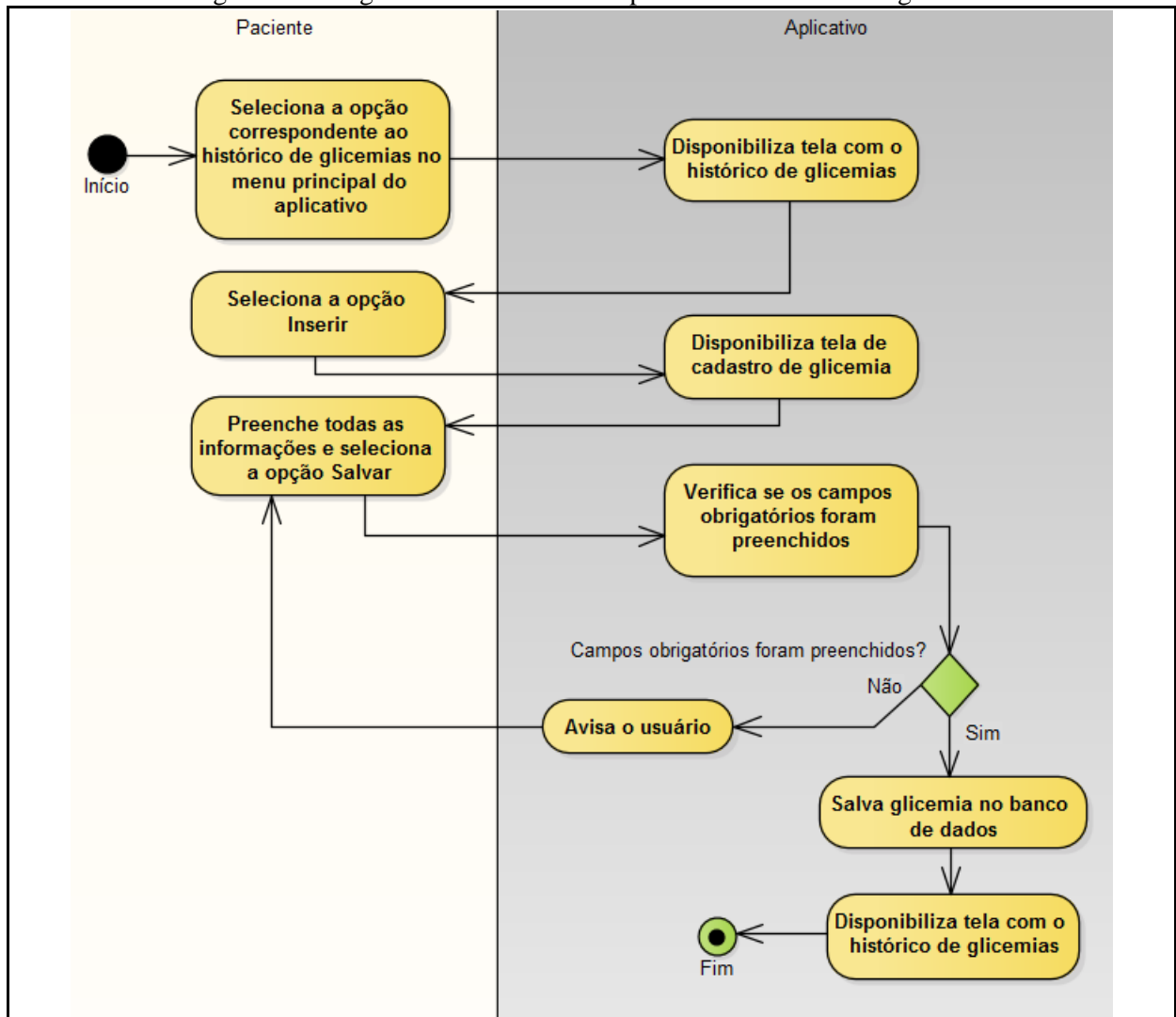
Figura 11 - Diagrama de Atividades do processo de abertura do aplicativo



Na Figura 12 estão representadas as atividades executadas pelo paciente e pelo aplicativo no processo de cadastramento de uma nova glicemia. A partir da tela principal, o usuário abre o menu e seleciona a opção referente ao histórico de glicemias, onde seleciona a opção de inserção. Na tela de cadastro, o paciente preenche todas as informações sobre a glicemia obtida e seleciona a opção de salvar. Caso algum campo obrigatório não tenha sido

preenchido, o aplicativo alerta o paciente e permanece na tela de cadastro. Caso todas as informações tenham sido devidamente fornecidas, o aplicativo salva a glicemia no banco de dados e retorna ao histórico. Este processo é similar para os demais cadastros disponíveis ao paciente.

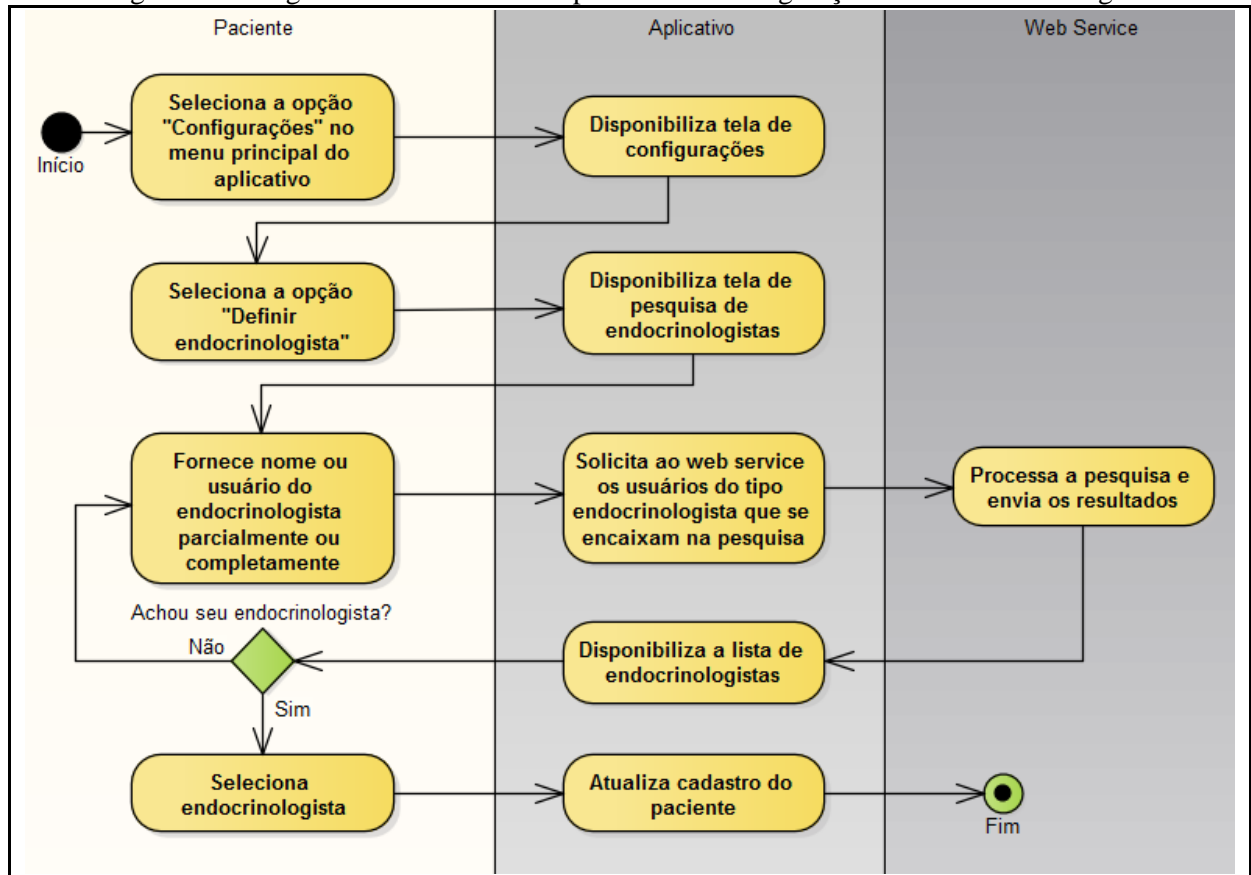
Figura 12 - Diagrama de Atividades do processo de cadastro de glicemia



Na Figura 13 estão representadas as atividades executadas pelo paciente, pelo aplicativo e pelo *web service* no processo de configuração de um endocrinologista. A partir da tela principal, o paciente abre o menu e seleciona a opção referente às configurações. Na tela de configurações, escolhe a opção “Definir endocrinologista”, e, na tela que se abre, fornece o *e-mail* ou o nome do endocrinologista para que seja efetuada uma pesquisa na base de dados do servidor. O *web service* retorna os usuários do tipo endocrinologista que satisfazem à pesquisa solicitada, e os resultados são então exibidos para o paciente em forma de lista, contendo seus nomes e *e-mails*. Caso o paciente encontre seu endocrinologista, pode

selecioná-lo e ter seu cadastro atualizado. Caso o endocrinologista procurado não esteja nos resultados, assume-se que o paciente não forneceu os dados corretos, e o mesmo pode refazer a pesquisa.

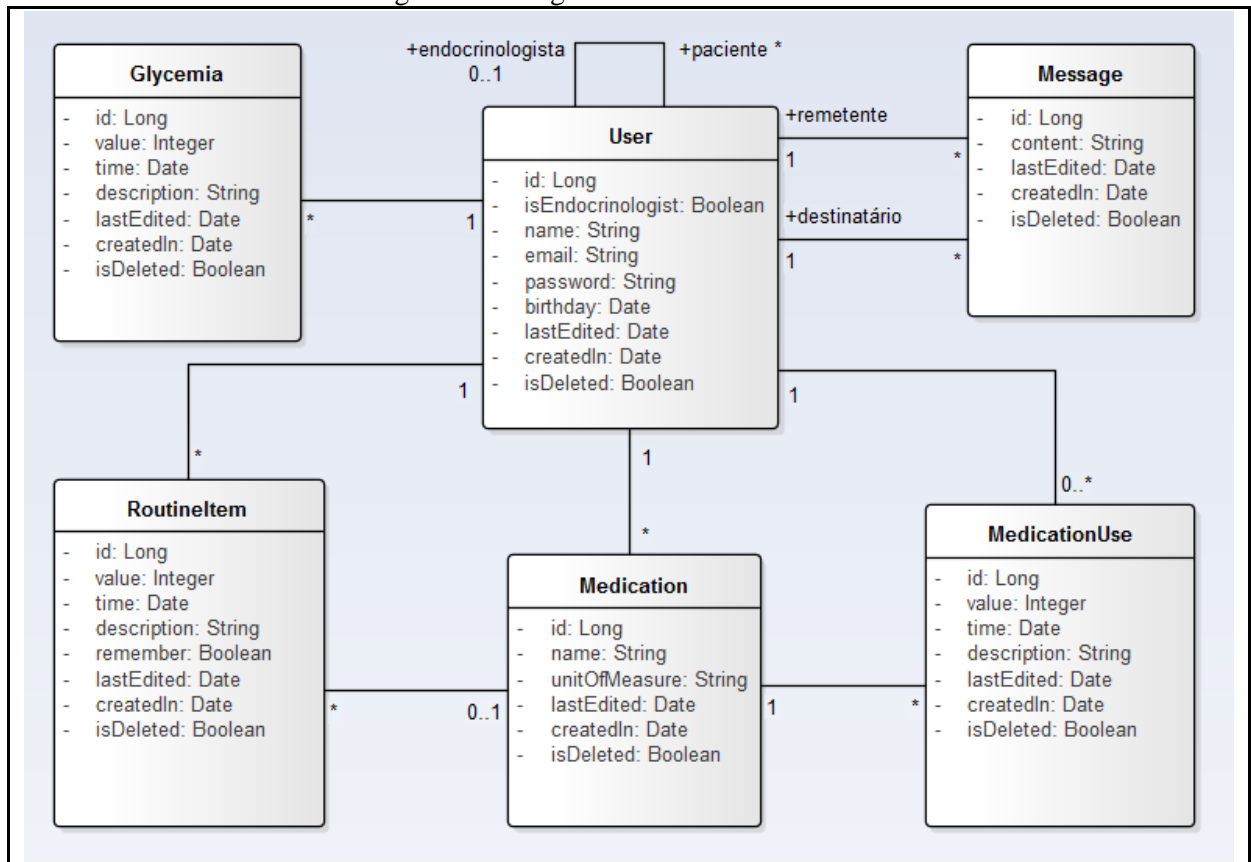
Figura 13 - Diagrama de Atividades do processo de configuração de um endocrinologista



3.2.5 DIAGRAMAS DE CLASSES

Nesta seção são apresentados os diagramas de classes do aplicativo e do servidor desenvolvidos. Os Modelos de Entidade e Relacionamento encontram-se no Apêndice B. A Figura 14 apresenta o diagrama das classes presentes no servidor do sistema.

Figura 14 - Diagrama de classes do servidor

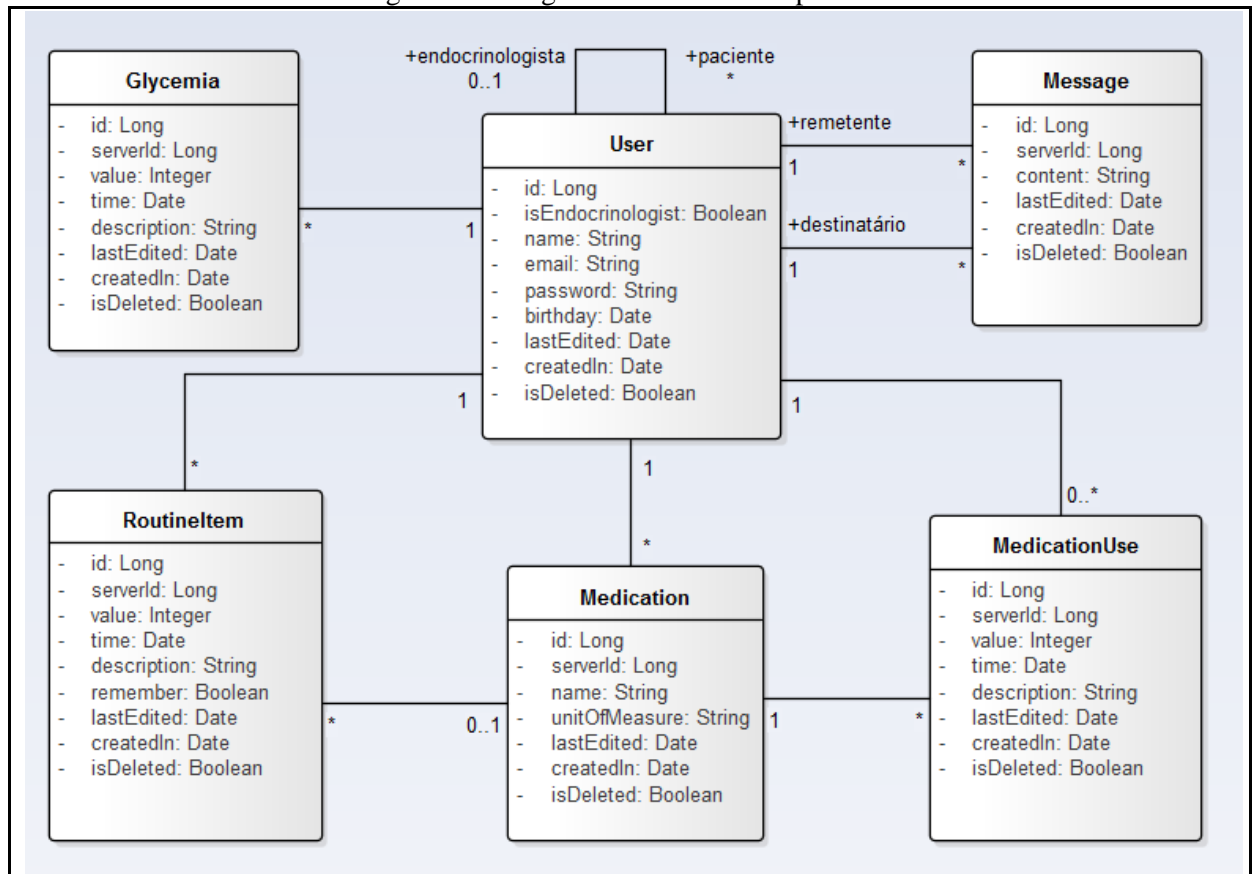


A seguir é apresentada uma breve descrição de cada classe desenvolvida:

- Glycemia**: Classe referente às glicemias dos usuários;
- Medication**: Classe referente aos medicamentos dos usuários;
- MedicationUse**: Classe referente aos usos de medicamentos dos usuários;
- Message**: Classe referente às mensagens enviadas pelos usuários;
- RoutineEntry**: Classe referente aos itens de rotina dos usuários;
- User**: Classe referente às informações dos usuários.

Na Figura 15 apresenta-se o diagrama das classes presentes no aplicativo desenvolvido.

Figura 15 - Diagrama de classes do aplicativo



Tanto o servidor quanto o aplicativo possuem as mesmas classes, no entanto, todas as classes presentes no banco de dados do aplicativo, com exceção de `User`, possuem o atributo `serverId`, utilizado para fins de sincronia.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as ferramentas, técnicas utilizadas e a operacionalidade da implementação.

3.3.1 FERRAMENTAS E TÉCNICAS UTILIZADAS

Nessa seção são apresentadas as ferramentas utilizadas, seguidas pela arquitetura do sistema e demais técnicas, divididas em Persistência, Interface, Comunicação com o servidor e Sincronização de dados para facilitar o entendimento.

3.3.1.1 Ferramentas

O aplicativo GluControl foi desenvolvido na linguagem de programação Java, utilizando o *Integrated Development Environment (IDE)* Eclipse em conjunto com o *plug-in Android Developer Tools (ADT)*.

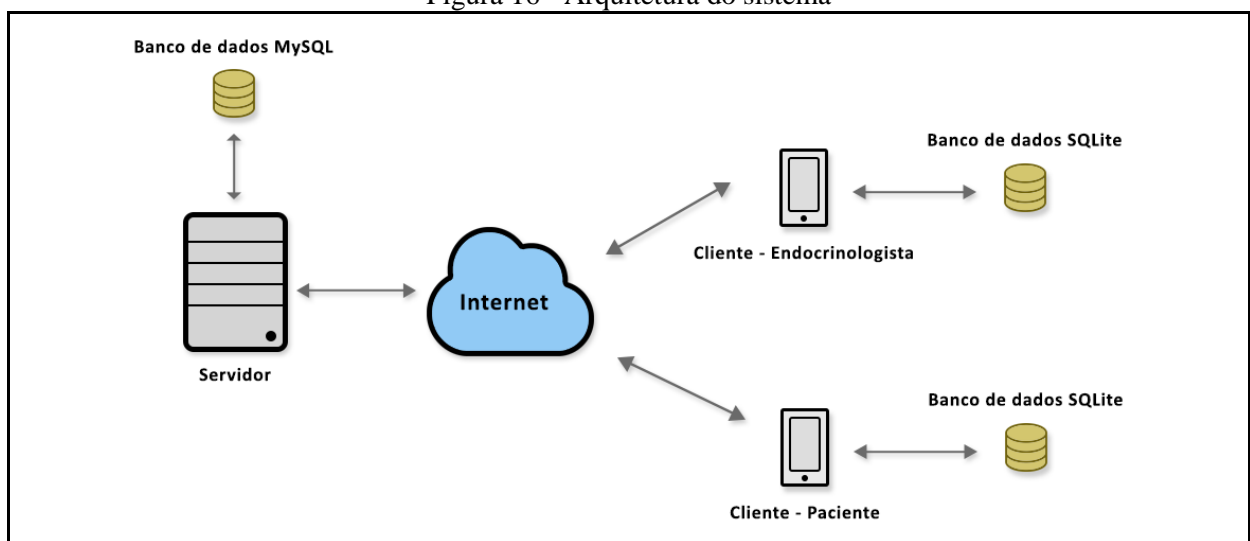
A persistência dos dados na aplicação fica a cargo do sistema de gerenciamento de banco de dados (SGBD) SQLite, nativo da plataforma Android. Para serialização e desserialização de objetos de/para *JavaScript Object Notation (JSON)* foi utilizada a biblioteca de código aberto Gson. Para geração de gráficos e manipulação de datas e horários foram utilizadas as bibliotecas MPAndroidChart e Joda Time, respectivamente, ambas também de código aberto.

O *web service* também foi desenvolvido com o IDE Eclipse, na linguagem de programação Java e utilizando o *framework* Jersey. Para persistência dos dados foi utilizado o SGBD MySQL.

3.3.1.2 Arquitetura do Sistema

A arquitetura do sistema desenvolvido consiste em um servidor, que armazena as informações de todos os usuários, e clientes, que possuem uma base de dados própria que sincroniza as informações pertinentes ao usuário utilizador com as informações presentes no servidor. Isso permite que o usuário autorizado consiga registrar informações mesmo sem acesso à internet, com tais informações sendo sincronizadas quando possível. Na Figura 16 está representada esta arquitetura.

Figura 16 - Arquitetura do sistema



3.3.1.3 Persistência

O armazenamento dos dados no aplicativo foi desenvolvido da seguinte forma: para cada entidade existe uma classe modelo, uma classe com todas as informações referentes à sua tabela no banco de dados e uma classe responsável pela construção dos comandos e consultas em *Structured Query Language* (SQL). Na Figura 17 é apresentada a classe modelo utilizada para os usuários.

Figura 17 - Classe modelo da entidade User

```

7 public class User {
8     @Expose
9     private Long id;
10    @Expose
11    private User endocrinologist;
12    @Expose
13    private Boolean isEndocrinologist;
14    @Expose
15    private String name;
16    @Expose
17    private String email;
18    @Expose
19    private String password;
20    @Expose
21    private Date birthday;
22    @Expose
23    private Date lastEdited;
24    @Expose
25    private Date createdIn;
26    @Expose
27    private Boolean isDeleted;
28
29    public User() {
30    }
31
32    public User(Long id, User endocrinologist, Boolean isEndocrinologist, String name, String email, String password,
33                Date birthday, Date lastEdited, Date createdIn, Boolean isDeleted) {
34        this.id = id;
35        this.endocrinologist = endocrinologist;
36        this.isEndocrinologist = isEndocrinologist;
37        this.name = name;
38        this.email = email;
39        this.password = password;
40        this.birthday = birthday;
41        this.lastEdited = lastEdited;
42        this.createdIn = createdIn;
43        this.isDeleted = isDeleted;
44    }

```

As classes modelo não possuem nenhum método além dos métodos de encapsulamento, pois sua função é apenas fornecer uma camada de abstração entre a aplicação e o banco de dados, permitindo que as telas de cadastro e edição trabalhem com objetos e não diretamente com registros.

Na Figura 18 é apresentada parcialmente a classe responsável pela inserção, alteração, exclusão e consulta de usuários. Nessa classe também ocorre a relação dos atributos da classe modelo com as colunas da entidade no banco de dados.

Figura 18 - Classe responsável pelas operações no banco de dados da entidade User

```

16 public class DaoUser extends Dao {
17
18     public DaoUser(Context context) {
19         super(context);
20     }
21
22     public long insert(User user) {
23         long time = System.currentTimeMillis();
24         ContentValues values = new ContentValues();
25         values.put(UserTable.COLUMN_ID, user.getId());
26         values.put(UserTable.COLUMN_ENDOCRINOLOGIST, user.getEndocrinologist() != null ? user.getEndocrinologist().getId() : null);
27         values.put(UserTable.COLUMN_IS_ENDOCRINOLOGIST, user.getIsEndocrinologist());
28         values.put(UserTable.COLUMN_NAME, user.getName());
29         values.put(UserTable.COLUMN_EMAIL, user.getEmail());
30         values.put(UserTable.COLUMN_PASSWORD, user.getPassword());
31         values.put(UserTable.COLUMN_BIRTHDAY, user.getBirthday().getTime());
32         values.put(UserTable.COLUMN_LAST_EDITED, time);
33         values.put(UserTable.COLUMN_CREATED_IN, time);
34         values.put(UserTable.COLUMN_IS_DELETED, false);
35         return insert(UserTable.TABLE_NAME, values);
36     }
37
38     public long insertFromServer(User user) {}
39
40     public int update(User user) {}
41
42     public int updateFromServer(User user) {}
43
44     public int delete(User user) {}
45
46     public User user(Cursor c) {}
47
48     public User userForAdapter(Cursor c) {}
49
50     public User get(long id) {}
51
52     public Cursor getAll() {}
53
54     public Cursor getPacients(long id) {}
55 }

```

Para facilitar a manutenção e leitura do código, foram utilizadas classes para armazenar os metadados de cada tabela. Essas classes contêm constantes com nome da tabela, os nomes de cada coluna e o código para criação da tabela, como apresentado na Figura 19.

Figura 19 - Classe onde são definidas as propriedades da tabela para a entidade User

```

11 public class UserTable {
12
13     public static final String TABLE_NAME = "User";
14
15     public static final String COLUMN_ID = "id";
16     public static final String COLUMN_NAME = "name";
17     public static final String COLUMN_EMAIL = "email";
18     public static final String COLUMN_PASSWORD = "password";
19     public static final String COLUMN_BIRTHDAY = "brithday";
20     public static final String COLUMN_IS_ENDOCRINOLOGIST = "is_endocrinologist";
21     public static final String COLUMN_ENDOCRINOLOGIST = "endocrinologist";
22     public static final String COLUMN_LAST_EDITED = "last_edited";
23     public static final String COLUMN_CREATED_IN = "created_in";
24     public static final String COLUMN_IS_DELETED = "is_deleted";
25
26     private UserTable() {
27     }
28
29     public static final String[] COLUMN_ARRAY = { COLUMN_ID, COLUMN_NAME, COLUMN_EMAIL, COLUMN_PASSWORD, COLUMN_BIRTHDAY,
30         COLUMN_IS_ENDOCRINOLOGIST, COLUMN_ENDOCRINOLOGIST, COLUMN_LAST_EDITED, COLUMN_CREATED_IN, COLUMN_IS_DELETED};
31
32     public static final String DEFAULT_SORT_ORDER = COLUMN_NAME + " DESC";
33
34     public static final String CREATE_TABLE =
35         "CREATE TABLE " + TABLE_NAME + " (" +
36         COLUMN_ID + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
37         COLUMN_NAME + DataType.TEXT + DefinitionsDatabase.COMMA_SEP +
38         COLUMN_EMAIL + DataType.TEXT + DefinitionsDatabase.COMMA_SEP +
39         COLUMN_PASSWORD + DataType.TEXT + DefinitionsDatabase.COMMA_SEP +
40         COLUMN_BIRTHDAY + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
41         COLUMN_IS_ENDOCRINOLOGIST + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
42         COLUMN_ENDOCRINOLOGIST + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
43         COLUMN_LAST_EDITED + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
44         COLUMN_CREATED_IN + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
45         COLUMN_IS_DELETED + DataType.INTEGER + DefinitionsDatabase.COMMA_SEP +
46         "PRIMARY KEY(" + COLUMN_ID + ")")";

```


Para a persistência dos dados no servidor foi utilizada a mesma metodologia, porém, os metadados das tabelas foram incluídos nas classes responsáveis pela manipulação dos dados.

3.3.1.4 Interface

Na plataforma Android, a interface é construída através de arquivos utilizando *eXtensible Markup Language* (XML). Os elementos definidos nestes arquivos são manipulados através de classes do tipo *Activity* ou *Fragment* (módulos de uma *Activity*). Na Figura 20 é apresentado parte do código XML responsável pela estrutura e aparência da tela de cadastro.

Figura 20 - Parte do arquivo XML referente à tela de cadastro

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:background="@color/my_window_secondary_background"
7      android:orientation="vertical" >
8
9      <include
10         android:id="@+id/toolbar"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         layout="@layout/toolbar" />
14
15     <ScrollView
16         android:layout_width="match_parent"
17         android:layout_height="match_parent"
18         android:layout_below="@id/toolbar" >
19
20         <LinearLayout style="@style/Form" >
21
22             <RelativeLayout style="@style/FormItemLine" >
23
24                 <ImageView
25                     style="@style/FormItemIcon"
26                     android:contentDescription="@string/hint_name"
27                     android:src="@drawable/ic_user" />
28
29                 <LinearLayout style="@style/FormItemLayout" >
30
31                     <TextView
32                         android:id="@+id/label_name"
33                         style="@style/FormItemLabel"
34                         android:labelFor="@+id/input_name"
35                         android:text="@string/hint_name" />
36
37                     <EditText
38                         android:id="@+id/input_name"
39                         style="@style/FormItemInput.Name" />
40                 </LinearLayout>
41             </RelativeLayout>

```

As *activities* são classes responsáveis pela exibição e manipulação dos elementos presentes nos arquivos XML. Na Figura 21 pode-se observar a classe referente à tela de cadastro de usuários. A associação entre a *activity* e o seu *layout* em XML ocorre através do método `setContentView()`, no momento em que a classe é criada.

Figura 21 - Classe responsável pela tela de cadastro

```

25 public class RegistrationActivity extends FragmentActivity implements OnClickListener, OnDateSetListener,
26     AsyncTaskCompleteListener<User> {
27     private final static String DATE_FORMAT = "dd/MM/yyyy";
28     private Calendar cal;
29     private EditText edtName, edtEmail, edtPassword, edtDate;
30     private CheckBox cbIsEndocrinologist;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_registration);
36
37         Toolbar mToolbar = (Toolbar) findViewById(R.id.toolbar);
38         setSupportActionBar(mToolbar);
39         getActionBar().setDisplayHomeAsUpEnabled(true);
40
41         cal = Calendar.getInstance();
42         getFields();
43         edtDate.setOnClickListener(this);
44     }
45
46     private void getFields() {
47         edtName = (EditText) findViewById(R.id.input_name);
48         edtEmail = (EditText) findViewById(R.id.input_email);
49         edtPassword = (EditText) findViewById(R.id.input_password);
50         edtDate = (EditText) findViewById(R.id.input_date);
51         cbIsEndocrinologist = (CheckBox) findViewById(R.id.input_is_endocrinologist);
52     }
53
54     @Override
55     public boolean onCreateOptionsMenu(Menu menu) {
56         getMenuInflater().inflate(R.menu.cancel, menu);
57         getMenuInflater().inflate(R.menu.save, menu);
58         return true;
59     }

```

Após a associação entre a *activity* e seu *layout*, os elementos da interface podem ser recuperados através do método `findViewById()`. Todas as *activities* precisam ser declaradas no arquivo `AndroidManifest.xml` do aplicativo, ou o Android não permitirá que as mesmas sejam instanciadas.

Figura 22 - Declaração da *activity* da tela de cadastro no arquivo `AndroidManifest.xml`

```

45     <activity
46         android:name="br.com.glucontrol.view.activities.RegistrationActivity"
47         android:label="@string/activity_title_registration"
48         android:parentActivityName="br.com.glucontrol.view.activities.LoginActivity"
49         android:screenOrientation="portrait" />

```

3.3.1.5 Comunicação com servidor

O *web service* da aplicação foi desenvolvido baseado na arquitetura *Representational State Transfer* (REST), utilizando o *framework* Jersey. Todos os recursos disponíveis foram divididos em classes nomeadas em referência à entidade a qual manipulam. Para consultas de informações consideradas não-sigilosas são utilizados métodos GET. Para consultas de

credenciais e manipulação de dados são utilizados métodos POST que consomem objetos em JSON. Na Figura 23 apresenta-se a classe responsável pelas requisições referentes à consulta e manipulação de glicemias.

Figura 23 - Classe do *web service* responsável pelas requisições referentes às glicemias

```

18 @Path("/glycemia")
19 @Produces(MediaType.APPLICATION_JSON)
20 public class GlycemiaResource {
21     @GET
22     @Path("/listAllFromUser/{userid}")
23     public ArrayList<Glycemia> listAllFromUser(@PathParam("userid") long userId) {
24         return GlycemiaDAO.getInstance().listAllFromUser(userId);
25     }
26
27     @GET
28     @Path("/{id}")
29     public Glycemia getGlycemia(@PathParam("id") long id) {
30         return GlycemiaDAO.getInstance().get(id);
31     }
32
33     @POST
34     @Path("/save")
35     @Consumes(MediaType.APPLICATION_JSON)
36     public Response saveGlycemia(String json) {
37         GlycemiaDAO dao = GlycemiaDAO.getInstance();
38         Glycemia obj;
39         try {
40             obj = JsonUtil.getDefaultMapper().readValue(json, Glycemia.class);
41             Long objId = dao.save(obj);
42             if (objId != null) {
43                 return Response.status(200).entity(dao.get(objId)).build();
44             } else {
45                 return Response.status(400).build();
46             }
47         } catch (Exception e) {
48             e.printStackTrace();
49             return Response.status(500).build();
50         }
51     }
52 }

```

As anotações acima da classe definem o caminho base dos métodos nela contidos e o tipo de resposta que eles produzem. As anotações acima dos métodos definem o tipo de requisição, o caminho utilizado para acessá-los e, no caso das requisições POST, o tipo de mensagem que consomem. Na classe apresentada, por exemplo, o método `saveGlycemia()` pode ser acessado remotamente através da *Uniform Resource Locator* (URL) `http://[ip_do_servidor:porta]/glucontrolrestful/glycemia/save`, onde `[ip_do_servidor:porta]` deve ser substituído pelo endereço IP do servidor juntamente com a porta disponível para conexão.

No aplicativo, processos que efetuam requisições *Hypertext Transfer Protocol* (HTTP) são executados em *threads* separadas que utilizam os métodos responsáveis pela troca de

mensagens com o *web service*. Na Figura 24 apresenta-se o código responsável pelas requisições POST.

Figura 24 - Código responsável pelas requisições HTTP através do método POST

```

83e public String[] postRequest(String local, String json) {
84     try {
85         Log.i(TAG, "Accessing web service at " + local);
86         Log.i(TAG, "POST Parameters: " + json);
87         URL url = new URL(local);
88         HttpURLConnection conn = (HttpURLConnection) url.openConnection();
89         conn.setDoOutput(true);
90         conn.setRequestMethod("POST");
91         conn.setRequestProperty("Content-Type", "application/json");
92         conn.setChunkedStreamingMode(0);
93         conn.setConnectTimeout(10000);
94         conn.connect();
95
96         // Write
97         OutputStream os = conn.getOutputStream();
98         BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os));
99         writer.write(json.toString());
100        writer.close();
101        os.close();
102
103        // Read
104        int responseCode = conn.getResponseCode();
105        String response = streamToString(conn.getInputStream());
106
107        conn.disconnect();
108
109        Log.i(TAG, "Response code: " + responseCode);
110        Log.i(TAG, "Response content: " + response);
111
112        return new String[] { String.valueOf(responseCode), response };
113    } catch (Exception e) {
114        Log.e(TAG, "Error accessing web service", e);
115    }
116    return null;
117 }

```

Quando a resposta esperada pelo *web service* está no formato JSON, é necessário converter o código recebido para as classes modelo que a aplicação utiliza. Na Figura 25 apresenta-se um dos métodos responsáveis por essa conversão.

Figura 25 - Método responsável pela conversão de código JSON em um único objeto

```

152e public <T> T buildObjectFromJson(Class<T> clazz, String json) {
153     try {
154         if (json != null) {
155             if (!json.equals("null") && !json.isEmpty()) {
156                 JsonParser parser = new JsonParser();
157                 JsonObject object = parser.parse(json).getAsJsonObject();
158                 return gson.fromJson(object, clazz);
159             }
160         }
161     } catch (Exception e) {
162         Log.e(TAG, "Error converting json to object", e);
163     }
164     return null;
165 }

```

3.3.1.6 Sincronização de dados

A sincronização dos dados do aplicativo com os dados presentes no servidor é efetuada com base em dois atributos contidos em todas as entidades: `serverId` e `lastEdited`. Na entidade `User` não é necessário o atributo `serverId` pois seu atributo `id` sempre será igual ao do servidor.

A lógica de sincronização é simples, todos os registros locais são comparados com suas contrapartes no servidor. Caso o registro não possua `serverId`, é necessário inseri-lo na base de dados do servidor. Caso o registro possua `serverId`, prevalece o que tiver sido editado mais recentemente. Caso existam registros presentes apenas no servidor, os mesmos são inseridos na base local. A Figura 26 mostra esta lógica implementada para a sincronização de glicemias.

Figura 26 - Código responsável pela sincronização de glicemias

```

56= private void syncGlycemias(long id) {
57     Class<Glycemia> clazz = Glycemia.class;
58     DaoGlycemia dao = new DaoGlycemia(context);
59     Cursor c = dao.getAll(id);
60
61     ArrayList<Glycemia> objectsServer = rest.getObjectsFromUser(id, Glycemia.class);
62
63     if (objectsServer != null) {
64         for (Glycemia i : objectsServer) {
65             Glycemia obj = dao.getByServerId(i.getServerId());
66             if (obj == null) {
67                 Log.i(TAG, "Inserting server glycemia id=" + i.getServerId() + " into database");
68                 dao.insertFromServer(i);
69             } else {
70                 DateTime dt1 = new DateTime(obj.getLastEdited().getTime());
71                 DateTime dt2 = new DateTime(i.getLastEdited().getTime());
72                 int dif = dt1.compareTo(dt2);
73                 if (dif < 0) {
74                     Log.i(TAG, "Updating local glycemia id=" + obj.getServerId() + " with server data. Local last edited: "
75                         + obj.getLastEdited() + ". Server last edited: " + i.getLastEdited() + ".");
76                     i.setId(obj.getId());
77                     dao.updateFromServer(i);
78                 } else if (dif > 0) {
79                     Log.i(TAG, "Updating server glycemia id=" + i.getServerId() + " with local data. Local last edited: "
80                         + obj.getLastEdited() + ". Server last edited: " + i.getLastEdited() + ".");
81                     rest.saveObject(obj, clazz);
82                 }
83             }
84         }
85     }
86
87     if (c != null) {
88         while (c.moveToNext()) {
89             Glycemia obj = dao.glycemiaForAdapter(c);
90             if (obj.getServerId() == null) {
91                 Log.i(TAG, "Inserting local glycemia into server");
92                 Glycemia t = rest.saveObject(obj, clazz);
93                 t.setId(obj.getId());
94                 dao.updateFromServer(t);
95             }
96         }
97     }
98 }

```

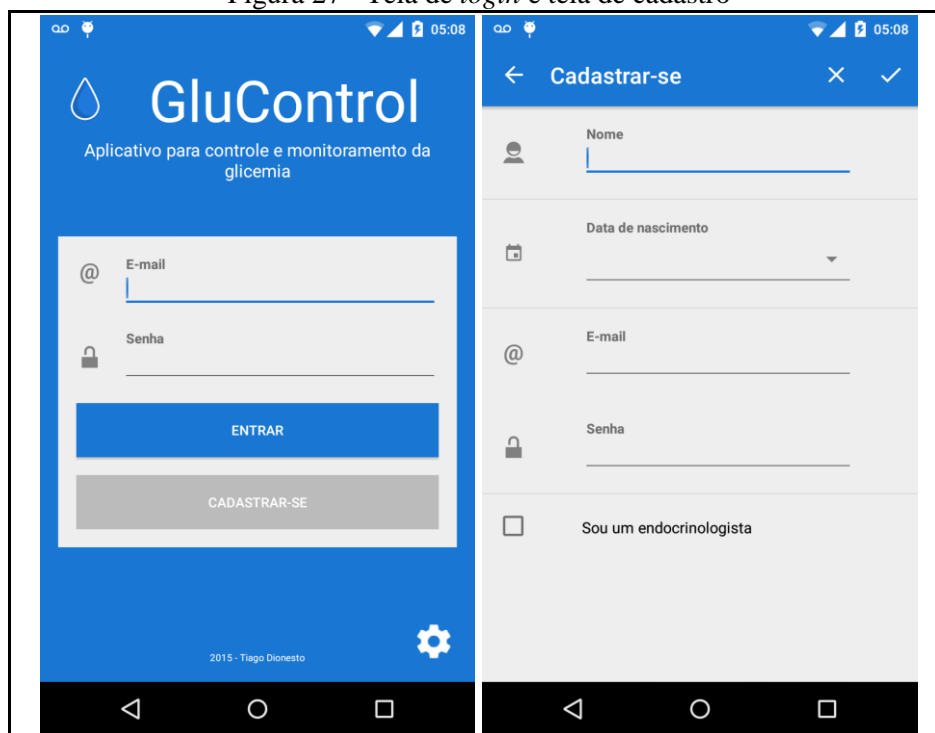
Para o protótipo, verificou-se que era mais apropriado iniciar o processo de sincronização de dados manualmente, portanto não foi desenvolvido um serviço responsável pela sincronia constante. O processo pode ser iniciado a partir do menu principal da aplicação, através da opção “Sincronizar”.

3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Esta seção contém a operacionalidade do sistema desenvolvido, explicada de acordo com o tipo de cadastro do usuário para melhor entendimento. Para o aplicativo móvel foi dado o nome GluControl.

Para utilizar o aplicativo, o usuário deve primeiramente efetuar seu cadastro, fornecendo informações básicas como nome, data de nascimento, *e-mail* e senha, além de informar se o mesmo é um endocrinologista ou apenas um usuário comum (paciente). Após o cadastro, o usuário deve efetuar o *login* no sistema, utilizando o *e-mail* e senha anteriormente fornecidos, obtendo acesso às demais funcionalidades. Na Figura 27 são apresentadas a tela de *login* e a tela de cadastro.

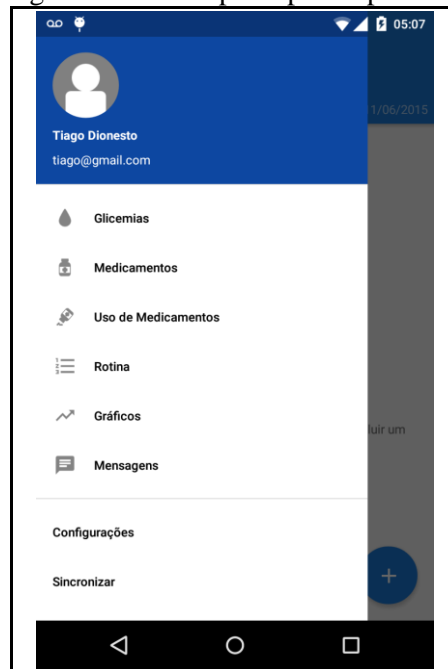
Figura 27 - Tela de *login* e tela de cadastro



3.3.2.1 Paciente

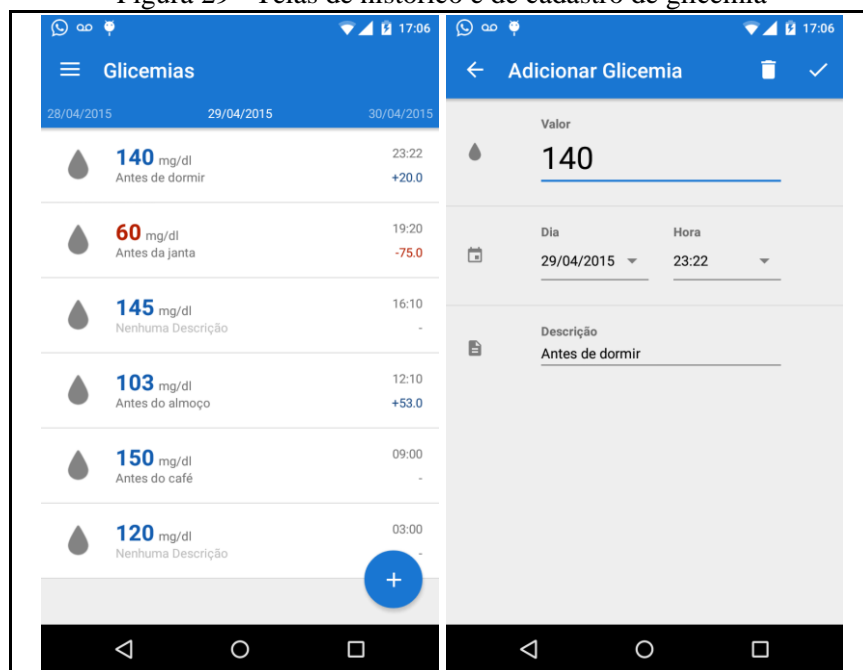
O paciente, após efetuar o *login*, tem acesso ao seu histórico de glicemias e de uso de medicamentos, ao cadastro de medicamentos, à sua rotina, aos gráficos de desempenho, ao chat (caso possua um endocrinologista configurado) e à janela de configurações. Todas essas opções são disponibilizadas no menu principal da aplicação, conforme demonstrado na Figura 28.

Figura 28 - Menu principal do paciente



No histórico de glicemias, o paciente pode visualizar as glicemias de determinado dia, incluir uma nova ou selecionar uma existente para editá-la ou excluí-la. As glicemias são exibidas em ordem cronológica, com as mais recentes no topo, e para cada uma é exibido seu valor, o horário em que foi obtida e a descrição fornecida pelo paciente. A Figura 29 apresenta as telas de histórico e de cadastro de glicemia.

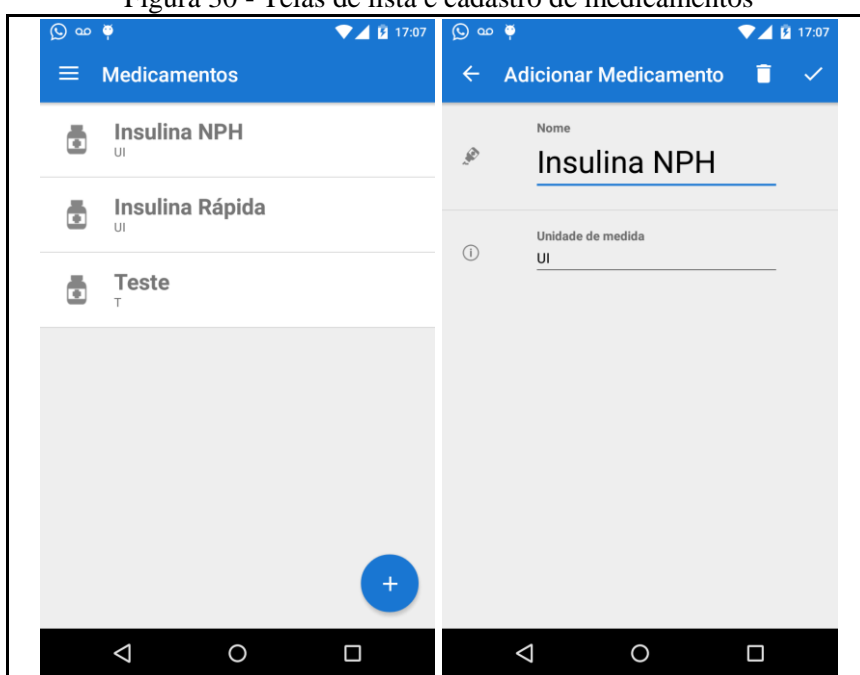
Figura 29 - Telas de histórico e de cadastro de glicemia



A cor com a qual o valor da glicemia é exibido no histórico depende se a mesma está dentro do intervalo desejado que foi definido. Além disso, o sistema também exibe abaixo do horário de cada glicemia, a diferença do seu valor quando comparado com o valor das glicemias obtidas no dia anterior dentro de um intervalo de meia hora.

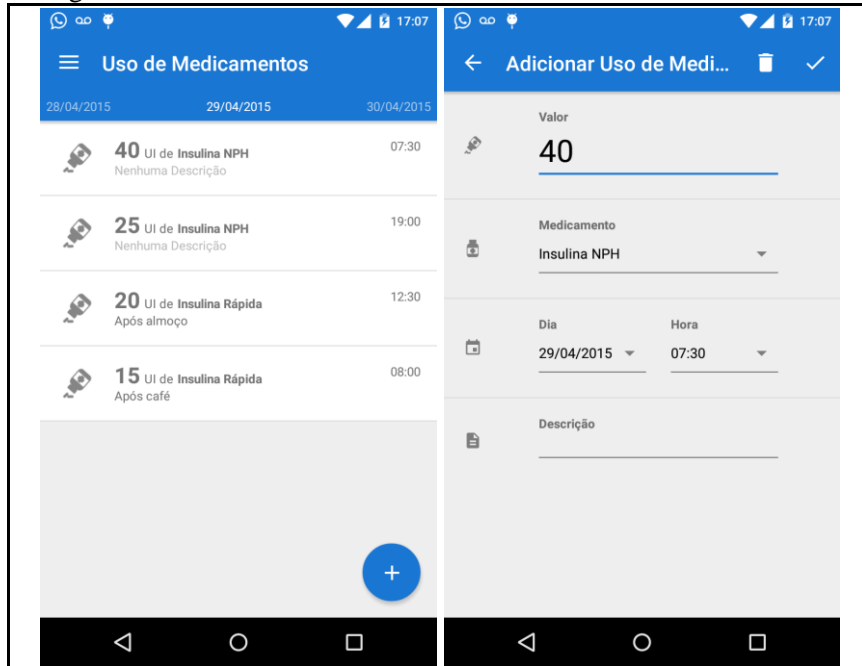
Na tela de medicamentos, o paciente pode incluir, editar ou excluir os medicamentos que podem ser selecionados nos usos de medicamentos. Os medicamentos são ordenados alfabeticamente e para cada um é exibido seu nome e sua unidade de medida. A Figura 30 apresenta a tela com a lista de medicamentos do usuário e a tela de cadastro de medicamento.

Figura 30 - Telas de lista e cadastro de medicamentos



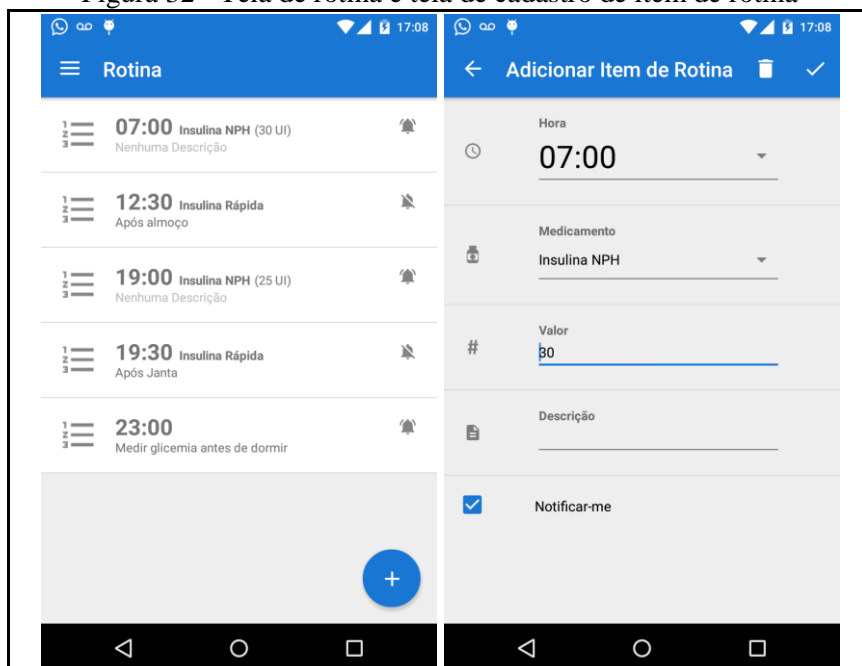
No histórico de uso de medicamentos, o paciente pode visualizar os medicamentos que usou em determinado dia, incluir uma nova utilização ou selecionar um registro existente para editá-lo ou excluí-lo. Os usos de medicamentos são exibidos em ordem cronológica, com os mais recentes no topo, e para cada um é exibido o nome do medicamento, a quantidade utilizada, o horário de utilização e descrição fornecida pelo paciente. A Figura 31 apresenta a tela de histórico e a tela de cadastro de uso de medicamento.

Figura 31 - Telas de histórico e de cadastro de uso de medicamento



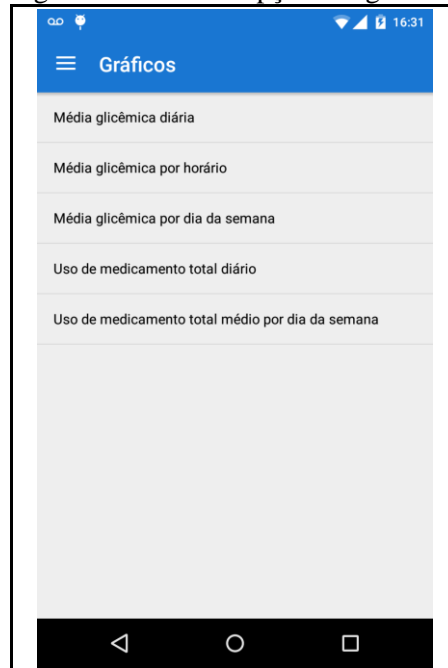
Na tela de rotina, o paciente pode montar uma rotina a ser seguida, incluindo itens de rotina. Para cada item de rotina é exibido seu horário, o medicamento (caso possua um), um valor (caso tenha sido selecionado um medicamento), a descrição fornecida pelo paciente e a opção de notificação, que caso selecionada, gera uma notificação de sistema no horário selecionado para o item. A Figura 32 apresenta a tela de rotina e a tela de cadastro de item de rotina.

Figura 32 - Tela de rotina e tela de cadastro de item de rotina



A tela com as opções de gráficos apresenta uma lista simples ao paciente, com a descrição de cada gráfico disponível, como demonstra a Figura 33.

Figura 33 - Tela de opções de gráficos



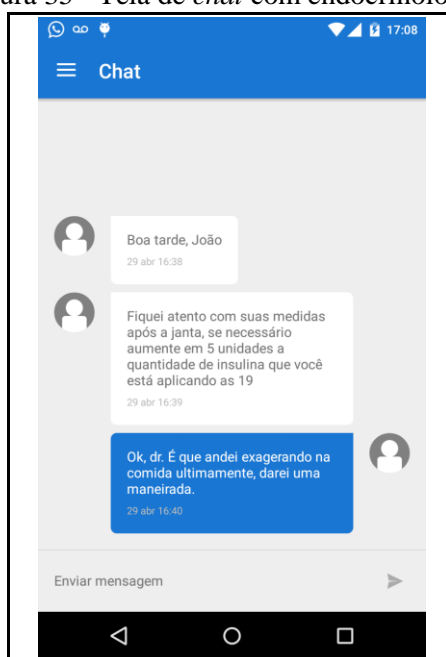
Após selecionar uma opção, o sistema gera uma nova tela com o respectivo gráfico e parametrizações possíveis. Além disso, é possível que o paciente selecione a opção “Gerar *e-mail*” no canto superior direito para que seja gerado um *e-mail* contendo o gráfico no aplicativo de sua escolha. A Figura 34 apresenta um gráfico gerado pelo aplicativo e o respectivo *e-mail* gerado contendo o mesmo gráfico, no aplicativo Gmail.

Figura 34 - Gráfico gerado pelo aplicativo e *e-mail* contendo-o



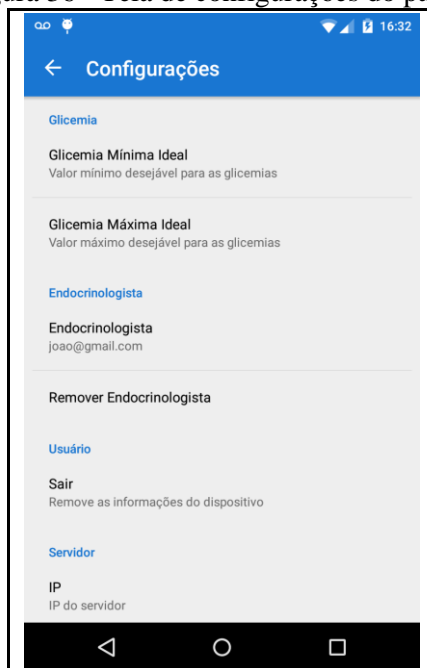
A tela de *chat* (Figura 35) só está disponível para o paciente caso ele possua um endocrinologista configurado. Nela, é possível enviar e ler as mensagens recebidas, com seu conteúdo e horário de envio. As mensagens enviadas pelo usuário ficam dispostas à direita, enquanto as recebidas ficam dispostas à esquerda, e todas são ordenadas cronologicamente. Para enviar uma nova mensagem basta digitar seu conteúdo no campo de texto na parte inferior e selecionar a opção de envio, representada por uma seta.

Figura 35 - Tela de *chat* com endocrinologista



Na tela de configurações, o paciente pode definir o intervalo de glicemia ideal para suas medidas, definir ou remover seu endocrinologista ou efetuar o *logoff*. As opções referentes ao intervalo de glicemia ideal exibem um *input* numérico quando selecionadas. A opção de definir um endocrinologista exhibe a tela de pesquisa de endocrinologista, e a opção *logoff* faz com que os dados do paciente sejam apagados da base local e o retorna para a tela de *login*. A Figura 36 apresenta a tela de configurações do paciente.

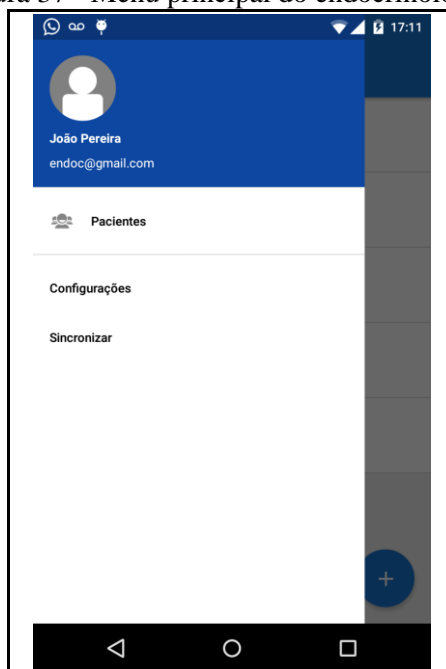
Figura 36 - Tela de configurações do paciente



3.3.2.2 Endocrinologista

O endocrinologista, após efetuar o *login*, tem acesso à sua lista de pacientes e à janela de configurações. Essas opções são disponibilizadas no menu principal da aplicação, conforme demonstrado na Figura 38.

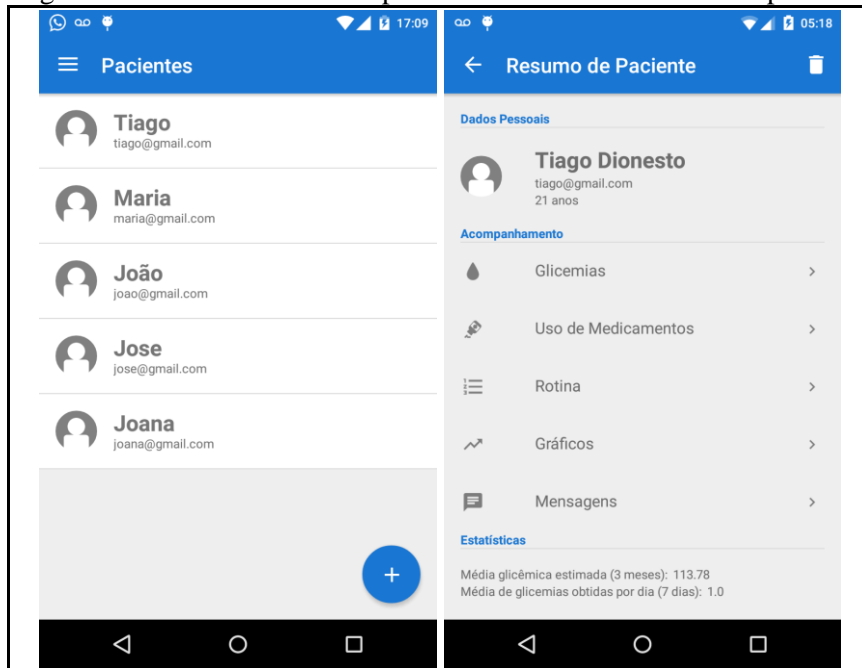
Figura 37 - Menu principal do endocrinologista



Na sua lista de pacientes são exibidos os nomes e *e-mails* dos mesmos. Ao selecionar um paciente, é exibida uma tela com informações adicionais sobre o mesmo, além das opções

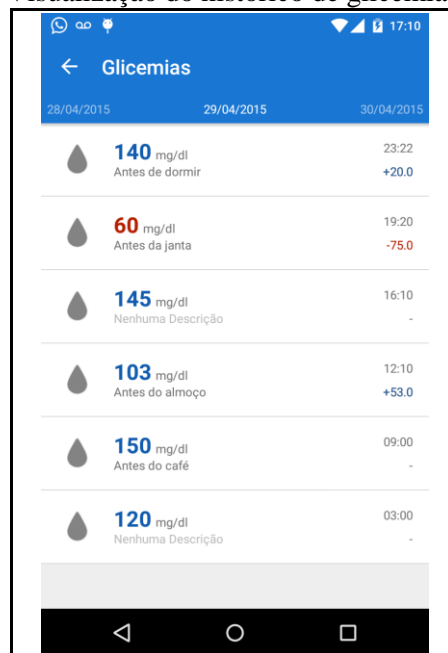
para visualizar seus históricos, rotina, gráficos de desempenho e chat. No canto direito superior o endocrinologista pode encontrar a opção para remover o paciente. A Figura 38 apresenta a lista de pacientes e a tela com detalhes do paciente.

Figura 38 - Tela com a lista de pacientes e tela com detalhes do paciente



As telas exibidas para o endocrinologista ao selecionar alguma das opções do paciente são semelhantes às telas apresentadas para o próprio paciente ao utilizar o aplicativo, porém sem as opções para inclusão e edição de informações. A Figura 39 apresenta o histórico do paciente ao ser visualizado pelo endocrinologista. Destaca-se a ausência do botão de adição.

Figura 39 - Visualização do histórico de glicemias do paciente



3.4 RESULTADOS E DISCUSSÃO

Este trabalho teve como objetivo o desenvolvimento de um aplicativo para a plataforma Android que oferecesse ao diabético uma forma de manter um diário de seu tratamento e possibilitasse o acompanhamento pelo seu endocrinologista, caso desejado. Este objetivo foi alcançado, com todos os requisitos definidos sendo atendidos com êxito.

O sistema produzido não só oferece ao usuário cadastros práticos que podem ser preenchidos em questão de segundos, como também uma interface bem organizada. A sincronização dos dados ocorre corretamente e permite que o usuário visualize e edite suas informações em qualquer dispositivo com o aplicativo, sendo necessário apenas entrar com suas credenciais.

Com relação ao acompanhamento médico, um endocrinologista pode utilizar o aplicativo para visualizar gráficos de performance de qualquer período desejado, além de ter acesso a cada registro do seu paciente, podendo ter uma noção aprimorada do seu tratamento e de sua rotina. Além disso, é possível ao paciente enviar *e-mails* contendo tais gráficos, não limitando o seu acompanhamento apenas a endocrinologistas que façam uso do sistema.

O *chat* presente no aplicativo permite a troca de mensagens entre paciente e endocrinologista sem que o profissional necessite fornecer informações pessoais, como número de celular ou *e-mail* ao paciente para resolução de dúvidas e sugestões no tratamento.

3.4.1 COMPARATIVO COM TRABALHOS CORRELATOS

No Quadro 3 é apresentado um comparativo entre as principais funcionalidades presentes no sistema desenvolvido (GluControl) e nos seus trabalhos correlatos.

Quadro 3 - Comparativo com trabalhos correlatos

Funcionalidade	GlicoCare (BAYER AG, 2013)	OnTrack (MEDIVO, 2014)	Diabetes Control (WIDÉN, 2010)	GluControl
Acompanhamento simultâneo pelo endocrinologista	Não	Não	Não	Sim
Armazenamento de dados na nuvem	Não	Não	Não	Sim

Configuração de lembretes	Sim	Não	Sim	Sim
Configuração do intervalo de glicemia desejado	Não	Sim	Sim	Sim
Geração de gráficos	Sim	Sim	Sim	Sim
Geração de relatórios	Não	Sim	Não	Não
Envio de gráficos por <i>e-mail</i>	Sim	Não	Sim	Sim
Gerenciamento de glicemias	Sim	Sim	Sim	Sim
Gerenciamento de medicamentos	Sim	Sim	Sim	Sim
Gerenciamento de outras informações (refeições, atividades físicas, etc.)	Sim	Sim	Sim	Não
Inserção de dados por <i>bluetooth</i>	Não	Não	Sim	Não
Troca de mensagens por <i>chat</i>	Não	Não	Não	Sim

Para efetuar o comparativo, foram utilizados os aplicativos disponíveis para download dos sistemas GlicoCare (BAYER AG, 2013) e OnTrack (MEDIVO, 2014). Para o DiabetesControl (WIDÉN, 2010) foram utilizadas as informações e *screenshots* apresentadas na sua respectiva tese.

Apesar de semelhantes em muitos aspectos, os sistemas se diferenciam em funcionalidades específicas, conforme o seu foco. Os aplicativos Glucocare (BAYER AG, 2013) e OnTrack (MEDIVO, 2014) focam na manutenção de um diário completo sobre as atividades do diabético, fornecendo opções para cadastro de outras informações além de glicemias e medicamentos, porém as opções para compartilhamento dessas informações são inexistentes ou bem limitadas. O aplicativo DiabetesControl (WIDÉN, 2010) é o único que oferece suporte teórico à inserção de dados via *bluetooth*, porém sua interface é simples se comparado aos demais sistemas. O GluControl, aplicativo desenvolvido neste trabalho, possui a maioria das funcionalidades implementadas pelos demais, também possibilitando o

acompanhamento por um endocrinologista e armazenando as informações do usuário na nuvem, permitindo sua utilização em diversos aparelhos sem a perda de dados.

4 CONCLUSÕES

Neste trabalho foi apresentado o desenvolvimento de um protótipo de aplicação que permite auxiliar o diabético no controle de seus indicadores de glicemia, possibilitando a manutenção de um diário do tratamento e o acompanhamento do mesmo por um médico endocrinologista. O sistema atendeu a todos os objetivos inicialmente definidos, se mostrando uma alternativa viável para diabéticos que demandem de acompanhamento especial.

Através do aplicativo, é possível ao diabético salvar suas glicemias, medicamentos e seus usos, configurar uma rotina, gerar gráficos de desempenho e conversar com seu endocrinologista, que tem acesso às suas informações a todo momento. A aplicação também permite a configuração de um intervalo ideal para as glicemias salvas, destacando para o usuário as que estiverem irregulares.

Todas as informações do usuário são salvas localmente e sincronizadas com o servidor do sistema, possibilitando a utilização do aplicativo mesmo sem internet e a troca de dispositivos sem perda de dados. É necessário apenas que o usuário efetue *login* na aplicação.

O aplicativo foi desenvolvido para a plataforma Android, utilizando a linguagem de programação Java, as bibliotecas Gson, MPAndroidChart e Joda Time e o banco de dados SQLite. O *web service* da aplicação também foi desenvolvido em Java, utilizando o *framework* Jersey e o banco de dados MySQL. Essas tecnologias e ferramentas se mostraram apropriadas para o sistema desenvolvido.

As maiores dificuldades encontradas pelo autor no desenvolvimento do trabalho ocorreram devido à falta de experiência no desenvolvimento para a plataforma Android e à documentação consideravelmente escassa da biblioteca MPAndroidChart.

4.1 EXTENSÕES

Apesar de satisfazer os objetivos estabelecidos, observa-se que para lançamento do aplicativo como produto comercial seriam necessários alguns ajustes, como possibilitar a sincronização automática de dados, ao invés de ser iniciada manualmente e maiores cuidados com a segurança das informações dos usuários. Além disso, seriam necessários testes extensivos com uma grande quantidade de usuários sincronizando informações para avaliar o comportamento do servidor.

Para trabalhos futuros, há várias possibilidades de melhorias para o sistema desenvolvido. Como principais, sugere-se:

- a) permitir que um maior número de informações sejam salvas pelo usuário, como exercícios físicos e alimentos consumidos, tornando ainda melhor o controle do diabético sobre sua doença;
- b) oferecer um maior número de gráficos e estatísticas, possibilitando também a geração de relatórios;
- c) possibilitar a sincronização automática com glicosímetros que possuíssem a tecnologia *bluetooth*;
- d) possibilitar a sincronização automática com bombas de infusão de insulina que possuíssem a tecnologia *bluetooth*;
- e) desenvolver uma interface *web*, possibilitando que os usuários utilizassem o sistema sem precisar de um smartphone com o aplicativo instalado;
- f) possibilitar a configuração de notificações para mensagens recebidas no aplicativo, e também possibilitar ao endocrinologista configurar notificações para quando algum de seus pacientes registrasse uma glicemia muito abaixo ou acima do esperado;
- g) possibilitar ao paciente decidir quais informações são disponibilizadas para seu endocrinologista.

REFERÊNCIAS

ACCU CHEK. **Accu-Chek Performa Nano**. [S.l.], 2014. Disponível em: <<https://www.accu-chek.com.br/br/produtos/monitoresdeglicemia/performanano.html>>. Acesso em: 15 jun. 2015.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES (Brasil). **Relatório Anual**: 2011. 2012. Disponível em: <<http://www.anatel.gov.br/Portal/verificaDocumentos/documento.asp?numeroPublicacao=278637&pub=original&filtro=1&documentoPath=278637.pdf>>. Acesso em: 16 jun. 2015.

BARRETO, Juliano. **Beep! A história dos Pagers**. [S.l.], 2011. Disponível em: <<http://info.abril.com.br/noticias/blogs/ctrlz/blog-info-ctrlz/beep-a-historia-dos-pagers/>>. Acesso em: 20 jun. 2015.

BAYER AG. **Aplicativo GlicoCare**. [S.l.], 2013. Disponível em: <<https://www.bayerparavoce.com.br/ferramentas/aplicativos/glicocare.aspx>>. Acesso em: 17 jun. 2015.

BRASIL. Ministério da Saúde. Define elenco de medicamentos e insumos disponibilizados pelo Sistema Único de Saúde, nos termos da Lei nº 11.347, de 2006, aos usuários portadores de diabetes mellitus. Portaria nº 2.583 de 10 de outubro de 2007. **Lex**: Diário oficial da República Federativa do Brasil, Brasília, 31 de dez. de 2007.

CONSUMER HEALTH INFORMATION CORPORATION. **Motivating Patients to Use Smartphone Health Apps**. [S.l.], 2011. Disponível em: <<http://www.consumer-health.com/press/2008/NewsReleaseSmartPhoneApps.php>>. Acesso em: 16 jun. 2015.

COSTA, Italo M. A. **The impact of mobile devices in the self-management of type II diabetes: A systematic review**. 2010. 38 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Bacharelado em Ciências da Computação, Universidade Federal de Pernambuco, Recife, 2010.

COSTA, Adriana Cássia da. **Um Modelo para Notificações em mHealth**. 2013. 99 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2013. Disponível em: <<http://repositorio.pucrs.br/dspace/bitstream/10923/1678/1/000449591-Texto+Completo-0.pdf>>. Acesso em: 18 jun. 2015.

GONÇALVES, Diana Maria Duarte. **Aplicação móvel para adoção de estilos de vida saudáveis em pessoas com Diabetes tipo 2**: Definição de estratégias e desenvolvimento de algoritmo. 2014. 145 f. Dissertação (Mestrado) - Curso de Engenharia Biomédica, Universidade de Coimbra, Coimbra, 2014. Disponível em: <[https://estudogeral.sib.uc.pt/bitstream/10316/26546/1/Tese Definitiva.pdf](https://estudogeral.sib.uc.pt/bitstream/10316/26546/1/Tese%20Definitiva.pdf)>. Acesso em: 20 jun. 2015.

GUARIENTE, Maria H. D. M. et al. Crianças e Adolescentes com Diabetes Mellitus: Vantagens e Limites da Monitorização. **Cogitare Enfermagem**, Curitiba, v. 7, n. 1, p.48-54, jan./jun. 2002.

INTERNATIONAL DIABETES FEDERATION. **IDF Diabetes Atlas**. 6. ed. Bruxelas, Bélgica: International Diabetes Federation, 2013. 160 p. Disponível em: <http://www.idf.org/sites/default/files/EN_6E_Atlas_Full_0.pdf>. Acesso em: 16 mar. 2015.

IWAYA, Leonardo Horn et al. Mobile health in emerging countries: A survey of research initiatives in Brazil. **International Journal of Medical Informatics**, [S.l.], v. 82, n. 5, p.283-298, maio 2013.

KUSZKA, Boris. **Dispositivos móveis: a interface com o mundo**. [S.l.], 2014. Disponível em: <<http://corporate.canaltech.com.br/coluna/mobile/Dispositivos-moveis-a-interface-com-o-mundo/>>. Acesso em: 17 jun. 2014.

MARIN, Heimar de Fátima. Sistemas de informação em saúde: considerações gerais. **Journal of Health Informatics**, [S.l.], v. 2, n. 1, p.20-24, jan. 2010. Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/viewFile/4/52>>. Acesso em: 17 jun. 2015.

MEDIVO. **OnTrack Diabetes**. [S.l.], 2014. Disponível em: <<http://www.medivo.com/ontrack/>>. Acesso em: 17 jun. 2015.

MURAKAMI, Alexandre. **vMonGluco - sistema de monitoramento contínuo de glicose**. 2007. Dissertação (Mestrado em Sistemas Eletrônicos) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-27062007-180347/>>. Acesso em: 17 jun. 2015.

PACAUD, Danièle et al. Successful Delivery of Diabetes Self-Care Education and Follow-Up through eHealth Media. **Canadian Journal of Diabetes**, [S.l.], v. 36, n. 5, p.257-262, out. 2012.

PAWAR, Pravin et al. A framework for the comparison of mobile patient monitoring systems. **Journal Of Biomedical Informatics**, Enschede, v. 45, n. 3, p.544-556, jun. 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1532046412000287>>. Acesso em: 16 jun. 2015.

PÉRES, Denise S. et al. Dificuldades dos Pacientes Diabéticos para o Controle da Doença: Sentimentos e Comportamentos. **Revista Latino-americana de Enfermagem**, São Paulo, v. 15, n. 6, p.1105-1112, dez. 2007.

SECRETARIA MUNICIPAL DA SAÚDE DE RIBEIRÃO PRETO. Ribeirão Preto, 2007?. Protocolo de Monitoramento da Glicemia. Disponível em: <<http://www.ribeiraopreto.sp.gov.br/ssauade/saudepessoal/farmacia/i16p-monit-glicemia.php>>. Acesso em: 17 jun. 2015.

TAN, Shannon Josephine. **The role of mHealth and eHealth in diabetes care**. 2014. 55 f. TCC (Graduação) - Curso de Health Policy and Management, Erasmus University Rotterdam, Rotterdam, 2014. Disponível em: <<http://thesis.eur.nl/pub/16464/Tan-S-J-353768-The-role-of-eHealth-mHealth-in-diabetes-care.pdf>>. Acesso em: 17 jun. 2015.

WIDÉN, Anders. **Diabetes care on smart phones running the Android platform: Design and implementation of a system to help self monitoring and managing**. 2010. 41 f. Dissertação

(Mestrado) - Curso de Projeto de Sistemas Eletrônicos Integrados, Departamento de Engenharia e Ciências da Computação, Chalmers University Of Technology, Göteborg, Sweden, 2010. Disponível em:

<<http://publications.lib.chalmers.se/records/fulltext/126271.pdf>>. Acesso em: 17 jun. 2015.

WORLD BANK. **Information and Communications for Development 2012: Maximizing Mobile**. Washington, D.C: World Bank, 2012. 244 p. Disponível em:

<<http://siteresources.worldbank.org/EXTINFORMATIONANDCOMMUNICATIONANDTECHNOLOGIES/Resources/IC4D-2012-Report.pdf>>. Acesso em: 16 jun. 2015.

WORLD HEALTH ORGANIZATION. **Global observatory for ehealth series: mHealth: new horizons for health through mobile technologies**. Genebra: World Health Organization, 2011. 111 p. Disponível em: <http://www.who.int/goe/publications/goe_mhealth_web.pdf>. Acesso em: 20 jun. 2015.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso do sistema.

No Quadro 4 tem-se o detalhamento do caso de uso “Efetuar cadastro”.

Quadro 4 - Caso de Uso 01 - "Efetuar cadastro"

UC01 - Efetuar cadastro

O sistema deverá permitir ao visitante efetuar o seu cadastro no sistema e se tornar um usuário.

Ator: Visitante.

Pós-Condição: O visitante teve seu cadastro efetuado.

Cenário Principal

1. Aplicativo exibe tela de cadastro;
2. Usuário fornece seus dados;
3. Aplicativo envia dados para o servidor;
4. Servidor verifica dados fornecidos;
5. Servidor salva o usuário no banco de dados;
6. Aplicativo exibe uma mensagem confirmando o cadastro.

Cenário de Exceção - Se no passo 2 do cenário principal o aplicativo verifica que alguma informação não foi preenchida:

1. Aplicativo informa ao usuário e não efetua seu cadastro.

Cenário de Exceção - Se no passo 4 do cenário principal o servidor verifica que o e-mail fornecido já foi utilizado por outro usuário:

1. Aplicativo informa ao usuário e não efetua seu cadastro.

Cenário de Exceção - Se no passo 3 do cenário principal não é possível a comunicação com o servidor:

1. Aplicativo informa ao usuário e não efetua seu cadastro.

No Quadro 5 tem-se o detalhamento do caso de uso “Efetuar login”.

Quadro 5 - Caso de Uso 02 - "Efetuar login"

UC02 - Efetuar login

O sistema deverá permitir ao usuário efetuar o *login* no sistema, fornecendo seu *e-mail* e senha.

Ator: Usuário.

Pós-Condição: O usuário efetuou *login* no sistema.

Cenário Principal

1. Aplicativo exibe tela de *login*;
2. Usuário fornece suas credenciais;
3. Aplicativo verifica credenciais junto ao servidor;
4. Aplicativo efetua o *login* do usuário.

Cenário de Exceção - Se no passo 3 do cenário principal o servidor verifica que as credenciais não estão corretas:

1. Sistema informa ao usuário e não efetua seu cadastro.

Cenário de Exceção - Se no passo 3 do cenário principal não é possível a comunicação com o servidor:

1. Sistema informa ao usuário e não efetua seu cadastro.

No Quadro 6 tem-se o detalhamento do caso de uso “Efetuar *logoff*”.

Quadro 6 - Caso de Uso 03 - "Efetuar *logoff*"

UC03 - Efetuar *logoff*

O sistema deverá permitir ao usuário logado efetuar o *logoff* do sistema, apagando as informações presentes na base de dados do aplicativo.

Ator: Usuário.

Pré-Condição: O usuário está logado no sistema.

Pós-Condição: O usuário efetuou *logoff* do sistema.

Pós-Condição: Os dados do usuário na base de dados do aplicativo foram apagados.

Cenário Principal

1. Aplicativo exibe tela de configurações;
2. Usuário seleciona a opção “Sair”;
3. Aplicativo recria o banco de dados e efetua o *logoff* do usuário.

No Quadro 7 tem-se o detalhamento do caso de uso “Configurar endocrinologista”.

Quadro 7 - Caso de Uso 04 - "Configurar endocrinologista"

UC04 - Configurar endocrinologista

O sistema deverá permitir ao paciente configurar um endocrinologista para acompanhar o seu tratamento.

Ator: Paciente.

Pré-Condição: O paciente está logado no sistema.

Pós-Condição: O paciente adicionou ou removeu um endocrinologista.

Cenário Principal

1. Aplicativo exibe tela de configurações;
2. Paciente seleciona a opção “Configurar endocrinologista”;
3. Aplicativo exibe tela de pesquisa de endocrinologistas;
4. Paciente digita nome ou usuário do endocrinologista;
5. Aplicativo requisita ao servidor o resultado da pesquisa;
6. Aplicativo exibe os resultados;
7. Paciente seleciona o endocrinologista;
8. Aplicativo atualiza o cadastro do paciente.

Cenário Exclusão

1. Aplicativo exibe tela de configurações;

2. Paciente seleciona a opção “Remover endocrinologista”;
3. Aplicativo atualiza o cadastro do paciente.

Cenário de Exceção - Se no passo 5 do cenário principal não é possível a comunicação com o servidor:

1. Sistema informa ao usuário e não mostra nenhum resultado.

Cenário de Exceção - Se no passo 6 do cenário principal o paciente não acha o seu endocrinologista:

1. Paciente refaz a pesquisa no passo 4.

No Quadro 8 tem-se o detalhamento do caso de uso “Gerenciar histórico de glicemias”.

Quadro 8 - Caso de Uso 05 - “Gerenciar histórico de glicemias”

UC05 - Gerenciar histórico de glicemias

O sistema deverá permitir ao paciente adicionar, alterar e excluir suas glicemias.

Ator: Paciente.

Pré-Condição: O paciente está logado no sistema.

Pós-Condição: Paciente inseriu, alterou ou excluiu alguma glicemia.

Cenário Inclusão

1. Aplicativo exibe o histórico de glicemias do paciente;
2. Paciente seleciona a opção de inclusão;
3. Aplicativo abre a tela de cadastro de glicemias;
4. Paciente preenche o formulário;
5. Paciente seleciona a opção para salvar;
6. Aplicativo verifica se o campo obrigatório Valor foi preenchido;
7. Aplicativo insere na base de dados a glicemia informada;
8. Aplicativo exibe o histórico de glicemias do paciente.

Cenário Alteração

1. Aplicativo exibe o histórico de glicemias do paciente;
2. Paciente seleciona a glicemia que deseja alterar;
3. Paciente seleciona a opção de alteração;
4. Aplicativo abre a tela de cadastro de glicemias, com o formulário já preenchido com as informações da glicemia selecionada;
5. Paciente altera as informações que desejar (Valor, Dia, Hora, Descrição);
6. Paciente seleciona a opção para salvar;
7. Aplicativo verifica se o campo obrigatório Valor foi preenchido;
8. Aplicativo altera na base de dados o registro selecionado;
9. Aplicativo exibe o histórico de glicemias do paciente.

Cenário Exclusão

1. Aplicativo exibe o histórico de glicemias do paciente;
2. Paciente seleciona a glicemia que deseja excluir;
3. Paciente seleciona a opção de exclusão;

4. Aplicativo salva o registro selecionado como excluído;
5. Aplicativo exibe o histórico de glicemias do paciente.

Cenário de Exceção - Se no passo 5 do cenário inclusão ou no passo 6 do cenário alteração o paciente não preencheu todos os campos obrigatórios:

1. Aplicativo informa ao usuário e não conclui o cadastro.

No Quadro 9 tem-se o detalhamento do caso de uso “Gerenciar o histórico de uso de medicamentos”.

Quadro 9 - Caso de Uso 06 - “Gerenciar o histórico de uso de medicamentos”

UC06 - Gerenciar o histórico de uso de medicamentos

O sistema deverá permitir ao usuário inserir, editar e excluir seus usos de medicamentos.

Ator: Paciente.

Pré-Condição: O paciente deve estar logado.

Pós-Condição: Paciente inseriu, alterou ou excluiu algum uso de medicamento.

Cenário Inclusão

1. Aplicativo exibe histórico de uso de medicamentos do paciente;
2. Paciente seleciona a opção de inclusão;
3. Aplicativo abre a tela de cadastro de uso de medicamentos;
4. Paciente preenche o formulário;
5. Paciente seleciona a opção para salvar;
6. Aplicativo verifica se todos os campos obrigatórios foram preenchidos (Valor, Medicamento);
7. Aplicativo insere na base de dados o uso de medicamento informado;
8. Aplicativo exibe o histórico de uso de medicamentos do paciente.

Cenário Alteração

1. Aplicativo exibe histórico de uso de medicamentos do paciente;
2. Paciente seleciona o uso de medicamento que deseja alterar;
3. Paciente seleciona a opção de alteração;
4. Aplicativo abre a tela de cadastro de uso de medicamentos, com o formulário já preenchido com as informações do uso de medicamento selecionado;
5. Paciente altera as informações que desejar (Valor, Medicamento, Dia, Hora, Descrição);
6. Paciente seleciona a opção para salvar;
7. Aplicativo verifica se todos os campos obrigatórios foram preenchidos;
8. Aplicativo altera na base de dados o registro selecionado;
9. Aplicativo exibe o histórico de uso de medicamentos do paciente.

Cenário Exclusão

1. Aplicativo exibe histórico de uso de medicamentos do paciente;
2. Paciente seleciona o uso de medicamento que deseja excluir;
3. Paciente seleciona a opção de exclusão;
4. Aplicativo salva o registro selecionado como excluído;
5. Aplicativo exibe o histórico de uso de medicamentos do paciente.

Cenário de Exceção - Se no passo 5 do cenário inclusão ou no passo 6 do cenário alteração o paciente não preencheu todos os campos obrigatórios:

1. Aplicativo informa ao usuário e não conclui o cadastro.

No Quadro 10 tem-se o detalhamento do caso de uso “Configurar itens de rotina”.

Quadro 10 - Caso de Uso 07 - “Configurar itens de rotina”

UC07 - Configurar itens de rotina

O sistema deverá permitir ao usuário configurar uma rotina a ser seguida, inserindo, alterando ou excluindo itens.

Ator: Paciente.

Pré-Condição: O paciente deve estar logado.

Pós-Condição: Paciente inseriu, alterou ou excluiu algum item de rotina.

Cenário Inclusão

1. Aplicativo exibe a tela de rotina;
2. Paciente seleciona a opção de inserção;
3. Aplicativo abre a tela de cadastro de itens de rotina;
4. Paciente preenche o formulário;
5. Paciente seleciona a opção para salvar;
6. Aplicativo verifica se o campo obrigatório Hora foi preenchido;
7. Aplicativo insere na base de dados o item de rotina informada;
8. Aplicativo exibe a rotina do paciente.

Cenário Alteração

1. Aplicativo exibe a tela de rotina;
2. Paciente seleciona o item de rotina que deseja alterar;
3. Paciente seleciona a opção de alteração;
4. Aplicativo abre a tela de cadastro de itens de rotina, com o formulário já preenchido com as informações do item de rotina selecionada;
5. Paciente altera as informações que desejar (Hora, Medicamento, Quantidade do medicamento, Descrição, Notificação);
6. Paciente seleciona a opção para salvar;
7. Aplicativo verifica se o campo obrigatório Hora foi preenchido;
8. Aplicativo insere na base de dados o item de rotina informado;
9. Aplicativo exibe a rotina do paciente.

Cenário Exclusão

1. Aplicativo exibe a tela de rotina;
2. Paciente seleciona o item de rotina que deseja excluir;
3. Paciente seleciona a opção de exclusão;
4. Aplicativo salva o registro selecionado como excluído;
5. Aplicativo exibe a rotina do paciente.

Cenário de Exceção - Se no passo 5 do cenário inclusão ou no passo 6 do cenário alteração

o paciente não preencheu todos os campos obrigatórios:

1. Aplicativo informa ao usuário e não conclui o cadastro.

No Quadro 11 tem-se o detalhamento do caso de uso “Parametrizar intervalo de glicemia desejado”.

Quadro 11 - Caso de Uso 08 - “Parametrizar intervalo de glicemia desejado”

UC08 - Parametrizar intervalo de glicemia desejado

O sistema deverá permitir ao usuário parametrizar o intervalo de glicemia desejado para suas glicemias. Essa informação será utilizada para destacar as glicemias fora do intervalo no histórico de glicemias.

Ator: Paciente.

Pré-Condição: O paciente deve estar logado.

Pós-Condição: Paciente modificou o intervalo desejado.

Cenário Principal

1. Aplicativo exibe a tela de configurações;
2. Paciente seleciona a opção “Glicemia mínima ideal” ou “Glicemia máxima ideal”;
3. Aplicativo exibe campo numérico para digitação, preenchido caso já exista valor salvo;
4. Paciente fornece o valor;
5. Aplicativo atualiza as preferências do paciente.

No Quadro 12 tem-se o detalhamento do caso de uso “Conversar com chat por histórico”.

Quadro 12 - Caso de Uso 09 - “Conversar com chat por histórico”

UC09 - Conversar com chat por histórico

O sistema deverá permitir ao paciente e ao endocrinologista conversarem entre si por chat com histórico.

Atores: Paciente, Endocrinologista.

Pré-Condição: O usuário deve estar logado.

Pós-Condição: Usuário enviou uma mensagem.

Cenário Principal

1. Aplicativo exibe a tela de chat;
2. Usuário digita uma mensagem;
3. Usuário seleciona a opção “Enviar”;
4. Aplicativo salva a mensagem.

No Quadro 13 tem-se o detalhamento do caso de uso “Visualizar gráficos de performance”.

Quadro 13 - Caso de Uso 10 - “Visualizar gráficos de performance”

UC10 - Visualizar gráficos de performance

O sistema deverá permitir ao paciente e ao endocrinologista visualizarem gráficos de performance do paciente.

Atores: Paciente, Endocrinologista.

Pré-Condição: O usuário deve estar logado.

Cenário Principal

1. Aplicativo exibe a tela com gráficos disponíveis;
2. Usuário seleciona um gráfico que deseja visualizar;
3. Aplicativo exibe o gráfico selecionado.

No Quadro 14 tem-se o detalhamento do caso de uso “Gerar e-mail contendo gráfico”.

Quadro 14 - Caso de Uso 11 - “Gerar e-mail contendo gráfico”

UC11 - Gerar e-mail contendo gráfico

O sistema deverá permitir ao paciente e ao endocrinologista gerarem um *e-mail* contendo o gráfico selecionado.

Atores: Paciente, Endocrinologista.

Pré-Condição: O usuário deve estar logado.

Cenário Principal

1. Usuário seleciona a opção “E-mail” na tela com o gráfico desejado;
2. Aplicativo gera *e-mail* com as informações relativas ao gráfico, que vai em anexo.

No Quadro 15 tem-se o detalhamento do caso de uso “Visualizar lista de pacientes”.

Quadro 15 - Caso de Uso 12 - “Visualizar lista de pacientes”

UC12 - Visualizar lista de pacientes

O sistema deverá permitir ao endocrinologista visualizar uma lista simples com todos os seus pacientes.

Atores: Endocrinologista.

Pré-Condição: O endocrinologista deve estar logado.

Cenário Principal

1. Endocrinologista seleciona a opção “Pacientes” no menu principal da aplicação;
2. Aplicativo exibe uma lista com todos os pacientes atuais do endocrinologista, contendo nome e *e-mail* dos mesmos.

No Quadro 16 tem-se o detalhamento do caso de uso “Visualizar informações do paciente”.

Quadro 16 - Caso de Uso 13 - “Visualizar informações do paciente”

UC13 - Visualizar informações do paciente

O sistema deverá permitir ao endocrinologista visualizar informações mais detalhadas de seu

paciente.

Atores: Endocrinologista.

Pré-Condição: O endocrinologista deve estar logado.

Cenário Principal

1. Aplicativo exibe a tela com a lista de pacientes do endocrinologista;
2. Endocrinologista seleciona um paciente;
3. Aplicativo exibe tela com as informações do paciente, contendo nome, e-mail, idade, estatísticas, opções para visualização de históricos e rotina e a opção de remover o paciente.

No Quadro 17 tem-se o detalhamento do caso de uso “Remover paciente”.

Quadro 17 - Caso de Uso 14 - “Remover paciente”

UC14 - Remover paciente

O sistema deverá permitir ao endocrinologista remover um paciente da sua lista de acompanhamento.

Atores: Endocrinologista.

Pré-Condição: O endocrinologista deve estar logado.

Pós-Condição: O endocrinologista não mais tem acesso às informações do paciente.

Cenário Principal

1. Aplicativo exibe a tela com a lista de pacientes do endocrinologista;
2. Endocrinologista seleciona um paciente;
3. Aplicativo exibe a tela de informações do paciente;
4. Endocrinologista seleciona a opção “Remover”;
5. Aplicativo atualiza o cadastro do paciente;
6. Aplicativo exibe a tela com a lista de pacientes do endocrinologista.

No Quadro 18 tem-se o detalhamento do caso de uso “Visualizar históricos do paciente”.

Quadro 18 - Caso de Uso 15 - “Visualizar históricos do paciente”

UC15 - Visualizar históricos do paciente

O sistema deverá permitir ao endocrinologista visualizar os históricos de glicemia e uso de medicamentos dos seus pacientes.

Atores: Endocrinologista.

Pré-Condição: O endocrinologista deve estar logado.

Cenário Principal

1. Aplicativo exibe a tela com a lista de pacientes do endocrinologista;
2. Endocrinologista seleciona um paciente;
3. Aplicativo exibe a tela de informações do paciente;
4. Endocrinologista seleciona a opção “Glicemias” ou “Uso de medicamentos”;
5. Aplicativo exibe a tela com o histórico selecionado.

No Quadro 19 tem-se o detalhamento do caso de uso “Visualizar rotina do paciente”.

Quadro 19 - Caso de Uso 16 - “Visualizar rotina do paciente”

UC16 - Visualizar rotina do paciente

O sistema deverá permitir ao endocrinologista visualizar os itens de rotina dos seus pacientes.

Atores: Endocrinologista.

Pré-Condição: O endocrinologista deve estar logado.

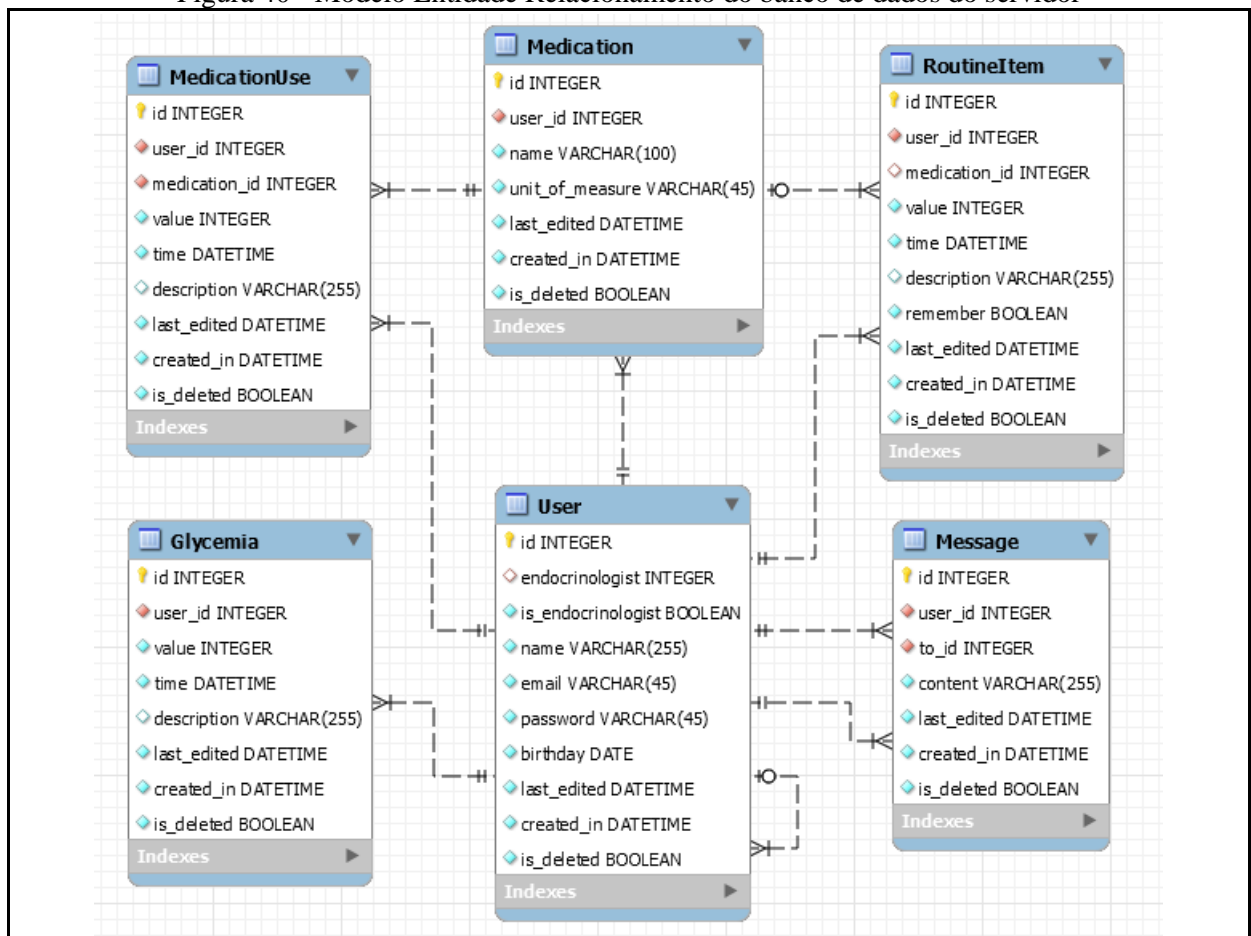
Cenário Principal

1. Aplicativo exibe a tela com a lista de pacientes do endocrinologista;
2. Endocrinologista seleciona um paciente;
3. Aplicativo exibe a tela de informações do paciente;
4. Endocrinologista seleciona a opção “Rotina”;
5. Aplicativo exibe a tela com a rotina do paciente.

APÊNDICE B – Modelos de Entidade e Relacionamento

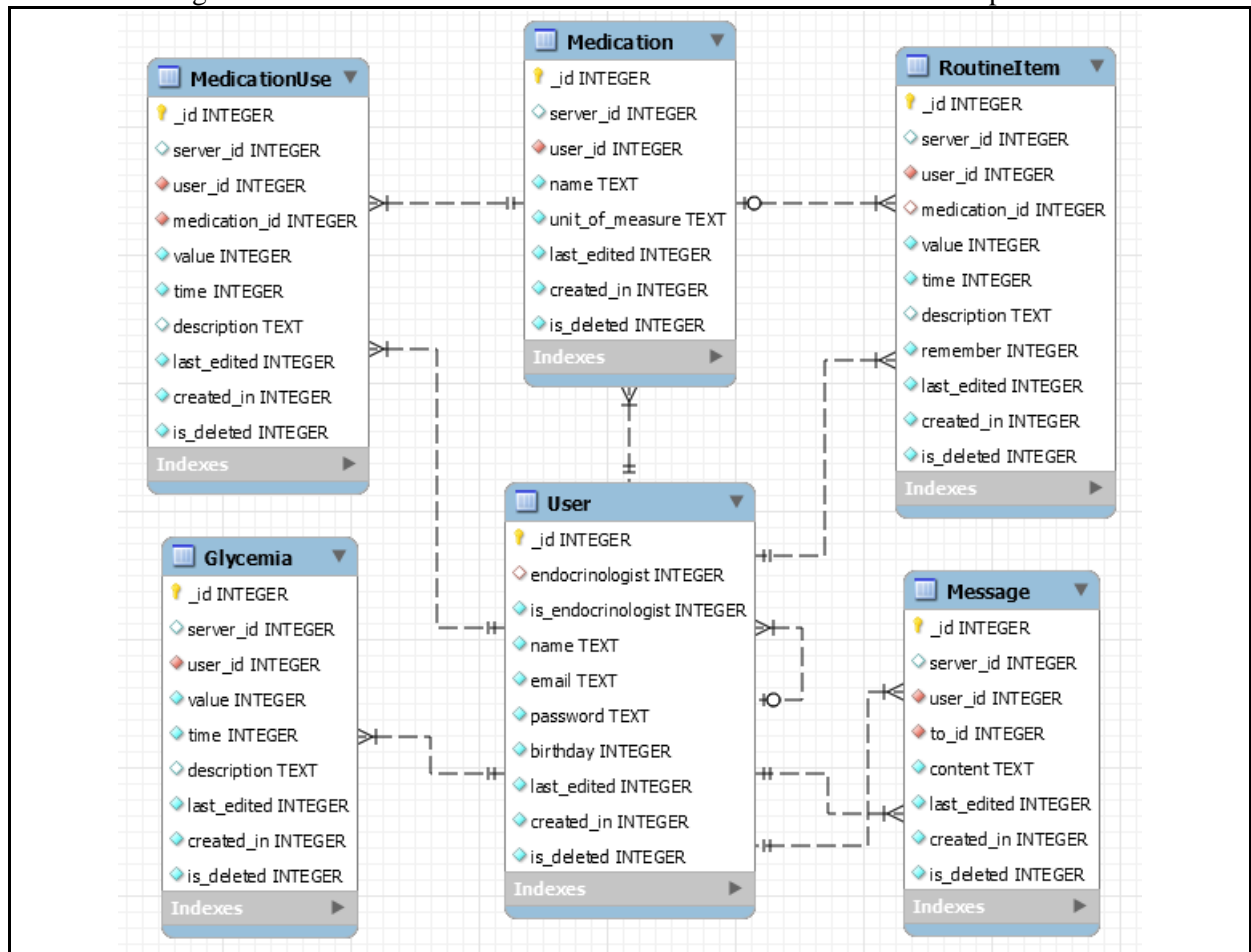
Neste apêndice são apresentados os modelos de entidade relacionamento do banco de dados do servidor e do aplicativo móvel, sendo que o detalhamento dos mesmos se encontra nos apêndices C e D, respectivamente. Para a criação dos MERs foi utilizada a ferramenta MySQL Workbench. A Figura 40 representa o MER referente ao banco de dados no servidor.

Figura 40 - Modelo Entidade Relacionamento do banco de dados do servidor



Na Figura 41 apresenta-se o MER referente ao banco de dados do aplicativo móvel. Ambos os bancos de dados possuem as mesmas entidades, no entanto, todas as entidades presentes no banco de dados do aplicativo, com exceção de `User`, possuem o atributo `server_id`, utilizado para fins de sincronia.

Figura 41 - Modelo Entidade Relacionamento do banco de dados do aplicativo



APÊNDICE C – Dicionário de Dados do Servidor

Este Apêndice apresenta a descrição das tabelas do banco de dados do servidor, apresentadas na seção de especificação deste trabalho. Os tipos de dados utilizados nos atributos são:

- a) *integer*: armazena números inteiros;
- b) *varchar*: armazena caracteres alfanuméricos;
- c) *datetime*: armazena data e hora;
- d) *date*: armazena apenas a data;
- e) *boolean*: armazena números binários.

No Quadro 20 tem-se o dicionário de dados da tabela “User”.

Quadro 20 - Tabela "User"

User					
Armazena os usuários					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código do usuário	<i>INTEGER</i>		Sim	Não
endocrinologist	Código do endocrinologista	<i>INTEGER</i>		Não	Sim
is_endocrinologist	Identifica se o usuário é um endocrinologista	<i>BOOLEAN</i>		Não	Não
name	Nome do usuário	<i>VARCHAR</i>	255	Não	Não
email	<i>E-mail</i> do usuário	<i>VARCHAR</i>	45	Não	Não
password	Senha do usuário	<i>VARCHAR</i>	45	Não	Não
birthday	Data de nascimento do usuário	<i>DATE</i>		Não	Não
last_edited	Momento em que o usuário foi editado pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que	<i>DATETIME</i>		Não	Não

	o usuário foi registrado				
is_deleted	Identifica se o usuário foi logicamente excluído	<i>BOOLEAN</i>		Não	Não

No Quadro 21 tem-se o dicionário de dados da tabela “Glycemia”.

Quadro 21 - Tabela "Glycemia"

Glycemia					
Armazena as glicemias					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código da glicemia	<i>INTEGER</i>		Sim	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
value	Valor da glicemia	<i>INTEGER</i>		Não	Não
time	Momento em que o usuário obteve a glicemia	<i>DATETIME</i>		Não	Não
description	Descrição da glicemia	<i>VARCHAR</i>	255	Não	Não
last_edited	Momento em que a glicemia foi editada pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que a glicemia foi registrada	<i>DATETIME</i>		Não	Não
is_deleted	Identifica se a glicemia foi logicamente excluída	<i>BOOLEAN</i>		Não	Não

No Quadro 22 tem-se o dicionário de dados da tabela “MedicationUse”.

Quadro 22 - Tabela "MedicationUse"

MedicationUse					
Armazena os usos de medicamentos					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código do uso de medicamento	<i>INTEGER</i>		Sim	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
medication_id	Código do medicamento	<i>INTEGER</i>		Não	Sim
value	Valor que foi utilizado do medicamento	<i>INTEGER</i>		Não	Não
time	Momento em que o usuário utilizou o medicamento	<i>DATETIME</i>		Não	Não
description	Descrição do uso de medicamento	<i>VARCHAR</i>	255	Não	Não
last_edited	Momento em que o uso de medicamento foi editado pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que o uso de medicamento foi registrado	<i>DATETIME</i>		Não	Não
is_deleted	Identifica se o uso de medicamento foi logicamente excluído	<i>BOOLEAN</i>		Não	Não

No Quadro 23 tem-se o dicionário de dados da tabela “Medication”.

Quadro 23 - Tabela "Medication"

Medication					
Armazena os medicamentos					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código medicamento	<i>INTEGER</i>		Sim	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
name	Nome do medicamento	<i>VARCHAR</i>	100	Não	Não
unit_of_measure	Unidade de medida utilizada para o medicamento	<i>VARCHAR</i>	45	Não	Não
last_edited	Momento em que o medicamento foi editado pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que o medicamento foi registrado	<i>DATETIME</i>		Não	Não
is_deleted	Identifica se o medicamento foi logicamente excluído	<i>BOOLEAN</i>		Não	Não

No Quadro 24 tem-se o dicionário de dados da tabela "RoutineItem".

Quadro 24 - Tabela "RoutineItem"

RoutineItem					
Armazena os itens de rotina					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código do item de rotina	<i>INTEGER</i>		Sim	Não

user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
medication_id	Código do medicamento	<i>INTEGER</i>		Não	Sim
value	Valor a ser utilizado do medicamento	<i>INTEGER</i>		Não	Não
time	Horário do item de rotina	<i>DATETIME</i>		Não	Não
description	Descrição do item de rotina	<i>VARCHAR</i>	255	Não	Não
remember	Se o usuário deve ser notificado no horário	<i>BOOLEAN</i>		Não	Não
last_edited	Momento em que o item de rotina foi editado pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que o item de rotina foi registrado	<i>DATETIME</i>		Não	Não
is_deleted	Identifica se o item de rotina foi logicamente excluído	<i>BOOLEAN</i>		Não	Não

No Quadro 25 tem-se o dicionário de dados da tabela “Message”.

Quadro 25 - Tabela "Message"

Message					
Armazena as mensagens					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código da mensagem	<i>INTEGER</i>		Sim	Não
user_id	Código do usuário remetente	<i>INTEGER</i>		Não	Sim

to_id	Código do usuário destinatário	<i>INTEGER</i>		Não	Sim
content	Conteúdo da mensagem	<i>VARCHAR</i>	255	Não	Não
last_edited	Momento em que a mensagem foi editada pela última vez	<i>DATETIME</i>		Não	Não
created_in	Momento em que a mensagem foi registrada	<i>DATETIME</i>		Não	Não
is_deleted	Identifica se a mensagem foi logicamente excluída	<i>BOOLEAN</i>		Não	Não

APÊNDICE D – Dicionário de Dados do Aplicativo

Este Apêndice apresenta a descrição das tabelas do banco de dados do aplicativo móvel, apresentadas na seção de especificação deste trabalho. O banco de dados SQLite não trabalha com tipos específicos para cada coluna, mas sim com cinco classes de armazenamento: *null*, *integer*, *real*, *text* e *blob*. Assim sendo, foram definidas as seguintes classes para os atributos:

- a) *integer*: armazena números inteiros;
- b) *text*: armazena caracteres alfanuméricos.

No Quadro 26 tem-se o dicionário de dados da tabela “User”.

Quadro 26 - Tabela "User"

User					
Armazena os usuários					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código do usuário	<i>INTEGER</i>		Sim	Não
endocrinologist	Código do endocrinologista	<i>INTEGER</i>		Não	Sim
is_endocrinologist	Identifica se o usuário é um endocrinologista	<i>INTEGER</i>		Não	Não
name	Nome do usuário	<i>TEXT</i>		Não	Não
email	<i>E-mail</i> do usuário	<i>TEXT</i>		Não	Não
password	Senha do usuário	<i>TEXT</i>		Não	Não
birthday	Data de nascimento do usuário	<i>INTEGER</i>		Não	Não
last_edited	Momento em que o usuário foi editado pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que o usuário foi	<i>INTEGER</i>		Não	Não

	registrado				
is_deleted	Identifica se o usuário foi logicamente excluído	<i>INTEGER</i>		Não	Não

No Quadro 27 tem-se o dicionário de dados da tabela “Glycemia”.

Quadro 27 - Tabela "Glycemia"

Glycemia					
Armazena as glicemias					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código local da glicemia	<i>INTEGER</i>		Sim	Não
server_id	Código da glicemia no servidor	<i>INTEGER</i>		Não	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
value	Valor da glicemia	<i>INTEGER</i>		Não	Não
time	Momento em que o usuário obteve a glicemia	<i>INTEGER</i>		Não	Não
description	Descrição da glicemia	<i>TEXT</i>		Não	Não
last_edited	Momento em que a glicemia foi editada pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que a glicemia foi registrada	<i>INTEGER</i>		Não	Não
is_deleted	Identifica se a glicemia foi logicamente excluída	<i>INTEGER</i>		Não	Não

No Quadro 28 tem-se o dicionário de dados da tabela “MedicationUse”.

Quadro 28 - Tabela "MedicationUse"

MedicationUse					
Armazena os usos de medicamentos					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código local do uso de medicamento	<i>INTEGER</i>		Sim	Não
server_id	Código do uso de medicamento no servidor	<i>INTEGER</i>		Não	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
medication_id	Código do medicamento	<i>INTEGER</i>		Não	Sim
value	Valor que foi utilizado do medicamento	<i>INTEGER</i>		Não	Não
time	Momento em que o usuário utilizou o medicamento	<i>INTEGER</i>		Não	Não
description	Descrição do uso de medicamento	<i>TEXT</i>		Não	Não
last_edited	Momento em que o uso de medicamento foi editado pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que o uso de medicamento foi registrado	<i>INTEGER</i>		Não	Não
is_deleted	Identifica se o uso de medicamento foi logicamente excluído	<i>INTEGER</i>		Não	Não

No Quadro 29 tem-se o dicionário de dados da tabela “Medication”.

Quadro 29 - Tabela "Medication"

Medication					
Armazena os medicamentos					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código local do medicamento	<i>INTEGER</i>		Sim	Não
server_id	Código do medicamento no servidor	<i>INTEGER</i>		Não	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
name	Nome do medicamento	<i>TEXT</i>		Não	Não
unit_of_measure	Unidade de medida utilizada para o medicamento	<i>TEXT</i>		Não	Não
last_edited	Momento em que o medicamento foi editado pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que o medicamento foi registrado	<i>INTEGER</i>		Não	Não
is_deleted	Identifica se o medicamento foi logicamente excluído	<i>INTEGER</i>		Não	Não

No Quadro 30 tem-se o dicionário de dados da tabela “RoutineItem”.

Quadro 30 - Tabela "RoutineItem"

RoutineItem					
Armazena os itens de rotina					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código local do item de rotina	<i>INTEGER</i>		Sim	Não
server_id	Código do item de rotina no servidor	<i>INTEGER</i>		Não	Não
user_id	Código do usuário	<i>INTEGER</i>		Não	Sim
medication_id	Código do medicamento	<i>INTEGER</i>		Não	Sim
value	Valor a ser utilizado do medicamento	<i>INTEGER</i>		Não	Não
time	Horário do item de rotina	<i>INTEGER</i>		Não	Não
description	Descrição do item de rotina	<i>TEXT</i>		Não	Não
remember	Se o usuário deve ser notificado no horário	<i>INTEGER</i>		Não	Não
last_edited	Momento em que o item de rotina foi editado pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que o item de rotina foi registrado	<i>INTEGER</i>		Não	Não
is_deleted	Identifica se o item de rotina foi logicamente excluído	<i>INTEGER</i>		Não	Não

No Quadro 31 tem-se o dicionário de dados da tabela "Message".

Quadro 31 - Tabela "Message"

Message					
Armazena as mensagens					
Campo	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
_id	Código local da mensagem	<i>INTEGER</i>		Sim	Não
server_id	Código da mensagem no servidor	<i>INTEGER</i>		Não	Não
user_id	Código do usuário remetente	<i>INTEGER</i>		Não	Sim
to_id	Código do usuário destinatário	<i>INTEGER</i>		Não	Sim
content	Conteúdo da mensagem	<i>TEXT</i>		Não	Não
last_edited	Momento em que a mensagem foi editada pela última vez	<i>INTEGER</i>		Não	Não
created_in	Momento em que a mensagem foi registrada	<i>INTEGER</i>		Não	Não
is_deleted	Identifica se a mensagem foi logicamente excluída	<i>INTEGER</i>		Não	Não