



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Doce Desafio - Sistema Integrado para Controle de Diabetes Tipo 1

Matheus Luís Teixeira Rosa
Guilherme de Oliveira Ozaka

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora
Prof.^a Dr.^a Aletéia Patricia Favacho

Coorientadora
Prof.^a Dr.^a Jane Dullius

Brasília
2016

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Henrique Veneziano

Banca examinadora composta por:

Prof.^a Dr.^a Aletéia Patricia Favacho (Orientadora) — CIC/UnB
Prof.^a Dr.^a Maristela Terto de Holanda — CIC/UnB
Prof. MSc Paulo Franklin von Paumgartten Júnior — Banco do Brasil

CIP — Catalogação Internacional na Publicação

Teixeira Rosa, Matheus Luís.

Doce Desafio - Sistema Integrado para Controle de Diabetes Tipo 1 /
Matheus Luís Teixeira Rosa, Guilherme de Oliveira Ozaka. Brasília :
UnB, 2016.

147 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2016.

1. Doce Desafio, 2. Diabetes Tipo 1, 3. Android, 4. Aplicativo para
Dispositivos Móveis

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Doce Desafio - Sistema Integrado para Controle de Diabetes Tipo 1

Matheus Luís Teixeira Rosa
Guilherme de Oliveira Ozaka

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof.^a Dr.^a Aletéia Patricia Favacho (Orientadora)
CIC/UnB

Prof.^a Dr.^a Maristela Terto de Holanda Prof. MSc Paulo Franklin von Paumgartten Júnior
CIC/UnB Banco do Brasil

Prof. Dr. Wilson Henrique Veneziano
Coordenador do Curso de Computação — Licenciatura

Brasília, 11 de Janeiro de 2016

Dedicatória

Guilherme de Oliveira Ozaka. Dedico a minha família, amigos, professores e comunidade este trabalho. Tentando assim retribuir um pouco do apoio e confiança que eles depositaram em mim.

Matheus Luís Teixeira Rosa. Dedico a meus pais, que sempre incentivaram meu desenvolvimento e busca pelo aprendizado, esforçando-se para me proporcionar uma boa educação e possibilitar meus sonhos.

Agradecimentos

Guilherme de Oliveira Ozaka. Agradeço a minha família por toda base afetiva e apoio que me proporcionaram desde que vim a esse mundo, aos professores por dedicarem suas vidas ao ensino e acreditarem que só assim o mundo pode vir a ser um mundo melhor e por fim a professora orientadora Aletéia Favacho, que dedicou boa parte do tempo e da paciência para nos auxiliar e dar suporte para a finalização dessa monografia.

Matheus Luís Teixeira Rosa. Agradeço a Deus pela vida, a minha família pelo apoio e amor, a professora e orientadora Aletéia Favacho pela colaboração e dedicação no desenvolvimento deste trabalho e a todos colegas e amigos que fiz durante estes anos de aprendizado.

Resumo

A diabetes *mellitus* é uma doença de abrangência mundial, que atinge 9 milhões de brasileiros. Seu tratamento exige acompanhamento clínico no qual se faz necessário monitorar e controlar dados do paciente no tocante à alimentação, prática de atividade física e administração de insulina. O aplicativo Doce Desafio foi desenvolvido em 2013 na Universidade de Brasília para facilitar o processo de acompanhamento da saúde e a realização de anotações pessoais feitas pelos portadores de diabetes tipo 1 diariamente. Após um ano de utilização do aplicativo pelos pacientes, observou-se a necessidade de implantação de melhorias, com a inclusão de novas funcionalidades no aplicativo para facilitar sua utilização e aumentar sua usabilidade. Também foi demanda a expansão do sistema para uma versão web, visando tornar mais prático o acesso aos dados e aprimorar o acompanhamento dos pacientes pelo corpo clínico. Portanto, após levantamento de requisitos, foram implantadas as melhorias consideradas necessárias no aplicativo *Android* e também foi desenvolvida uma aplicação *web* que se integra com o aplicativo via *RESTful Web Services*. O presente trabalho visa expor as melhorias que foram arquitetadas e implementadas para o aplicativo Doce Desafio e apresentar a aplicação web desenvolvida, incluindo a metodologia de desenvolvimento adotada.

Palavras-chave: Doce Desafio, Diabetes Tipo 1, Android, Aplicativo para Dispositivos Móveis

Abstract

Diabetes mellitus is a world-wide disease, which affects 9 million Brazilians. Its treatment requires clinical follow-up in which it is necessary to monitor and control patient data regarding to nutrition, physical activity and insulin administration. The Doce Desafio app was developed in 2013 at the University of Brasilia in order to facilitate the health monitoring process and note taking by people with type 1 diabetes in a daily basis. After one year using the app by patients, it was observed the need for improvements, with the inclusion of new features in the app for ease of use and increase its usability. It was also demanded the expansion of the system to a web version, in order to make more practical access to data and improve patients follow-up by medical staff. Therefore, after requirements elicitation it was implemented the improvements deemed necessary in the Android app and it was also developed a web application that is integrated with the application via RESTful Web Services. This study aims to expose the improvements that have been devised and implemented to the Doce Desafio app and present the developed web application, including the adopted development methodology.

Keywords: Doce Desafio, Type 1 Diabetes, Android, Mobile App

Sumário

1	Introdução	1
1.1	Problema	2
1.2	Justificativa	2
1.3	Objetivo	2
1.3.1	Objetivos Específicos	2
1.4	Resultados Esperados	3
1.5	Metodologia	3
1.6	Organização do Trabalho	3
2	Diabetes	5
2.1	Definição	5
2.2	Estatísticas	6
2.3	Diagnóstico	8
2.4	Causas e Sintomas	10
2.5	Diabetes Tipo 1 (DM1)	11
2.6	Diabetes Tipo 2 (DM2)	11
2.7	Considerações Finais	12
3	Desenvolvimento de Aplicativos <i>Android</i> e Aplicações Web	13
3.1	Desenvolvimento de Software	13
3.1.1	Processo de Software	13
3.1.2	Arquitetura de Software	15
3.2	Plataforma <i>Android</i>	16
3.2.1	<i>Kernel do Linux</i>	17
3.2.2	Arquitetura	17
3.2.3	<i>Android Runtime</i>	19
3.2.4	Versões	19
3.2.5	Desenvolvimento para <i>Android</i>	21
3.3	Desenvolvimento Web com Java	22
3.3.1	Fases de uma Aplicação Java	22
3.3.2	Características	23
3.3.3	<i>Java EE (Enterprise Edition)</i>	23
3.3.4	<i>JSP (JavaServer Pages)</i>	24
3.4	Considerações Finais	25

4	SOA e <i>Web Services</i>	26
4.1	SOA (<i>Service Oriented Architecture</i>)	26
4.1.1	Características	28
4.2	Composição de Serviços	29
4.3	Web Services	30
4.3.1	Especificações	30
4.4	Considerações Finais	33
5	Aplicativo Doce Desafio	34
5.1	Contextualização	34
5.2	Aplicativo <i>Android</i> - Primeira Versão	35
5.2.1	Requisitos	36
5.2.2	Requisitos não Funcionais	37
5.2.3	Banco de Dados	37
5.3	Considerações Finais	38
6	Sistema Doce Desafio - Aplicativo <i>Android</i> e Aplicação Web	39
6.1	Aplicativo <i>Android</i> - Segunda Versão	39
6.1.1	Requisitos	39
6.1.2	Melhorias e Otimizações	40
6.1.3	Banco de Dados	45
6.2	Aplicação Web	46
6.2.1	Estrutura da Aplicação Web	47
6.2.2	Banco de Dados	57
6.3	Integração via <i>Web Services</i>	57
6.4	Considerações Finais	59
7	Conclusão e Trabalhos Futuros	60
7.1	Conclusão	60
7.2	Trabalhos Futuros	61
	Referências	62

Lista de Figuras

2.1	Número de Indivíduos com Diabetes pelo Mundo [33].	6
2.2	Projeção de Crescimento da Diabetes até o Ano de 2025. [28]	7
3.1	Arquitetura do Android [4].	18
3.2	Distribuição de Uso por Versões [4].	21
4.1	Arquitetura Orientada a Serviços [23].	26
4.2	Mecanismo de Registro e Descoberta da Arquitetura Orientada a Serviço [23].	27
5.1	Tabela de Comparação entre os Aplicativos [16].	35
5.2	Modelagem do Banco de Dados Utilizado no Aplicativo [16].	38
6.1	Tela de <i>Login</i>	40
6.2	Tela de Últimos Dados.	41
6.3	Tela de <i>Backup</i>	41
6.4	Tela de Exercícios.	42
6.5	Tela de Glicemia.	42
6.6	Tela de Refeição.	43
6.7	Tela de inclusão de Refeição.	43
6.8	Tela com Lista de Opções.	44
6.9	Tela de Relatórios.	44
6.10	Tela de Relatórios por Tipo de Glicemia.	45
6.11	Mudanças no Banco de Dados do Aplicativo.	46
6.12	Tela de <i>Login</i>	47
6.13	Tela de Boas Vindas.	48
6.14	Tela de Exercícios.	49
6.15	Tela de Insulinas.	49
6.16	Tela de Glicemias.	50
6.17	Tela de Refeições.	50
6.18	Tela do Administrador.	51
6.19	Tela do Administrador para Consulta de Alimentos.	52
6.20	Tela do Administrador usando Filtro de Dados.	52
6.21	Tela do Administrador para Consulta de Exercícios.	53
6.22	Tela do Administrador para Consulta de Insulinas.	53
6.23	Tela do Administrador para Consulta de Glicemias.	54
6.24	Tela do Administrador para Consulta de Refeições.	55
6.25	Tela do Administrador para Cadastro de Usuários.	55

6.26 Tela do Administrador para Alteração de Usuários.	56
6.27 Tela do Administrador para Listagem dos Usuários.	56
6.28 Banco de Dados Utilizado na Aplicação Web.	57
6.29 Tabela para sincronização offline.	58

Lista de Tabelas

2.1	Diferença entre Hiperglicemia e Hipoglicemia.	9
3.1	Venda Mundial de <i>Smartphones</i> para Usuários por Sistema Operacional em 2015 (Milhares de unidades) [5].	16
3.2	Versões do Sistema Operacional Android.	20

Capítulo 1

Introdução

O diabetes *mellitus* tipo 1, ou DM1, é uma doença que ocorre inicialmente em jovens e crianças e que tem como principal característica a destruição das células betapancreáticas produtoras de insulina. Insulina é o hormônio que promove a absorção de glicose pelas células do corpo [30]. Sua produção em pouca escala acarreta vários problemas para o indivíduo, entre eles, o de maior risco é o armazenamento excessivo de açúcar no sangue. [13]. Existe uma variedade da diabetes *mellitus* tipo 1 cuja instalação é mais lenta encontrada em indivíduos mais velhos, que tem sido referenciada como LADA (*Latent Autoimmune Diabetes in Adults*) [30].

O tratamento da doença consiste, principalmente, no monitoramento da glicemia, no consumo de uma alimentação balanceada, na realização de atividades físicas e na administração de doses de insulina artificial. Por isso, hoje em dia, uma das grandes e massivas atividades dos portadores de diabetes é a responsabilidade de transcrever todo o seu cotidiano em anotações para controle e acompanhamento da doença, armazenando dados sobre sua alimentação, explicitando cada alimento que foi consumido, anotando as práticas de atividade física e realizando as medidas e controle das doses de insulina. Esses dados ajudam tanto o paciente quanto a equipe médica responsável pelo tratamento, que usam esses dados para análise e observação, ou seja, informações relevantes para que a equipe consiga controlar e minimizar as consequências da doença.

Todavia, constatou-se que não existem muitas opções de ferramentas que otimizem essas anotações, onde, por muitas vezes, é feito no papel e a caneta, tornando-se uma atividade de difícil execução, monótona e que consome muito tempo, visto que as anotações são diárias. Outro problema que esse cenário enfrenta, é que nem sempre esse material se encontra disponível, de forma que os pacientes se esqueçam de fazer as anotações ou as façam de maneira imprecisa.

Assim, para a facilitação desse processo, foi idealizado o desenvolvimento de um aplicativo para *Android*, tendo em mente que o uso de *smartphones* e de *tablets* cresceu exponencialmente nas últimas décadas, criando um enorme leque de possibilidades para que atividades cotidianas como essas sejam feitas de maneira menos árdua e mais prática, suavizando a vida desses pacientes. Nesse cenário, foi desenvolvido em 2013 o aplicativo Doce Desafio [16], o qual objetiva acompanhar o tratamento de um diabético tipo 1 e auxiliar no uso das doses diárias de insulina.

Contudo, com muitos usuários realizando suas anotações e armazenando os dados para controle, observou-se um relativo aumento no uso do aplicativo, a partir desse crescimento,

do *feedback* e entrevistas de alguns usuários, percebeu-se a necessidade de atualização do software, seja pela melhoria de alguns recursos do aplicativo, seja pela criação de novas funções. Além disso, foi levantada pela equipe de acompanhamento do tratamento (educadores físicos, enfermeiros, nutricionistas, técnicos, médicos e outros) a necessidade de ampliar o aplicativo Doce Desafio para uma plataforma web, na qual fosse possível concentrar e armazenar os dados de todos os pacientes do Brasil inteiro.

Diante do exposto, este trabalho visa disponibilizar essas melhorias, que vão desde mudanças estruturais no aplicativo para melhor usabilidade, até a criação de novas funções que ajude no controle dos dados. O trabalho propôs também a ampliação das funções do aplicativo para a web, desenvolvendo um sistema web que facilite o acesso aos dados do aplicativo, que agora pode ser feito por qualquer computador com acesso a internet. A tecnologia usada para desenvolver essa comunicação com o aplicativo foi *web services*.

1.1 Problema

O aplicativo Doce Desafio, criado em 2013 como resultado de um trabalho de conclusão de curso dos alunos Edison Carvalho e Jair Fernandes da Universidade de Brasília [16], destinado ao controle da saúde de pessoas com diabetes tipo 1, necessitou da implementação de novas funcionalidades para atender a demanda de seus usuários e também de proporcionar uma maneira mais acessível para o acompanhamento do tratamento pelos profissionais de saúde.

1.2 Justificativa

O aplicativo foi criado com funcionalidades que inicialmente atendiam a expectativa de seus usuários, porém com o decorrer do tempo e uso do aplicativo, novas necessidades surgiram, tais como a possibilidade de que os dados registrados e monitorados pelo próprio usuário pudessem ser monitorados pelos profissionais de saúde que acompanhavam seu tratamento, independentes da localização física do paciente e/ou da equipe de acompanhamento.

1.3 Objetivo

O objetivo deste trabalho é implementar melhorias e otimizações no aplicativo Doce Desafio, criado para dispositivos móveis, com o intuito de auxiliar indivíduos diagnosticados com diabetes tipo 1, assim como, desenvolver uma aplicação web que permita aos profissionais de saúde realizar o acompanhamento *online* do paciente.

1.3.1 Objetivos Específicos

Os objetivos específicos a serem alcançados neste trabalho são:

- Expandir as opções de registro de dados no aplicativo, permitindo mais detalhes de informações;

- Expandir opções de *backup* permitindo a geração de arquivos *CSV* (*Comma Separated Values*) a partir dos dados inseridos;
- Criar uma aplicação web que possibilite o acompanhamento *online* dos dados inseridos pelos usuários; e
- Criar um mecanismo de sincronização dos dados via *web services* entre o aplicativo e a aplicação web.

1.4 Resultados Esperados

Espera-se que esse aplicativo, validado junto aos especialistas em educação física, nutrição e saúde, auxilie os usuários portadores de diabetes, beneficiando-os no controle de sua saúde e propiciando um recurso adicional para o acompanhamento de seu tratamento, fundamental para a manutenção da saúde e a qualidade de vida desses usuários. Espera-se também que, com o desenvolvimento da aplicação web, os dados registrados no aplicativo possam ser acessados de maneira mais prática pelos profissionais de saúde, resultando em um acompanhamento mais próximo e periódico de seus pacientes.

1.5 Metodologia

A metodologia para a realização do trabalho proposto começou com o estudo bibliográfico a respeito da Diabetes *mellitus* tipo 1 e compreensão do aplicativo Doce Desafio e sua utilidade.

Em seguida, dedicou-se esforço para entender sobre a diabetes mellitus ou diabetes tipo 1, dessa seria possível entender quais as necessidades e as dificuldades que esses indivíduos possuem para então ajuda-los.

A terceira fase focou no estudo da linguagem de programação para criação de aplicativos no sistema *Android*, sua sintaxe e semântica, assim como quais IDEs (Integrated Development Environment) o suportavam.

A fase seguinte foi constituída por encontros com os responsáveis pelo projeto Doce Desafio, assim como com usuários do aplicativo, para entender e colher sugestões de melhoria. Também foram realizadas reuniões com usuários do aplicativo para colher requisitos a fim de desenvolver a aplicação web e entender como essa deveria funcionar.

Em seguida, iniciou-se a verificação do código fonte do aplicativo, procurando entender e identificar os padrões de projeto utilizados, a lógica usada na construção das classes e dos métodos, assim como a disposição das telas.

Posteriormente, começou a fase de implementação das funcionalidades levantadas pelos usuários e equipe técnica. Com as telas e as funcionalidades operando conforme esperado, os esforços foram direcionados para viabilizar a comunicação entre o aplicativo e o sistema web, por meio de *web services*.

1.6 Organização do Trabalho

Este trabalho está dividido em 8 capítulos. O presente capítulo introduz o trabalho proposto e apresenta sua estruturação.

No Capítulo 2 é exposto o resultado de um levantamento teórico sobre o diabetes, suas causas, distribuição no Brasil e no mundo.

No Capítulo 3 é apresentado as etapas do desenvolvimento de um software, o sistema operacional para dispositivos móveis *Android*, bem como seu histórico, arquitetura, funcionamento e desenvolvimento de aplicativos *Android*. Por fim, é apresentado um levantamento teórico a respeito do desenvolvimento de aplicações web com a linguagem Java.

No Capítulo 4 são apresentados os conceitos de *SOA (Service Oriented Architecture)* e *web services*, que serviram como base para o desenvolvimento deste trabalho.

No Capítulo 5 é apresentado o aplicativo Doce Desafio, cujo desenvolvimento é fruto de um Projeto de conclusão de curso da Universidade de Brasília e sobre o qual este trabalho é baseado.

No Capítulo 6 é apresentada a implementação prática proposta neste trabalho, que foi a melhoria e a otimização do aplicativo Doce Desafio e o desenvolvimento de uma aplicação web, que se integra ao aplicativo via *web services*.

No Capítulo 7 são apresentadas as considerações finais sobre o trabalho realizado, sugestões de melhorias e indicações para trabalhos futuros.

Capítulo 2

Diabetes

Neste capítulo são apresentados conceitos sobre a diabetes *mellitus* tipo 1 e de tipo 2, suas características, sintomas e causas, assim como as principais dificuldades que os portadores da doença encontram na sua rotina.

2.1 Definição

O diabetes *mellitus* se caracteriza como uma doença crônica, ou seja, uma doença difícil de ser tratada efetivamente em um curto período de tempo. Apenas para constatar a doença faz-se necessário um diagnóstico de três meses. No entanto, doenças crônicas são classificadas como doenças de baixa periculosidade por não provocar grandes riscos, no curto prazo, para a vida do paciente, portanto, clinicamente, não se entende doenças crônicas como emergências médicas. Outra característica de doenças crônicas é a existência de um sintoma contínuo, o que afeta diretamente a qualidade de vida e a rotina dos pacientes diagnosticados com essa categoria de doença, mesmo que a saúde física do indivíduo não esteja posta em risco.

O primeiro caso constatado de diabetes foi no Egito 1500 anos a.c.. O fato de ter sido, até então, sua primeira aparição, levou a doença a ser considerada como desconhecida, assim não possuindo um nome em específico. Contudo, somente 250 anos depois, com os pesquisadores Memphis e Apolonio, que surgiu a ideia para o nome, a palavra escolhida pelos doutores foi "diabetes" e sua denominação tem base ontológica do Grego que significa sifão (tubo para aspirar a água), a explicação pela escolha do nome veio da sintomologia (parte da Patologia que tem por objeto o estudo dos sintomas das doenças), pois dois dos sintomas fortemente presentes nos indivíduos portadores da doença são a intensa sede e a grande quantidade de urina que é expelida. A variação da doença, que teve a terminologia *Mellitus*, surgiu somente no século I d.c.. *Mellitus* por sua vez, tem origem no latim, significa mel, assim a patologia passou a ser chamada de urina doce [19].

Estima-se que 0,3% da população mundial seja acometida por diabetes *mellitus* tipo 1, sendo sua incidência maior na Escandinávia e na região nordeste da Europa, nas quais são encontrados cerca de quarenta casos por cem mil habitantes na faixa etária inferior a 14 anos [30]. No restante da Europa esse número diminui para cerca de dez casos para cem mil habitante, nos Estados Unidos ocorrem cerca de quinze casos por cem mil habitantes. A menor incidência de diabetes tipo 1 é na China e em partes da América do Sul, com um caso por cem mil habitantes. No Brasil, o diabetes atinge 9 milhões de brasileiros –

o que corresponde a 6,2% da população adulta. As mulheres (7%) apresentaram maior proporção da doença do que os homens (5,4%) [7].

Graças à evolução da medicina e da tecnologia que juntas descobriram uma maneira de disponibilizar a insulina, uma enorme gama de pacientes vem assistindo uma maneira de examinar parcialmente a doença, pois apesar da insulina fornecer um meio de monitorar a doença, essa não é a cura, visto que, existem as complicações secundárias da doenças que afetam o sistema endócrino (endocrinopatia). Além disso, os pacientes passam por um regime alimentar altamente restrito [32].

Para se ter um melhor controle da doença e assegurar que o paciente viva com menos riscos e de forma mais saudável, é exigido do paciente, sua família e dos profissionais da saúde um enorme esforço, com o intuito de amenizar os sérios riscos que a doença acarreta.

Esses esforços em conjunto tem a finalidade de promover qualidade de vida ao paciente, porém uma das tarefas mais árduas desse individuo é se atentar e controlar sua rotina, que contempla tarefas como os regimes de insulina, as rigorosas dietas e a rotina de exercícios físicos. Em várias pesquisas voltadas para a análise do impacto da doença nos lares das famílias com crianças ou adolescentes portadores da doença crônica, tem-se dado ênfase na figura materna, pois como averiguado, as progenitoras são as que mais os acompanham, auxiliam e vivenciam as rotinas diárias dos portadores da doença [14].

2.2 Estatísticas

Segundo estudos feitos e levando para projeções mundiais, McCarty e Zimmet [1], levantaram que em 2000 teríamos 175 milhões de diabéticos em todo o mundo, já em 2010 este índice aumentaria para 236 milhões. A Figura 2.1 ajuda a observar como a doença está espalhada pelo mundo, e a estimativa de pessoas que sofrem com ela.

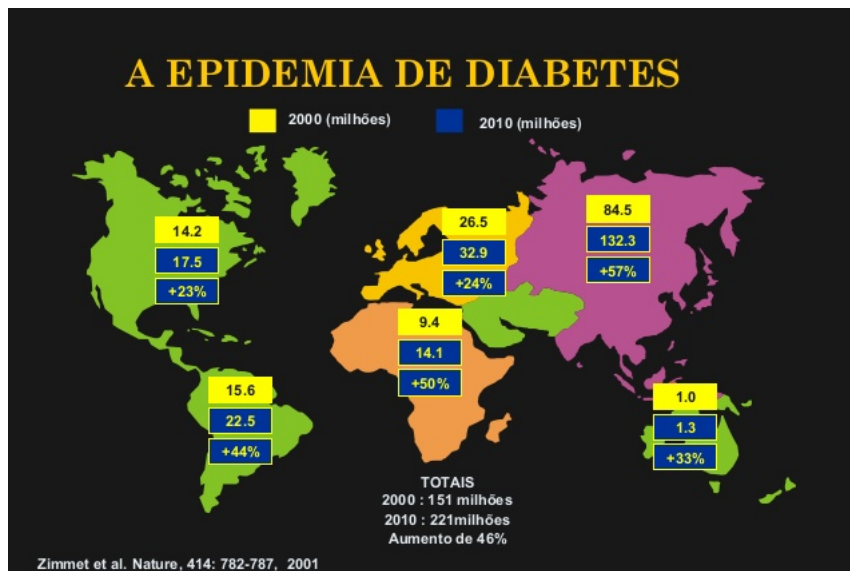


Figura 2.1: Número de Indivíduos com Diabetes pelo Mundo [33].

Ao redor do mundo existem muitos institutos e organizações que visam estudar e verificar estes índices, para observar a prevalência e a incidência do diabetes. Um deles é o

Projeto Multinacional para o Diabetes na Infância, que vem revelando um aumento significativo na frequência do diabetes tipo 1 [19]. Como uma tentativa de alertar a população e tentar controlar a doença, várias campanhas nacionais do Ministério da Saúde realizaram, em 2001, 21 milhões de glicemias capilares e detectaram 3 milhões de suspeitos de diabetes, o que equivale a 7% da população pesquisada.

Entre suas variações, destacam-se duas, que são as diabetes de tipo 1 e de tipo 2. Apesar de suas diferenças na sintomatologia (estudo dos sintomas da doença) e tratamento, elas se diferenciam no perfil da população que sofre com a doença. A diabetes *mellitus* tipo 1 atinge crianças e adolescentes e é uma das doenças crônicas de maior relevância e estudo dirigido no Mundo. Nos Estados Unidos, dos 651.000 novos casos de doenças que os médicos diagnosticaram em seus consultórios, 11.000 são em crianças e adolescentes, colocando a doença na impressionante posição de segunda maior doença crônica infantil em aparições neste país [18].

Por sua vez, a diabetes do tipo 2 atinge principalmente a população entre 30 e 69 anos, porém podendo ser também observado em crianças e adolescentes devido a obesidade e ao sedentarismo infantil. Esse tipo constitui cerca de 90% dos casos de diabéticos, por exemplo, só na China, estima-se que aproximadamente 10 milhões de pessoas possuem a doença, sendo uma das justificativas para o aumento no número de afetadas a ocidentalização da população.

Segundo a OMS (Organização Mundial da Saúde), estima-se que cerca de 5% da população mundial entre 20-79 anos de idade possua a diabetes tipo 2, esse número demonstra uma crescente, tanto nos países desenvolvidos quanto nos industrializados. [6]

Somente no Brasil, é estimado que 5,2% de toda a população sofra com a doença. Projeções feitas pela Federação Internacional de Diabetes prevê que os números apresentados dobrem até 2025, como mostrado na Figura 2.2. Esse número é preocupante pelo fato da maioria dos indivíduos que possuem a doença desconhecerem que são afetados por esta. Nos Estados Unidos, um estudo feito em 2014, mostrou que cerca de 40% da população adulta, ou seja, 2 em cada 5 americanos, iria desenvolver a diabetes tipo 2 [25].

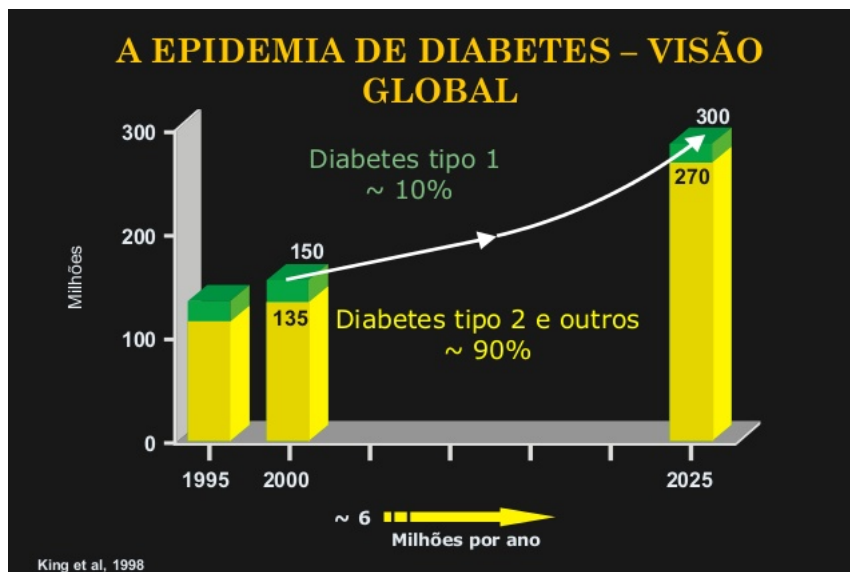


Figura 2.2: Projeção de Crescimento da Diabetes até o Ano de 2025. [28]

2.3 Diagnóstico

Toda a alimentação que consumimos passa por um processo de digestão, esse processo tem a finalidade de quebrar o conteúdo ingerido em partes menores que resultarão em energia para o organismo. Nessa sequência, a glicose é formada, absorvida pelo sangue e é levada por esse até os tecidos, posteriormente repondo as células com energia. O que metaboliza essa glicose em energia dentro da célula é a insulina, que age aumentando a permeabilidade celular, permitindo que a glicose seja transformada em energia, fazendo com que as funções do corpo continuem a funcionar.

Descoberto na Alemanha pelo estudante de medicina Paul Langerhans [32], existem dois grupos de células do pâncreas, que são circundadas por vasos sanguíneos, onde ocorre a produção de dois hormônios, os quais são a insulina e o glucagon. A partir desses hormônios é possível a síntese dos hormônios pancreáticos, as células alfa, que produzem o glucagon, e as células betas, responsáveis pela síntese de insulina.

A insulina é um hormônio hipoglicemiante, ou seja, ele facilita a absorção da glicose existente no sangue em diversos tipos de células, como as células musculares e as do fígado, onde essas moléculas são armazenadas sob a forma de substância de reserva.

Quando o pâncreas possa a produzir uma quantidade menor de insulina, ou se diminui a capacidade de utilização dessa, o processo de sintetização das moléculas, ou também chamado de digestão, não pode ser realizado, assim, a quantidade de glicose que está no sangue tende a aumentar, pois o ingresso dessas na célula fica impossibilitado pela falta de insulina, logo não há a absorção de açúcar pela célula, causando o que é chamado de hiperglicemia, sendo essa um das principais características da diabetes [13]. Os indivíduos com hiperglicemia, ou a elevação da taxa de glicose no sangue, vivenciam a aparição de vários sintomas e sinais, sendo alguns deles [26]:

- Aumento do apetite;
- Maior produção de urina, resultando em desidratação e sede;
- Alteração da visão;
- Perda de peso, essa ocorrendo até em obesos, mesmo que o individuo mantenha uma alimentação desbalanceada e em excesso;
- Impotência sexual;
- Infecções provocadas por fungos na pele e nas unhas.
- Feridas que demoram a cicatrizar, principalmente, nos membros inferiores;
- Distúrbios cardíacos e renais;
- Neuropatias diabéticas, causadas pelo comprometimento das terminações nervosas.

Outra variação que pode ocorrer pelo erro no uso da medicação, principalmente a insulina, pelo atraso em se alimentar e excesso de exercício físico sem monitoramento, é a hipoglicemia, ou seja, a baixa taxa de insulina no corpo. Esse por sua vez tem alguns sintomas que são agrupados como ligeiros, ou menos graves, e em grupos de sintomas graves. Os sintomas ligeiros podem ser [26]:

- Tremor;
- Ansiedade;
- Fome aguda;
- Suor frio;
- Tontura;
- Sonolência.

No grupo de sintomas graves, passam a ser [26]:

- Indisposição;
- Visão dupla ou embaçada;
- Confusão mental;
- Dificuldade na respiração;
- Descoordenação motora;
- Convulsão.

Para que seja vivida uma vida sem muitas complicações, estima-se que a concentração de glicose no sangue, ou glicemia, esteja em uma faixa que oscila entre 70 e 140mg/dL (miligramas por decilitro), essa faixa pode variar de pessoa para pessoa. A hipoglicemia é identificada quando se analisa uma glicemia abaixo de 70mg/dL [10]. Algumas das principais diferenças entre a hiperglicemia e a hipoglicemia (diminuição de glicose no sangue) estão relatadas na Tabela 2.1.

Tabela 2.1: Diferença entre Hiperglicemia e Hipoglicemia.

Sintomas	Hiperglicemia	Hipoglicemia
Início	Lento	Súbito
Sede	Muito	Inalterado
Urina	Muita Qualidade	Inalterado
Fome	Muita	Muita ou Normal
Perda de Peso	Frequente	Não
Pele	Seca	Normal ou Úmida
Mucosa da Boca	Seca	Normal
Suores	Ausente	Frequente e Frios
Tremores	Ausente	Frequentes
Fraqueza	Ausente	Sim ou Não
Cansaço	Presente	Presente
Glicemia	Superior a 200 mg/dL	Inferior a 70 mg/dL

2.4 Causas e Sintomas

A principal causa da diabetes *mellitus* se dá pela elevação da glicose (açúcar) presente no sangue do indivíduo, quando diagnosticada e constatado a extrapolação do limite da taxa considerada normal (hiperglicemia) [10].

A doença pode ocorrer por fatores genéticos, ou seja, a pessoa que é gerada no ventre de um portador de diabetes, tem a possibilidade de desenvolvê-la, e/ou fatores ambientais, quando o indivíduo se alia a um cenário de sedentarismo, alimentação inadequada, estresse emocional, usos de medicamentos diabetogênicos, obesidade e problemas com o sono, acelerando a probabilidade da doença surgir mais cedo. Existem outros fatores que acarretam o diabetes, alguns exemplos são o excesso de doces e açúcar, a menopausa, a ingestão demasiada de certos medicamentos, cirurgias, estresse ou uma alimentação rica em carboidratos [19].

Juntamente com as causas, a doença apresenta alguns sintomas que necessitam de uma atenção especial, pois podem ser sinais da doença. Alguns dos principais sintomas são [26]:

- Muita sede (polidipsia);
- Excesso de urina (poliúria);
- Muita fome (polifagia);
- Emagrecimento.

Porém, podendo apresentar outros sintomas como:

- Sonolência;
- Dores generalizadas;
- Formigamentos e dormências;
- Cansaço doloroso nas pernas;
- Câimbras;
- Nervosismo;
- Indisposição para o trabalho;
- Desânimo;
- Turvação da visão;
- Cansaço físico e mental.

Caso não haja um controle adequado da doença, a probabilidade é muito alta dessa criar grandes complicações futuramente, causando dificuldades na visão (retinopatia), problemas nos rins (nefropatia) e problemas neurológicos (neuropatia). Dessa forma, alguns distúrbios que podem ser constatados são:

- Cefaléia (dor de intensidade leve a moderada na região da cabeça);

- Inquietude;
- Irritabilidade;
- Palidez;
- Sudorese (Transpiração);
- Taquicardia;
- Confusões mentais;
- Desmaios;
- Convulsões;
- Coma (Caso extremo).

2.5 Diabetes Tipo 1 (DM1)

A diabetes tipo 1 (DM1) é uma doença autoimune, ou seja, o organismo destrói as células beta produtoras de insulina por engano. Assim a doença surge quando o organismo deixa de produzir quantidades suficientes de insulina [10].

O diagnóstico da doença pode ser feito por um exame de glicemia, que significa a quantidade de açúcar presente no sangue [9], sendo necessário o paciente estar em jejum de 8 horas. Em pessoas não portadoras da doença, a taxa de glicose varia entre 70 mg/dL e 140 mg/dL em jejum e duas horas após a refeição pode ir até 138 mg/dL. Quando esses valores passam a medir 100 a 125 mg/dL em jejum e após duas horas da refeição esse valor se encontrar entre 140 a 199mg/dL, é um sinal de alerta da doença, sendo necessário a procura de um corpo clínico para maiores orientações.

Não se sabe ao certo a causa da doença, mas pesquisas demonstram que a pré-disposição genética é uma forma de contrair a diabetes, contudo não sendo uma regra, existindo indivíduos que possuem os genes das doenças, porém não a desenvolvem. Em geral a doença se desenvolve em pessoas com menos de 35 anos, mas pode surgir em qualquer idade.

O tratamento da diabetes tipo 1, traduz-se em injeções diárias de insulina, para controlar os níveis de açúcar no sangue, assim como uma alimentação balanceada e a prática de atividades físicas. O controle desses aspectos facilitam o controle da doença [25].

2.6 Diabetes Tipo 2 (DM2)

É a diabetes mais comumente encontrada, em que não se faz necessário o uso da insulina inicialmente. Sua aparição ocorre normalmente na faixa etária acima dos 30 anos de idade, sendo mais constante nos indivíduos entre 50 e 60 anos. No entanto, por conta dos maus hábitos alimentares e sedentarismo, se constatou uma alta nos índices de crianças e adolescentes obesos ou com sobrepeso diagnosticadas com diabetes.

Sua aparição ocorre em 90% dos casos constatados de diabetes *mellitus*, e sua causa pode ser atrelada tanto à produção e secreção insuficiente da insulina produzida pelo

pâncreas quanto por problemas nos receptores, dificultando a sua utilização. Quando há esse distúrbio, recebe o nome de resistência insulínica [30].

Estudos têm mostrado que existe um menor número de receptores de insulina em pessoas obesas comparadas com as pessoas de pesos normais. Mas sabe-se que essa resistência tem sido causada por alterações na via de sinalização desencadeada pelo hormônio. Na maioria dos portadores do diabetes tipo 2, não existe uma incidência que pudesse ter contribuído ou gerado a doença, que se desenvolve de maneira gradual no decorrer dos anos. Esta doença ocorre quando a ação da insulina torna-se insuficiente para manter os níveis plasmáticos da glicose [20].

Com a resistência à insulina o organismo acumula a glicose plasmática. Como uma tentativa de regularizar o excesso, o corpo expulsa a quantidade desnecessária de glicose pela urina o que faz aumentar ainda mais a produção de insulina. Se não tratada, esse processo em longo prazo pode deixar as células betas esgotadas, o que as tornaria incapazes de produzir quantidades suficientes para manter a regularização. O controle do diabetes tipo 2 baseia-se em uma dieta alimentar regrada, com o uso de medicamentos orais ou injetáveis.

2.7 Considerações Finais

Abordou-se nesse capítulo características, causas e sintomas da diabetes tipo 1 e tipo 2. Definiu-se também suas diferenças e seus efeitos nos indivíduos, assim como as principais dificuldades que os portadores da doença enfrentam. Com isso, é possível entender melhor as necessidades dos indivíduos que possuem a doença e promover ferramentas que os auxiliem na sua rotina.

Como observado por Edison Carvalho e Jair Rodrigues [16], notou-se um crescimento do número de aparelhos *Android* vendidos no mundo, sendo esse um grande potencial para ajudar a solucionar esse problema. Assim surgiu o aplicativo Doce Desafio, que tem o intuito facilitar o controle da doença por meio de anotações a respeito da rotina alimentícia, atividades físicas e aplicação de insulina. Por isso, para melhor entendimento desse sistema operacional para *smartphones* e *tablets*, dedicamos o próximo capítulo para a explicação teórica a respeito da plataforma *Android*.

Capítulo 3

Desenvolvimento de Aplicativos *Android* e Aplicações Web

Este capítulo apresenta os conceitos envolvidos na criação de um programa, o processo de desenvolvimento de um software, as definições arquiteturais do projeto, assim como, estabelece um entendimento sobre o sistema operacional para dispositivos móveis *Android* e o desenvolvimento de aplicativos para esta plataforma. Por fim, aborda o desenvolvimento de aplicações web com a linguagem Java.

3.1 Desenvolvimento de Software

Atualmente, a elaboração de um programa de computador requer a utilização de diversos conceitos e metodologias de desenvolvimento, em grande parte, provenientes da Engenharia de Software. Os fundamentos científicos encontrados nessas metodologias envolvem o uso de modelos abstratos que permitem ao desenvolvedor garantir a qualidade do programa gerado.

3.1.1 Processo de Software

Segundo Sommerville [36], um processo de software é um conjunto de atividades que leva a produção de um produto de software. Isso significa dizer que o processo de software abrange as atividades necessárias para o desenvolvimento, desde a criação até a etapa de manutenção. Quatro atividades são elencadas como fundamentais a esse processo: especificação, desenvolvimento, validação e evolução de software.

- Especificação de software: As funcionalidades do software e as restrições sobre sua operação devem ser definidas;
- Projeto e implementação de software: Um software que esteja de acordo com as especificações acordadas deve ser produzido;
- Validação de software: O software deve ser validado junto aos seus usuários para garantir que ele atenda as suas necessidades;
- Evolução de software: O software deve evoluir para atender as necessidades mutáveis de seus usuários.

Essas atividades estão presentes na maioria dos processos de software, porém diferentes tipos de sistemas necessitam de diferentes processos de desenvolvimento. Dessa maneira, as mesmas atividades podem ser elaboradas de formas diferentes, dependendo das características do sistema em questão. A consequência disso foi a criação de modelos de processo de software que "são uma descrição simplificada do processo de software que apresenta uma visão dele onde cada modelo de processo representa um processo sob determinada perspectiva"[36].

Modelos de Processo de Software

Um modelo de processo de software é uma representação abstrata de um processo de software. Eles definem um conjunto distinto de atividades, ações, tarefas, marcos e produtos de trabalho que são necessários para fazer engenharia de software com qualidade. Pressman [35] divide os modelos em:

- Modelo em Cascata: Considera cada uma das atividades do desenvolvimento de software como etapas distintas, separadas e sequenciais;
- Modelo Evolucionário: Intercala as diferentes etapas do desenvolvimento de software repetindo as diversas vezes em ciclos;
- Modelo Incremental: Considera cada uma das atividades do desenvolvimento de software como sequências lineares e iterativas.

- Modelo em Cascata

O modelo em cascata, algumas vezes chamado de ciclo de vida clássico, sugere uma abordagem sistemática e sequencial para o desenvolvimento de softwares com a especificação dos requisitos pelo cliente, e progride ao longo do planejamento, modelagem, construção e implantação, resultando na manutenção progressiva do software. As vantagens do modelo em cascata consistem na documentação produzida em cada fase e sua aderência a outros modelos de processos de engenharia. Seu maior problema é a divisão inflexível do projeto em estágios distintos [35].

- Modelo Evolucionário

O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando esse resultado por meio de várias versões até que seja desenvolvido um sistema adequado. As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de serem separadas. Uma abordagem evolucionária para desenvolvimento de software é frequentemente mais eficaz do que a abordagem em cascata na produção de sistemas que atendam as necessidades imediatas dos clientes. A vantagem de um processo de software baseado na abordagem evolucionária é que a especificação pode ser desenvolvida de forma incremental. A medida em que os usuários compreendem melhor seu problema, isso pode ser refletido no sistema de software [35].

- Modelo Incremental

O modelo incremental combina elementos do modelo em cascata aplicado de maneira iterativa. O modelo incremental aplica sequências lineares de uma forma racional a medida que o tempo passa. Quando um modelo incremental é usado, o primeiro incremento é frequentemente chamado de núcleo do produto, isto é, os requisitos básicos são satisfeitos, mas muitas características suplementares deixam de ser elaboradas. O núcleo do produto é usado pelo cliente, um plano é desenvolvido para o próximo incremento como resultado do uso e da avaliação. O plano visa a modificação do núcleo do produto para melhor satisfazer as necessidades do cliente e a elaboração de características e funcionalidades adicionais. Esse processo é repetido após a realização de cada incremento, até que o produto completo seja produzido [35].

Para o desenvolvimento do sistema Doce Desafio, foi utilizado o modelo evolucionário em que uma implementação inicial exposta aos comentários dos clientes ou usuários é refinada até atingir uma versão adequada as suas necessidades e expectativas.

3.1.2 Arquitetura de Software

A arquitetura de software é composta por elementos de software com suas propriedades externamente visíveis e o relacionamento entre eles [15]. Apresenta-se a seguir, alguns dos requisitos considerados importantes em um software e que podem afetar o estilo de sua arquitetura [15].

- Desempenho: Se o desempenho for um requisito crítico, a arquitetura do software deve ser projetada para que haja o menor número de comunicação possível entre os subsistemas que englobam o software, significando o uso de componentes de alta granularidade em detrimento dos de baixa granularidade.
- Proteção: Se a proteção for um requisito crítico, deve-se utilizar um estilo de arquitetura em camadas, com os itens mais críticos protegidos por camadas mais internas com um alto nível de validação.
- Disponibilidade: Se a disponibilidade for um requisito crítico, a arquitetura deve ser elaborada para comportar a utilização de componentes redundantes que permitam a atualização e a modificação do sistema sem a necessidade de parar o sistema.
- Manutenibilidade: Se a facilidade de manutenção for um requisito crítico, a arquitetura deve ser elaborada utilizando componentes de baixa granularidade e auto contidos para que possam ser facilmente modificados.

Modelos de Arquitetura de Software

Segundo Sommerville [36], existem três estilos ou modelos de arquitetura de software, os quais são Modelo de Repositório, Modelo Cliente-Servidor e Modelo em Camadas, distintos a seguir:

- Modelo de Repositório

A principal característica deste modelo é o fato dos dados estarem armazenados em banco de dados, que podem ser acessados por outros subsistemas. Ainda existe a possibilidade de cada sistema possuir seu próprio banco de dados, desta forma as informações são trocadas entre outros subsistemas.

- Modelo Cliente-servidor

Este modelo é constituído de um conjunto de servidores que fornecem serviços e um conjunto de clientes que utilizam esses serviços prestados.

- Modelo em Camadas

Organiza um sistema em camadas na qual cada uma fornece um conjunto de serviços para as camadas acima dela. Uma camada utiliza os serviços fornecidos pela camada abaixo dela, não sendo necessário conhecer a forma como é realizado ou como ocorre, só é necessário uma interface de comunicação entre a camada superior com a camada inferior.

3.2 Plataforma *Android*

Android é o sistema operacional para *smartphones* e *tablets* mais utilizados no mundo. Ele surgiu juntamente com a enorme demanda e oferta de *smartphones*, que nos últimos anos vem sendo o mercado que mais vende no planeta, onde foram vendidos 271 milhões de *smartphones* no ano de 2015, o que corresponde a 82,2% do total de vendas de *smartphones* conforme mostrado na Tabela 3.1. A criação de um sistema operacional com maior capacidade de gerenciamento e novos componentes para celulares decorreu do fato desses começarem a vir mais robustos, complexos e com um hardware mais potente, necessitando de uma forma mais efetiva para gerenciá-lo.

Tabela 3.1: Venda Mundial de *Smartphones* para Usuários por Sistema Operacional em 2015 (Milhares de unidades) [5].

Sistema Operacional	2015 (unidades)	2015 fatia de mercado (%)	2014 (unidades)	2014 fatia de mercado (%)
Android	271.010	82,2	243.484	83,8
iOS	48.086	14,6	35.345	12,2
Windows	8.198	2,5	8.095	2,8
BlackBerry	1.153	0,3	2.044	0,7
Outros	1.229	0,4	1.416,8	0,5
Total	329.676,4	100	290.384,4	100

O *android* nasceu após a observação do mercado emergente de smartphones, quando a Google em julho de 2005, adquiriu a *Android Inc*, uma empresa então sediada em Palo

Alto na Califórnia-EUA, que na época desenvolvia uma plataforma para telefone móvel baseada em Linux, com o objetivo de ser uma plataforma flexível e com maior capacidade de migração entre os *smartphones* de diferentes fabricantes. A Google, inicialmente, pretendia apenas disponibilizar seus serviços de busca para *smartphones*, contudo, em 2007 aconteceu uma aliança entre diversas empresas, entre elas, HTC, Samsung Eletronicas, LG, Motorola, T-Mobile, dentre outras, concretizando a "*Open Handset Alliance*", com o intuito de criar padrões abertos de telefonia móvel, logo depois de sua formação, revelou-se o sistema operacional para smartphones *android* [3].

A plataforma *Android* é um sistema operacional baseado em Java que é executado no *kernel* do Linux composto por uma rica UI (User Interface), aplicativos, bibliotecas de código, *frameworks* de aplicativo, suporte a multimídia entre outros. [12].

Os aplicativos do *Android* são desenvolvidos na linguagem Java, usando o *Android SDK (Software Development Kit)* e podem ser portados com bastante facilidade. *Android* também inclui aceleração 3D, motor gráfico baseado no suporte de hardware, suporte de banco de dados alimentado por *SQLite* e um navegador web integrado. O paradigma é orientação a objeto, baseados em *AndroidXML*, o *layout* da interface do usuário [34].

A principal característica do *Android* é que seu código-fonte é aberto, dessa forma componentes e funcionalidades não disponíveis nativamente podem ser fornecidas pela comunidade global de desenvolvedores sem depender do Google.

3.2.1 *Kernel do Linux*

Projetado sobre a versão 2.6 do *kernel* Linux e desde a versão 4.4 (KitKat) do *Android* sobre a versão 3.10 do *kernel* Linux, para serviços essenciais do sistema, como segurança, gerenciamento de memória, gerenciamento de processos, rede e *drivers*. O *kernel* também funciona como uma camada de abstração entre o hardware do dispositivo e o resto do conjunto de softwares que são desenvolvidos em paralelo. O Linux 2.6 foi escolhido por já conter uma grande quantidade de *drivers* de dispositivos sólidos e por ter um bom gerenciamento de memória e processos. Várias funções do *kernel* são utilizadas diretamente pelo *Android*, mas muitas modificações foram feitas para otimizar memória e tempo de processamento das aplicações. Essas modificações incluem novos dispositivos de *drivers*, adições no sistema de gerenciamento de energia e um sistema que possibilita terminar processos de maneira criteriosa quando há pouca memória disponível.

3.2.2 *Arquitetura*

Em sua arquitetura, os aplicativos de terceiros são executados com a mesma prioridade daqueles que estão junto com o núcleo do sistema, dando flexibilidade para executar aplicações próprias ou modificar os aplicativos nativos. Outro ponto interessante é a possibilidade de acesso a qualquer parte que o sistema operacional gerencie. Em outras palavras, se o desenvolvedor desejar criar um aplicativo que faça discagem, por exemplo, será possível. Além disso, se desejar criar uma aplicação que utilize o GPS interno, também terá acesso.

O potencial para desenvolvedores criarem aplicações dinâmicas e integradas é enorme, além de todos os recursos que estão disponíveis é possível a integração com aplicativos do Google. A Figura 3.1 apresenta em detalhes a arquitetura *Android*.



Figura 3.1: Arquitetura do Android [4].

Como podemos observar na Figura 3.1, a arquitetura *Android* pode ser dividida em 5 níveis, sendo elas:

- *Framework* da Aplicação: É usado na maioria das vezes por desenvolvedores de aplicativos. Desenvolvedor de hardware devem estar cientes das APIs (*Application Programming Interface*), pois muitas mapeiam diretamente para a interfaces HAL (*Hardware Abstraction Layer*) subjacentes e podem fornecer informações úteis de informações de drivers.
- *Binder IPC Proxies*: O mecanismo IPC (Inter-Process Communication) permite que a APIs de alto nível de *framework* da aplicação interajam com os serviços do sistema *Android*.
- Serviços do Sistema *Android*: Funcionalidades expostas pelas APIs se comunicam com os serviços do sistema para acessar o hardware subjacente, o *Android* inclui dois grupos de serviços, serviços do sistema e serviços de media.
- HAL (*Hardware Abstraction Layer*): A HAL define uma interface padrão para os fornecedores de *hardware* implementarem e permite que o *Android* seja agnóstico em relação aos *drivers* da camada do *Kernel* do Linux.

- *Kernel* do Linux: O *Android* usa uma versão do *Kernel* do Linux com algumas adições, por exemplo, um sistema de gerenciamento de memória mais agressivo, o *driver Binder IPC* e outras características para uma plataforma móvel embutida [4].

3.2.3 *Android Runtime*

É baseado no conceito de máquina virtual utilizado em Java. Dadas as limitações dos dispositivos em que o *Android* é executado, baixa memória e processador limitado, não é possível utilizar uma máquina virtual tradicional. Por isso, a Google decidiu criar uma nova máquina virtual.

A primeira máquina virtual utilizada foi a Dalvik, um interpretador que executa apenas arquivos Dex *Dalvik Executable (.dex)*, esse formato é otimizado para o armazenamento eficiente da memória, gerenciamento de *threads (multithreading)*, memória e processos. A ferramenta DX incluída no *Android SDK* transforma as classes (*.class*) compiladas pelo compilador Java na linguagem Dex, dessa forma não é possível executar diretamente o *bytecode*.

Primeiro, a máquina virtual Dalvik combina as classes geradas em Java em um ou mais arquivos Dex, que por sua vez são comprimidos em um único arquivo (*.apk*) (*Android Package*). Desse modo, ela reutiliza informações duplicadas de múltiplos arquivos (*.class*) e reduz pela metade o espaço que um arquivo (*.jar*) ocuparia [34].

Após o lançamento do *Android 4.4 KitKat*, a Google resolveu embarcar uma nova máquina virtual no seu sistema operacional móvel, a ART (*Android Runtime*), que promete tornar a execução de apps muito mais eficiente. Após o *Android 5*, o ART sairá da sua fase de testes e substituirá efetivamente a máquina virtual Dalvik.

O ART foi concebido para usar uma técnica de compilação chamada AOT (*Ahead of Time*), que ocorre antes da execução do aplicativo, eliminando esta etapa durante o tempo de execução. Diferente em relação ao Dalvik que utiliza a técnica JIT (*Just in Time*) que ocorre durante a execução. Com isso, espera-se um aumento de velocidade de execução de até 2 vezes em relação ao Dalvik [17].

Os aplicativos serão executados mais rápido, mas também ocuparão mais memória. Como o código do aplicativo é compilado na instalação e depois mantido na memória ao invés de ser compilado sempre que aberto, isso significa espaço extra sendo ocupado. Em compensação, o processador irá trabalhar menos quando os aplicativos forem abertos, o que leva a menor consumo de bateria. A desvantagem é que sempre que iniciar o dispositivo ou um aplicativo pela primeira vez levará um tempo bem maior.

3.2.4 Versões

O sistema operacional *Android* passou por diversas evoluções desde a sua primeira versão, lançada no ano de 2008. Na Tabela 3.2 são apresentadas todas as versões do Android e a Figura 3.2 apresenta a relação de uso por versão. Quando foi desenvolvido, o aplicativo Doce Desafio foi projetado para ser suportado a partir da versão 2.3 (*Gingerbread*) do Android, devido ser a versão de maior popularidade na época. Atualmente, como mostra a Figura 3.2, a versão 4.4 (*KitKat*) é a mais utilizada.

Tabela 3.2: Versões do Sistema Operacional Android.

Logo	Versão	API	Data de Lançamento
	Android 1.0	Nível 1	23 de Setembro de 2008
	Android 1.1	Nível 2	9 de Fevereiro de 2009
	Android 1.5 (Cupcake)	Nível 3	30 de Abril de 2009
	Android 1.6 (Donut)	Nível 4	15 de Setembro de 2009
	Android 2.0 a 2.1 (Eclair)	Níveis 5, 6 e 7	26 de Outubro de 2009
	Android 2.2 a 2.2.3 (Froyo)	Nível 8	20 de Maio de 2010
	Android 2.3 a 2.3.7 (Gingerbread)	Níveis 9 e 10	6 de Dezembro de 2010
	Android 3.0 a 3.2.6 (Honeycomb)	Níveis 11, 12 e 13	22 de Fevereiro de 2011
	Android 4.0 a 4.0.4 (Ice Cream Sandwich)	Níveis 14 e 15	19 de Outubro de 2011
	Android 4.1 a 4.3.1 (Jelly Bean)	Níveis 16, 17 e 18	9 de Julho de 2012
	Android 4.4 a 4.4.4 (KitKat)	Nível 19	15 de Janeiro de 2014
	Android 5.0 a 5.1.1 (Lollipop)	Nível 21	3 de Novembro de 2014
	Android 6 (Marshmallow)	Nível 23	5 de Outubro de 2015

Versão	Apelido	API	Distribuição
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.9%
4.1.x	Jelly Bean	16	10.0%
4.2.x		17	13.0%
4.3		18	3.9%
4.4	KitKat	19	36.6%
5.0	Lollipop	21	16.3%
5.1		22	13.2%
6.0	Marshmallow	23	0.5%

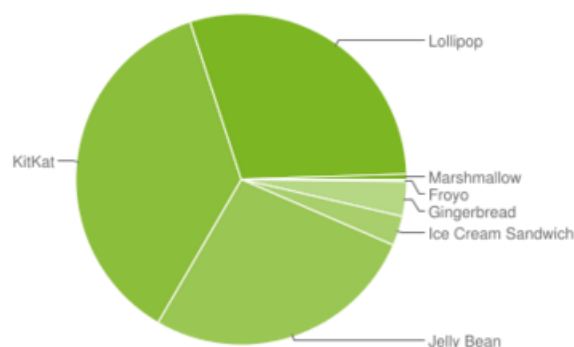


Figura 3.2: Distribuição de Uso por Versões [4].

3.2.5 Desenvolvimento para *Android*

O desenvolvimento para *Android* é baseado na linguagem de programação Java, devido grande parte das bibliotecas Java serem suportadas no Android. No entanto, existem algumas diferenças importantes, apresentadas a seguir:

- Um dos conceitos chave para o desenvolvimento de aplicativos *Android* é o entendimento do ciclo de vida de um aplicativo. Com este conceito é possível entender o funcionamento de um aplicativo e realizar alterações para que ele ocorra como desejado. O desenvolvedor não tem controle sobre as mudanças de estado de uma aplicação, isto é responsabilidade do sistema operacional, no entanto, ele pode ser notificado quando um estado está prestes a mudar, através de métodos que são executados pela linguagem Java. Modificando esses métodos, o desenvolvedor garante a execução das atividades desejadas no momento apropriado.
- Ao contrário do Java, aplicativos *Android* não possuem uma função principal, que é executada no início e chama outras funções em tempo de execução. Eles têm funções como *onCreate*, *onResume*, *onPause* e *onDestroy* que são executadas de acordo com os eventos realizados pelo usuário e devem ser substituídas pelos desenvolvedores para desempenhar as funções desejadas. A função *onCreate*, por exemplo, é executada assim que uma tela é iniciada. A função *onResume* é chamada quando é iniciada a interação com o usuário; *OnPause* é chamada quando o sistema está perto de iniciar uma nova tela ou então para gravar informações que ainda não foram salvas, e a *onDestroy* é chamada quando uma tela foi encerrada ou o sistema precisa

liberar recursos. Além dessas, existem outras funções que se comportam seguindo o mesmo modelo.

- Códigos Java são compilados para *bytecode* Java e, posteriormente, executados na Máquina Virtual Java (*Java Virtual Machine*), enquanto códigos *Android* são executados na Máquina Virtual ART.

A plataforma da Google permite combinar todo o poder da linguagem Java somado às APIs específicas do Android. É possível utilizar o GPS, interferir em chamadas, acessar a câmera, usar o acelerômetro, entre muitos outros recursos que possibilitam a criação de aplicações completas.

Contudo, é importante ressaltar que não somente no campo dos *smartphones* o Android tem obtido êxito. *Tablets*, *netbooks*, máquinas digitais, televisão e até mesmo carros que usam a plataforma Android estão surgindo, fazendo com que ela atinja mercados onde o Java *mobile* não conseguiu grande aceitação.

3.3 Desenvolvimento Web com Java

Java é uma linguagem de programação completa, adequada para o desenvolvimento de aplicações baseadas na Internet, redes fechadas ou ainda programas *stand-alone*. Foi desenvolvida na primeira metade da década de 90 nos laboratórios da Sun Microsystems com o objetivo de ser mais simples e eficiente que suas antecessoras.

O alvo inicial era a produção de software para produtos eletrônicos de consumo (fornos de microondas, agendas eletrônicas, etc.). Um dos requisitos para esse tipo de software é ter código compacto e de arquitetura neutra.

A linguagem obteve sucesso em cumprir os requisitos de sua especificação, mas apesar de sua eficiência não conseguiu sucesso comercial. Com a popularização da rede Internet, os pesquisadores da Sun Microsystems perceberam que aquele seria um nicho ideal para aplicar a recém criada linguagem de programação [22].

A partir disso, adaptaram o código Java para que pudesse ser utilizado em microcomputadores conectados a Internet, mais especificamente no ambiente da *World Wide Web*. Java permitiu a criação de programas chamados *applets*, que trafegam e trocam dados através da Internet e utilizam da interface gráfica de um *web browser*. Com isso, a linguagem conseguiu grande popularização, passando a ser usada amplamente na construção de documentos *web* que permitem maior interatividade.

A linguagem de programação Java foi escolhida neste trabalho porque, além de ser uma linguagem padrão de mercado, é interoperável, funcionando em qualquer sistema operacional, possui um conjunto de APIs que facilitam o trabalho do desenvolvedor poupando esforços e tempo de programação, além de ser um padrão bem aceito no mercado de desenvolvimento de software.

3.3.1 Fases de uma Aplicação Java

Para que um sistema desenvolvido em Java funcione corretamente necessita-se de algumas ferramentas e processos, como um ambiente de execução, a linguagem, a interface de programas aplicativos (API) Java e as bibliotecas que serão utilizadas. Os programas Java frequentemente passam por cinco fases, sendo elas [22]:

- Edição: Processo de escrever o código-fonte em um programa editor. Isso pode ser feito com um programa simples como Notepad++ ou bloco de notas ou em IDEs (Ambientes de Desenvolvimento Integrado) robustas como o Netbeans e o Eclipse.
- Compilação: Acontece com o comando *javac* para compilar o programa. O que acontece é que o compilador Java interpreta o código e passa esse para *bytecodes*, para que o compilador possa entendê-lo.
- Carga: Essa fase começa passando as classe em *bytecodes* para a memória do computador executados pelo interpretador Java.
- Verificação: Essa fase ocorre na transição dos códigos em *bytecodes* para o interpretador. O que acontece é que esses *bytecodes* são verificados pelo verificador de *bytecodes*, garantindo assim que o código não viole as restrições de segurança do Java.
- Execução: A CPU (Unidade Central de Processamento), realiza a interpretação de cada *bytecodes* e então efetua a ação solicitada.

3.3.2 Características

A linguagem de programação Java foi escolhida porque, além de ser a linguagem de programação mais usada no mundo, possui características que facilitam o desenvolvimento de qualquer programa. As principais dessas características são elencadas a seguir:

- Alto grau de Concisão: Possui um facilidade de entendimento. Por ser orientada a objetos, a organização do código é facilitada, organizando os projetos em pacotes, classes e métodos. Isso auxilia na reusabilidade e extensibilidade.
- Reuso: Permitir integração com pacotes e bibliotecas feitas para Java. O que nos permitiu acesso a Internet por meio de protocolos TCP/IP como HTTP, FTP ou acesso por URLs.
- Linguagem Fortemente Tipada: O que torna o programa confiável e mais seguro em termos de código, não deixando nenhuma incoerência de tipos de dados, chamadas de função e rotinas.
- Linguagem portátil: Isso significa que um programa irá funcionar independente da arquitetura ou sistema operacional utilizado, basta ter o Java instalado que ele será capaz de ser executado.

3.3.3 Java EE (*Enterprise Edition*)

Com a evolução dos sistemas, criou-se muitos padrões de projeto para padronizar e facilitar o modo como se programa. Evoluiu-se as aplicações web, tornando os requisitos funcionais, que dizem respeito às funções e componentes do sistema, mais complexos. Porém em um projeto não existe apenas os requisitos funcionais, contendo também os requisitos não-funcionais que estão relacionados com usabilidade, segurança, manutenção e tecnologias envolvidas. Alguns exemplos são: persistência em banco de dados, transação,

acesso remoto, *web services*, gerenciamento de *threads*, gerenciamento de conexões HTTP, cache de objetos entre outros [22].

Assim, para facilitar o desenvolvimento, a *Sun Microsystems*, que em 2010 foi adquirida pela *Oracle Corporation*, criou uma série de especificações que podem ser utilizadas pelos desenvolvedores para tirar proveito e reutilizar toda essa infraestrutura já feita. Logo, o Java EE (*Enterprise Edition*) consiste nessas implementações bem detalhadas, que podem ser tanto gratuitas quanto pagas, auxiliando no desenvolvimento de software que necessite de um desses serviços de infraestrutura [22].

O JEE ou Java EE consiste em uma série de especificações bem detalhadas, de requisitos sobre como deve ser implementado um software. As principais APIs, dentre as disponibilizadas pela especificação Java Enterprise Edition são Java Server Pages (JSP), Java Servlets, Java Server Faces (JSF), Enterprise JavaBeans Components (EJB), Java Persistence API (JPA), Java Naming and Directory Interface (JNDI) e Java Message Service (JMS) [22].

A instalação e execução de uma aplicação JEE depende da existência de um servidor de aplicações JEE, que forneça um *web container* e possa prover um ambiente de execução (JRE - *Java Runtime Environment*) adequado com todas as bibliotecas necessárias. Neste trabalho, foi utilizado o servidor de aplicações *Glassfish Server*, mas existem diversos servidores de aplicação no mercado compatíveis com a especificação Java EE. Abaixo estão listados os mais conhecidos:

- Apache, Apache Geronimo;
- IBM, IBM Websphere Application Server;
- Jboss, Jboss Application Server;
- Oracle, Oracle Weblogic Server;
- Oracle, Glassfish Server.

3.3.4 *JSP (JavaServer Pages)*

O desenvolvimento de conteúdo dinâmico no Java é complexo, tendo o programador que lidar principalmente com os *layouts* das páginas em HTML e CSS, as funcionalidades da aplicação no Java e a comunicação com o banco de dados. Nesse sentido, para facilitar e organizar a implementação do código, criou-se tecnologias que colaboram com esse trabalho, sendo umas delas o *Java Server Pages* (JSP) que permite usar o HTML de forma direta junto com o Java.

O *Java Server Pages* é uma linguagem *script* para geração de conteúdo dinâmico para páginas da Internet. Sua utilização visa o dinamismo, possibilitando que as páginas em HTML que são estáticas, comecem a ser dinâmicas, possibilitando uma interação do usuário com a página. O seu funcionamento ocorre da seguinte maneira: O servidor recebe uma requisição para uma página JSP, interpretando e gerando a codificação HTML, retornando o resultado da solicitação ao cliente.

Quando se codifica para a Web, é necessária a compreensão de sua arquitetura cliente-servidor. Assim, tem-se como cliente o navegador, que será o componente que fará as requisições. Em contrapartida, o servidor processará essas requisições e retornará uma resposta ao cliente.

Nos clientes são utilizadas as tecnologias HTML, que é o conteúdo apresentado nas páginas web; CSS, que tem o papel de formatar esse conteúdo; e por último o *JavaScript* que possibilita interações entre a página web e o usuário. No lado do servidor são alocadas as tecnologias como o JSP, que será responsável por receber a solicitação do protocolo HTTP, processá-la e enviar uma resposta para o cliente.

3.4 Considerações Finais

Neste capítulo foram apresentados conceitos que servem como referencial teórico, relacionados ao processo de desenvolvimento e arquitetura de um software e foram abordadas as fases para construção de um programa de acordo com conceitos da Engenharia de Software. Também foi abordada a origem e evolução do sistema operacional *Android*, aspectos de sua arquitetura e características importantes para o desenvolvimento de aplicativos para esta plataforma. Por fim, foi apresentado o embasamento teórico a respeito do desenvolvimento de aplicações web com a linguagem de programação Java, que foi utilizada neste trabalho para desenvolver a versão web do aplicativo Doce Desafio.

Capítulo 4

SOA e *Web Services*

Este capítulo estabelece um entendimento sobre SOA (*Service Oriented Architecture* - Arquitetura Orientada a Serviços) e *web services*, seu conjunto de termos associados e suas tecnologias.

4.1 SOA (*Service Oriented Architecture*)

É um modelo de arquitetura de software voltado para a construção de aplicações que implementam processos de negócio ou serviços utilizando um conjunto de componentes fracamente acoplados e orquestrados, a fim de prover um nível de serviço bem definido.

Simplificadamente, um requerente de serviço (cliente) envia uma mensagem a um provedor de serviço (servidor), e o provedor envia uma resposta, podendo ser uma informação requisitada ou a confirmação de que alguma ação foi tomada, conforme ilustra a Figura 4.1. A transação pode ser síncrona ou assíncrona e é independente de protocolo de transporte utilizado [27].

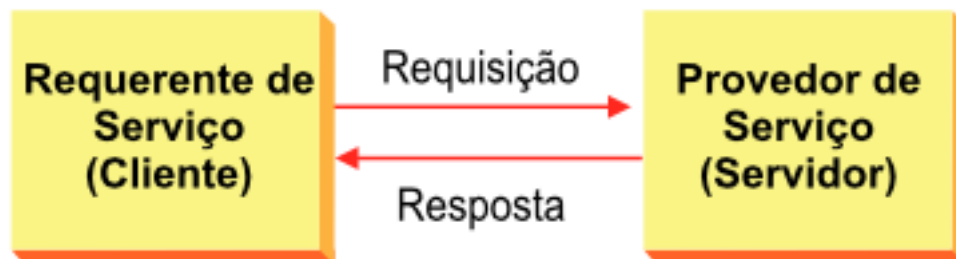


Figura 4.1: Arquitetura Orientada a Serviços [23].

Para possibilitar dinamismo na localização dos serviços, o conceito de SOA prevê a utilização de um mecanismo de registro (*registry*) e descoberta de serviços (*discovery*) que resulta em três papéis bem definidos na arquitetura:

- Provedor de Serviços: Entidade que provê serviços acessíveis pela rede. Ela é responsável pelo gerenciamento, manutenção e publicação do serviço.

- Cliente de Serviços: Entidade que consome um serviço fornecido por um provedor de serviços. Para conhecimento das funcionalidades, localização e requisitos específicos do serviço pode realizar consultas a uma agência de registro de serviços.
- Agência de Registro de Serviços: Entidade que mantém registro dos serviços disponíveis na rede, bem como funcionalidades, localização e protocolos utilizados pelo serviço, entre outras informações.

A interação entre essas três entidades se dá em quatro passos, conforme a Figura 4.2.

1. Publicação: Inicialmente o provedor publica o serviço na agência de registro, fornecendo informações relevantes quanto a utilização do mesmo, como localização do serviço, interfaces de interação, protocolos de transporte e aplicação utilizados, formato dos dados requeridos e parâmetros de qualidade do serviço.
2. Busca: Em seguida, o cliente do serviço solicita à agência de registro a busca por um serviço que satisfaça certos parâmetros.
3. Resposta: Esta, responde ao cliente uma lista com serviços que satisfazem os parâmetros recebidos.
4. Requisição: Baseado em algum critério de avaliação, o cliente escolhe o serviço que mais lhe convém e envia uma requisição ao respectivo provedor do serviço, que por sua vez envia o resultado da requisição.

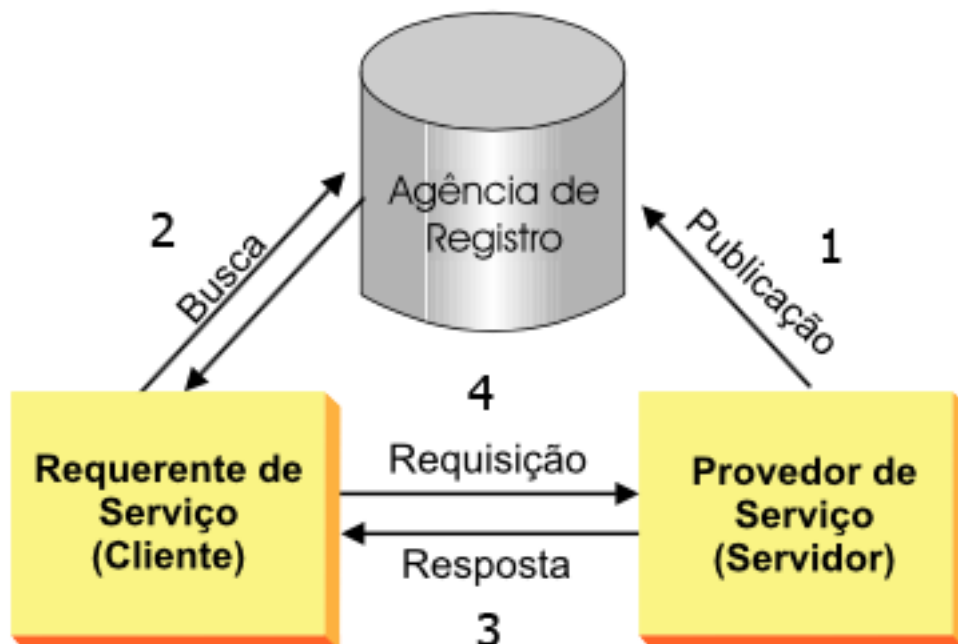


Figura 4.2: Mecanismo de Registro e Descoberta da Arquitetura Orientada a Serviço [23].

4.1.1 Características

Segundo Lazzeri [29], as principais características de SOA são:

- Acoplamento Fraco

Aplicações desenvolvidas com base na arquitetura SOA são inerentemente fracamente acopladas. Isso ocorre porque cada entidade é responsável pelo desenvolvimento de uma unidade lógica de software e sua interface de acesso. A implementação é isolada da interface, pois existe o encapsulamento da lógica de implementação do serviço que fica transparente ao cliente. Assim, através da utilização do mecanismo de descoberta evita-se acoplamento quanto a localização do serviço com o qual deseja-se interagir, ou seja, existe transparência de localização do serviço. Essa é uma das grandes vantagens do desenvolvimento baseado em SOA. O serviço com o qual deseja-se interagir pode ser descoberto em tempo de execução levando em conta parâmetros a priori desconhecidos.

Não existem restrições quanto a linguagem de programação e a plataforma de software a serem utilizados, uma vez que um protocolo de aplicação independente de plataforma é utilizado. Isso resulta em flexibilidade no desenvolvimento do serviço, podendo a escolha da plataforma e da linguagem de programação ser baseada apenas em restrições do próprio serviço, não do sistema como um todo.

- Interoperabilidade

Serviços devem ser implementados com uso de tecnologias padronizadas e disponíveis a um grande número de plataformas de software. Isto implica que os mecanismos de invocação (protocolos, descrições de interfaces, mecanismos de descoberta) devem ser neutros e ser compatíveis com padrões largamente aceitos. O uso de um protocolo de aplicação independente de plataforma resulta em interoperabilidade entre serviços. Protocolos proprietários e fracamente padronizados apresentam problemas de interoperabilidade entre softwares desenvolvidos por diferentes fabricantes. Por vezes são bloqueados por *firewalls* e impossibilitados de serem aplicados em um ambiente multi-organizacional.

- Composição

Serviços podem ser compostos para formar novos serviços com um nível maior de abstração e provendo funcionalidades agregadas. A flexibilidade na composição de novos serviços a partir de serviços já disponíveis na rede é um grande benefício da arquitetura. Por meio de plataformas de composição de serviços um projetista pode facilmente compor novos serviços.

- Reusabilidade

Por serem módulos independentes de código e fracamente acoplados, os serviços podem ser facilmente reutilizados, resultando em um alto nível de produtividade no desenvolvimento de software. O reuso de código gera redução no tempo de desenvolvimento, bem como redução de custos.

- Alta Granularidade

O encapsulamento de funcionalidades no nível de serviço evoca um alto grau de granularidade nos componentes básicos da arquitetura. Para o desenvolvimento de aplicações complexas e extensas, a alta granularidade traz vantagens na medida em que detalhes de implementação são deixados à equipe de desenvolvimento responsável por aquele serviço.

- Ubiquidade

Os serviços devem ser acessíveis a partir de qualquer lugar e a qualquer momento, facilitando a composição de serviços entre empresas.

4.2 Composição de Serviços

Uma das principais características de um serviço é a sua capacidade de se compor com outros serviços para a elaboração de um novo com mais alto nível de abstração. Esta seção descreve os mecanismos de composição de serviços presentes e possíveis de acordo com a arquitetura orientada a serviço.

- Orquestração

Orquestração é um mecanismo de composição ou agregação de serviços visando à obtenção de um novo serviço com funcionalidades mais significativas. É um componente importante da arquitetura orientada a serviços, pois encapsula logicamente serviços de forma transparente ao cliente. A orquestração permite que o desenvolvimento de novos serviços seja realizado de forma dinâmica, flexível e adaptável às necessidades de negócios que se alteram frequentemente. O tempo de desenvolvimento de novos serviços é fortemente reduzido em função do reuso de código. A orquestração de serviços define um fluxo de execução (*workflow*) de um processo, ou seja, um *script* de orquestração que é processado por um programa denominado máquina de orquestração, um controlador central. Nesse *script* são definidas as chamadas a serviços componentes necessários à obtenção do novo serviço [21].

- Coreografia

Coreografia é um mecanismo de especificação de um protocolo de troca de mensagens entre duas entidades participando em uma interação. Seu objetivo é especificar um contrato entre essas duas entidades independentes de software localizados em corporações distintas. Ao contrário da orquestração, na coreografia, os serviços conhecem uns aos outros e cada um conhece sua função dentro do fluxo do processo. A coreografia é realizada através da troca de mensagem entre os serviços, não havendo um controlador central [21].

4.3 Web Services

A tecnologia atual que apresenta maior conformidade com a arquitetura e o paradigma de computação orientada a serviços é a de *web services* ou Serviços Web. A difusão de tal tecnologia alavancou a arquitetura orientada a serviços no mercado de desenvolvimento de software. As vantagens de utilização de SOA em um ambiente que possui restrições quanto a interoperabilidade de código são enormes, pois SOA é um modelo arquitetônico agnóstico a qualquer plataforma de tecnologia. *web services* é a plataforma mais associada atualmente à SOA [23]. Um serviço web é um sistema de software projetado para suportar interações máquina-máquina interoperáveis sobre uma rede [11].

Serviço é um recurso abstrato que representa a capacidade de executar tarefas que possuam uma funcionalidade coerente na visão do provedor e do consumidor. De forma geral, um serviço computacional pode ser oferecido através de uma interface com o usuário, como um aplicativo ou site web, ou pode ser implementado e disponibilizado por um componente de software para ser usado por outros componentes, como um serviço de impressão utilizado por diversos aplicativos. Assim como a interface com o usuário deve ser de fácil entendimento, um serviço prestado a outro programa deve ter uma interface e um protocolo de comunicação bem definidos para que os clientes possam utilizá-lo. Uma interface é uma especificação que existe entre componentes de software e que define as formas de interação entre eles — muitas vezes chamada de contrato. Uma interface pode definir, por exemplo, métodos, constantes e tipos de dados. O protocolo de comunicação é uma especificação do conjunto de mensagens trocadas entre os aplicativos.

4.3.1 Especificações

Segundo Newcomer [31] e Walsh [37] os web services são formados pela seguinte especificação:

- *Extensible Markup Language (XML)*

XML é uma linguagem que utiliza marcações e atributos para representação de dados. XML não impõe restrições quanto ao conjunto possível de marcações e atributos a serem utilizados em um dado documento. Consequentemente, é muito mais rica e descritiva que outras linguagens de marcação, que definem um conjunto limitado de elementos constituintes da linguagem. Dado que a interação entre cliente e serviço se dá de modo conversacional, a linguagem XML é utilizada na representação de mensagens de requisição e resposta de web services. A característica mais destacável da linguagem XML é que as mensagens têm codificação em texto puro, ou seja, não há necessidade de protocolos proprietários para codificação ou decodificação de mensagens e dados. Além disso, podem ser lidas por um ser humano, facilitando o processo de depuração de software [31].

- *Web Services Description Language (WSDL)*

O WSDL descreve os serviços disponibilizados na rede por meio de uma semântica XML, ele providencia a documentação necessária para se chamar um serviço e o procedimento necessário para que esta comunicação se estabeleça de forma padronizada e independente de plataforma. Um documento WSDL representa um contrato entre o provedor de serviços e seus clientes e define os serviços como uma coleção de portas na rede [31].

O WSDL, atualmente na versão 1.2, é composto por sete elementos XML, que representam as partes abstrata e concreta.

Na parte abstrata temos quatro elementos:

- types - fornece a definição dos tipos de dados para descrever as mensagens trocadas entre aplicações, normalmente representadas por um documento XSD (XML Schema Definition);
- message - representa a informação que será trocada através das definições dos tipos de dados;
- porttype - é um conjunto de operações suportadas por um ou mais end points, onde cada operação se refere a uma mensagem de entrada, saída ou erro;
- operation - descreve a ação suportada pelo serviço.

Na parte concreta, temos os outros três elementos:

- binding - define uma especificação de protocolo e formato de dados para as mensagens definidas em um port type;
- port - é um end point, representa a combinação de um binding e um endereço de rede;
- service - é uma coleção de portas. Cada elemento port se relaciona com um elemento binding particular, indicando qual interface e qual protocolo de comunicação estão sendo utilizados nessa implementação.

- *Simple Object Access Protocol (SOAP)*

SOAP é um protocolo de comunicação que define a formatação e a codificação de mensagens XML a serem enviadas pela rede. O protocolo define um mecanismo simples e leve para troca de informações entre partes em um ambiente distribuído e descentralizado. Ele foi projetado para comunicação na Internet podendo utilizar os protocolos HTTP ou SMTP [12] para transporte de suas mensagens. O objetivo deste protocolo é prover um meio de comunicação entre aplicações com diferentes tecnologias e linguagens de programação. Ao utilizar protocolos padrões da Internet, ele não é bloqueado por *firewalls* nem por roteadores como outros protocolos de aplicação [31]

A especificação SOAP 1.2 contém três partes principais:

- Modelo de empacotamento - envelope SOAP - define o conteúdo da mensagem SOAP.

- Mecanismo de serialização - conjunto de regras de codificação - define os tipos de dados utilizados na aplicação;
- Mecanismo de comunicação – convenção - define as chamadas e respostas através de procedimentos remotos.

Toda mensagem SOAP obrigatoriamente contém os elementos <Envelope> e <Body> e opcionalmente os elementos <Header> e <Fault>.

- O elemento <Envelope> é a raiz do documento XML e representa a mensagem propriamente dita;
- O elemento <Body> contém a carga de informações (operações e parâmetros) que são entregues ao destinatário da mensagem. Este elemento, de acordo com a especificação SOAP, pode conter um elemento <Fault> que pode ser utilizado no processamento de falhas do serviço Web.
- O elemento <Fault> descreve erros de chamada de métodos remotos ou mantém informações acerca do tipo de erro. Portanto, ele conta com os elementos <FaultCode>, <FaultActor>, <FaultString> e <Detail>.
- O elemento <Header> expande uma mensagem SOAP. Ele define algumas características opcionais e acordos negociáveis entre as partes, seu conteúdo deve ser aceito pelas aplicações que estiverem se comunicando.

- *Universal Description, Discovery and Integration (UDDI)*

UDDI especifica protocolos necessários à descoberta de novos serviços disponíveis na rede. Funciona como uma agência de registro de web services. Dado um conjunto de parâmetros de especificação de um serviço (nome, qualidade, localização etc) a agência é responsável por encontrar serviços cadastrados que satisfaçam os parâmetros fornecidos. A agência mantém um banco de dados centralizado dos web services disponíveis na rede classificados por tipo de serviço, informações do provedor, qualidade dos serviços e assim por diante [31].

O protocolo UDDI, atualmente na versão 3.0, apresenta três papéis, representados sob a forma de XML Schemas.

- Páginas Brancas - contém identificadores sobre o contato técnico do serviço oferecido;
- Páginas Amarelas - contém informações genéricas sobre os tipos e localização dos serviços disponíveis;
- Páginas Verdes - contém informações técnicas sobre um determinado serviço Web.

A especificação UDDI define quatro estruturas de dados, também descritas como documentos XML, onde cada elemento descreve o tratamento dado ao serviço Web

- `businessEntity` - estrutura de alto nível (páginas brancas) que contém, para cada serviço, as informações (nome, categoria, identificadores, entre outros) sobre a organização que publicou o serviço Web.
- `businessService` - contém informações descritivas sobre web services (páginas amarelas), tais como nome e descrição do serviço publicado.
- `bindingTemplate` - contém informações técnicas sobre o serviço Web tais como forma de acesso e endereços dos pontos acesso ao serviço (páginas verdes).
- `tModel` - mecanismo usado para a troca de definições abstratas (metadados) sobre um serviço Web, contém as descrições do serviço Web, opcionalmente aponta para documentos WSDL.

- *Representational State Transfer (REST)*

Estilo arquitetônico para a criação de web services diferente do modelo adotado pelo SOAP. O REST foi desenvolvido juntamente com o protocolo HTTP 1.1 com ideia oposta à do SOAP, que tem como objetivo estabelecer um protocolo para comunicação de objetos e serviços. REST propõe utilizar os métodos HTTP (GET, POST, PUT, HEAD, OPTIONS e DELETE) para criar serviços que poderiam ser acessados por qualquer tipo de sistema. Portanto, sua ideia não é de um protocolo, mas sim de um padrão arquitetural para serviços expostos através do protocolo HTTP. Esse estilo de criação de serviços também pode ser chamado de *web services*, pois permite a comunicação entre programas de computador através da troca de arquivos XML pela web. Tem ainda a vantagem de permitir ao usuário a realização de consultas aos recursos com o uso de um navegador web comum. [24]

4.4 Considerações Finais

Neste capítulo foram apresentados os conceitos de Arquitetura Orientada a Serviços (SOA) e suas principais características, assim como a tecnologia *web services*, que implementa os conceitos de SOA e foi utilizada neste trabalho, especificamente com um RESTful *Web Service*, para viabilizar a sincronização entre os dados do Aplicativo Doce Desafio e da aplicação web que foi desenvolvida.

Capítulo 5

Aplicativo Doce Desafio

Este capítulo apresenta a primeira versão do aplicativo Doce Desafio, sua contextualização e as especificações técnicas.

5.1 Contextualização

O programa Doce Desafio (Diabetes, Educação em Saúde e Atividades Físicas Orientadas da FEF/UnB (Faculdade de Educação Física da Universidade de Brasília) , coordenado pela professora doutora Jane Dullius, orienta a diabéticos e familiares no alcance da qualidade de vida. O Doce Desafio oferece cursos sobre a doença, higiene pessoal, alimentação, insulinização, controle glicêmico e atividades físicas.

A diabetes é uma doença que requer atenção especial com a rotina do paciente, sendo necessário controlar a alimentação, medicação e atividades físicas, assim como a responsabilidade e a assiduidade na monitoração.

Muitas vezes, seja por falta de tempo, seja por esquecimento dos horários de aplicação da insulina ou da dieta a ser seguida, os pacientes acabam por não cumprir suas responsabilidades, negligenciando suas metas, não tomando o cuidado necessário para manter um estilo de vida saudável. No entanto, esses dados, que são coletados da rotina de cada paciente, são fundamentais para que haja um tratamento eficaz e bem sucedido.

Com isso em mente, foi sugerido utilizar das tecnologias e conhecimentos da Computação para criar um sistema que funcionaria como facilitador para esse grupo de pessoas. Assim nasceu a ideia de se criar um aplicativo para *smartphones* que permitisse aos diabéticos tipo 1, de uma forma menos desgastante, tomar notas de suas rotinas, buscando auxiliar também a equipe técnica que acompanha o paciente.

Nesse cenário, optou-se pela plataforma móvel pela sua característica portátil, em que o usuário poderia utilizar seu *smartphone* a qualquer momento e qualquer lugar, possuindo acesso sempre que necessário. O software desenvolvido para um PC (Computer) perderia esse fator otimizador da tarefa, afinal eles necessitariam escrever suas rotinas em algum bloco de anotações ou caderno e depois transcrevê-las para o computador quando voltassem para casa, acrescentando dificuldade e aborrecimento quanto a essa tarefa.

5.2 Aplicativo *Android* - Primeira Versão

Decorrente das ações do programa Doce Desafio, surgiu o aplicativo Doce Desafio, desenvolvido no ano de 2013 como resultado de um trabalho de conclusão de curso dos alunos Edison e Jair [16] do departamento de Ciência da Computação da Universidade de Brasília.

A utilização da plataforma *Android* foi fundamental para entender e se familiarizar com as necessidades dos pacientes, tanto daqueles que possuem uma rotina mais ativa, quanto daqueles que possuíam menos atividades ao dia.

Foi realizada uma pesquisa de mercado com o intuito de localizar e analisar aplicativos de mesmo propósito, que já estavam disponibilizados para o público, buscando estabelecer como o problema e suas soluções já estavam sendo abordados.

A pesquisa foi realizada na loja virtual de aplicativos do Google (*Play Store*) [8], onde são disponibilizado todos os aplicativos, pagos ou gratuitos, para os *smartphones* que trabalham com o sistema operacional *Android*. Dentre todos os aplicativos que se encaixaram nos filtros de busca, foram escolhidos nove, dentre os mais baixados, que trabalhavam e proviam diferentes serviços, entretanto visavam resolver a mesma problemática, e então foi feito um trabalho de comparação entre eles.

Como pode ser visto na Figura 5.1, examinou-se os principais requisitos funcionais, categorizados por meio do *feedback* dos possíveis usuários.










NOME	VALOR	IDIOMA	LEMBRETES	MÓDULO GLICEMIA	MÓDULO REFEIÇÃO	TABELA DE ALIMENTOS	MÓDULO INSULINA	MÓDULO ATIV. FÍSICA	GRÁFICOS	SUGESTÃO DE INSULINA	FILTROS PARA OS RELATÓRIOS	GERAÇÃO DE PDF
 ONTRACK DIABETES	GRATUITO	INGLÊS	SIM	SIM	SIM	NÃO	SIM	SIM	TODOS OS MÓDULOS	NÃO	APENAS GLICEMIA	NÃO
 GLUCOSE BUDDY: DIABETES LOG	GRATUITO	INGLÊS	SIM	SIM	SIM	NÃO	SIM	SIM	APENAS GLICEMIA	NÃO	NÃO	NÃO
 GLUCOOL DIABETES PREMIUM	R\$ 16,72	INGLÊS	SIM	SIM	SIM	NÃO	SIM	SIM	APENAS GLICEMIA	SIM	NÃO	NÃO
 TRACK3 DIABETES PLANNER	R\$ 14,40	INGLÊS	SIM	SIM	SIM	SIM	SIM	SIM	APENAS GLICEMIA	NÃO	NÃO	NÃO
 SIDIARY DIABETES MANAGEMENT	GRATUITO	VÁRIOS INCLUSIVE PT	NÃO	SIM	SIM	NÃO	SIM	SIM	APENAS GLICEMIA	SIM	NÃO	NÃO
 DIABETES CONTROL	GRATUITO	PORTUGUÊS	NÃO	SIM	NÃO	NÃO	SIM	NÃO	APENAS GLICEMIA	NÃO	SIM	NÃO
 DBEES.COM DIABETES MANAGEMENT	GRATUITO	VÁRIOS INCLUSIVE PT	SIM	SIM	SIM	NÃO	SIM	SIM	APENAS GLICEMIA	NÃO	NÃO	NÃO
 SOCIAL DIABETES	GRATUITO	VÁRIOS INCLUSIVE PT	SIM	SIM	SIM	NÃO FUNCIONA	SIM	SIM	APENAS GLICEMIA	SIM	NÃO	NÃO
 DIABETES	GRATUITO	VÁRIOS INCLUSIVE PT	SIM	SIM	NÃO	NÃO	NÃO	NÃO	APENAS GLICEMIA	NÃO	NÃO	NÃO

Figura 5.1: Tabela de Comparação entre os Aplicativos [16].

Após a pesquisa, notou-se que algumas funcionalidades consideradas essenciais não tinham ocorrência em todos os nove aplicativos, foram elas:

- A existência de uma tabela de alimentos;
- Uma ferramenta que funcionasse como um filtro de relatórios; e
- Uma ferramenta de geração de gráficos, que usaria os dados registrados pelos usuários.

Comparando todos os nove aplicativos, nenhum apresentou todas as funcionalidades incorporadas, como mostra a Figura 5.1. Nenhum desses aplicativos apresentou também uma ferramenta ou uma opção de geração de PDF dos dados, dificultando assim o serviço de análise por parte do médico ou do responsável clínico do paciente [16].

5.2.1 Requisitos

O levantamento de requisitos foi feito levando em consideração as necessidades obtidas por meio de reuniões com profissionais dos Departamentos de Nutrição (NUT), Medicina (FS) e Educação Física (FEF) da Universidade de Brasília (UnB). As reuniões eram feitas no Centro Olímpico da Universidade, na sede do programa Doce Desafio [16].

Requisitos Funcionais

Grande parte dos requisitos foram coletados por profissionais da FEF (Faculdade de Educação Física). Os primeiros requisitos a serem coletados e implementados foram os solicitados pelo Departamento de Nutrição da Universidade de Brasília. Entre esses requisitos estava uma tabela de alimentos, que inicialmente o usuário informaria qual a refeição do dia ele realizou (café da manhã, almoço, lanche da tarde, jantar e lanche da noite). Posteriormente, seria informado quais foram os alimentos que ele ingeriu naquela refeição, procurando assim nessa tabela, e o aplicativo no final informaria qual foi a quantidade total de carboidrato que ele absorveu naquela refeição.

Logo após, participando de reuniões com profissionais de Medicina, entrou-se em consenso quanto a importância e relevância de conter no aplicativo um filtro de necessidades e limites, que levassem em conta o perfil do paciente e as medidas da sua glicemia. Os principais argumentos destacados por esses profissionais, era que a partir desses dados (perfil do paciente e medidas de sua glicemia), colheriam informações significativas para o tratamento da doença, pois por meio do perfil do paciente é possível, por exemplo, chegar a algumas variáveis imprescindíveis para a contagem de carboidratos.

Algumas das necessidades observadas juntos aos portadores de diabetes que frequentavam o programa foram:

- Armazenar os dados das atividades físicas realizadas;
- Armazenar os dados da dosagem de insulina administrada durante o dia;
- Salvar alarmes para lembrar do horário da medicação;
- Com base na glicemia informada e na quantidade de carboidratos que pretende se ingerida, calcular quantas unidades de insulina deveriam ser administradas para atingir uma determinada glicemia.

Alguns pacientes também sugeriram que o aplicativo, além de auxiliar os pacientes em suas necessidades e tarefas, pudesse facilitar o acompanhamento com os profissionais de saúde. Assim, chegou-se a conclusão da importância de aplicar alguns filtros aos dados para gerar diversos tipos de relatórios, como por exemplo:

- Filtrar todos os dados por um intervalo de tempo de hora e data;
- Filtrar os dados das glicemias por um intervalo de medidas;
- Filtrar os dados das refeições por um intervalo de carboidratos ;
- Filtrar os dados das insulinas pelo tipo da medicação;
- Visualizar os dados desses relatórios em forma de gráfico;
- Exportar os dados desses relatórios para o formato PDF.

5.2.2 Requisitos não Funcionais

Para o desenvolvimento do aplicativo, levou-se em conta também a necessidade e importância da usabilidade e facilidade de interação. Afinal, o público alvo do aplicativo podia variar de pacientes que estivessem na adolescência (na faixa dos 14 anos) até aqueles que atingiam uma idade mais avançada (na faixa dos 50 anos) [16].

Além disso, pesquisas mostraram que parte considerável dos usuários do sistema operacional *Android* ainda fazem uso da versão 2.3. Dessa forma, a implementação realizada foi desenvolvida para dar suporte, desde a versão em questão até a última versão, lançada até o presente momento.

Finalmente, para o armazenamento dos dados foi necessário utilizar um banco de dados que apresentasse bom desempenho em dispositivos móveis a atendesse as necessidades da aplicação.

5.2.3 Banco de Dados

Continuando o desenvolvimento do aplicativo Doce Desafio, foi utilizado o próprio banco de dados interno do Android, o banco *Sqlite* [2], que suporta de forma eficiente e flexível o armazenamento dos dados e permite ser posteriormente copiado e restaurado.

A modelagem do banco de dados, como pode ser vista na Figura 5.2, possui nove tabelas, das quais apenas três estão relacionadas entre si.

As tabelas perfil, glicemia, insulina, exercícios, metas e lembretes não possuem relacionamentos, pois na primeira versão do aplicativo, cada informação armazenada era vinculada a apenas uma pessoa, o usuário em questão. Portanto, todos os dados seriam uma relação um para um. As tabelas que se relacionam, refeição e alimentos, possuem um relacionamento muitos para muitos, assim sendo necessário a criação de uma tabela auxiliar, pois toda vez que se insere uma refeição são adicionados diversos alimentos e um único alimento pode estar relacionado a várias refeições.

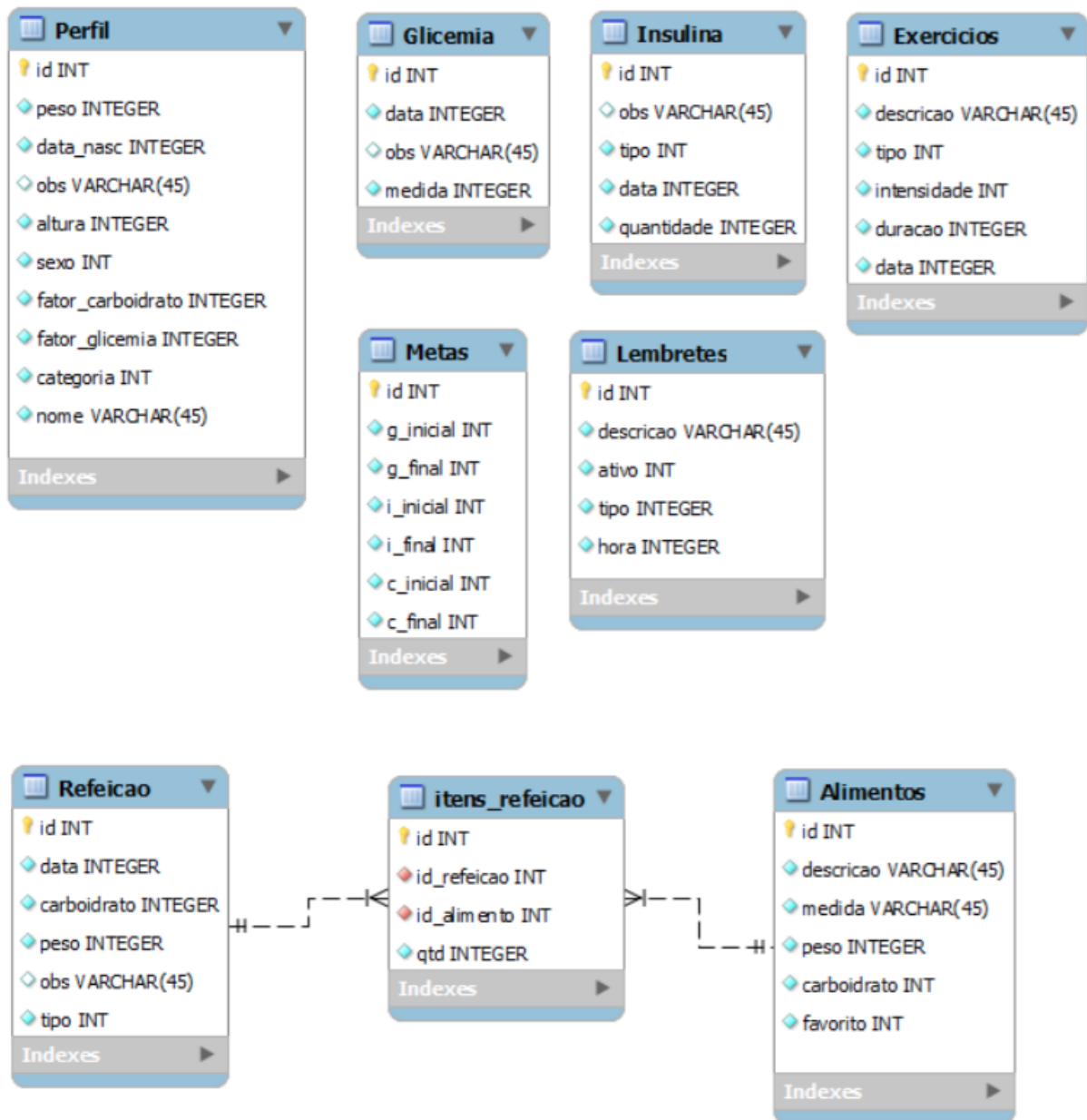


Figura 5.2: Modelagem do Banco de Dados Utilizado no Aplicativo [16].

5.3 Considerações Finais

Neste capítulo foi apresentado, a primeira versão do aplicativo Doce Desafio, criada em 2013, a motivação para sua criação e sua estrutura. O próximo capítulo destina-se a abordagem da proposta deste trabalho, que foi a melhoria da primeira versão do aplicativo e a criação de uma aplicação web.

Capítulo 6

Sistema Doce Desafio - Aplicativo *Android* e Aplicação Web

Este capítulo apresenta a segunda versão do aplicativo, após a implementação das mudanças e melhorias propostas e também de sua expansão para um versão web.

6.1 Aplicativo *Android* - Segunda Versão

Com dois anos de existência do aplicativo Doce Desafio, notou-se a necessidade de expandir e melhorar algumas funcionalidades. Com a coleta e análise dos *feedbacks* dos usuários do aplicativo e com a experiência da professora doutora Jane Dullius, da FEF UnB (Faculdade de Educação Física), responsável pelo projeto Doce Desafio, notou-se a necessidade de atualização do software provendo novas funcionalidades e também seguindo a proposta de trabalhos futuros do trabalho de conclusão de curso, que deu origem ao aplicativo. Nesse contexto, constatou-se que o desenvolvimento de uma aplicação web integrada ao aplicativo auxiliaria no acompanhamento pelos profissionais de saúde do tratamento do diabetes.

6.1.1 Requisitos

Com as avaliações e sugestões dos usuários, foram elencados alguns pontos para manutenção e atualização:

- Notou-se que a página de perfil não abrangia todas as possíveis categorias, como, por exemplo, os obesos ou sedentários, tornando as possibilidades de perfil muito restritas, portanto decidiu-se retirar essa tela do aplicativo;
- A possibilidade de mesclar os gráficos de consumo de carboidrato, a quantidade de insulina aplicada e as atividades praticadas, para melhor visualização dos níveis de açúcar no sangue, demonstrando como cada atividade influencia no controle;
- Organizar a tabela de alimentos por ordem alfabética e alterar o formato de exibição dos nomes;
- Colocar uma barra lateral na aba de alimentos que trafegue pelas letras do alfabeto;

- Definir o tipo de refeição de acordo com o horário que foi inserida;
- Criar mais opções de refeição: Café da Manhã, Colação, Almoço, Lanche, Janta e Ceia;
- No menu atividades físicas, criar um espaço de favoritos, exibindo as atividades físicas preferidas do usuário;
- No menu atividades físicas, criar mais opções de intensidade na prática de atividades físicas: Muito leve, Leve, Moderado, Intenso e Muito Intenso;
- No menu atividades físicas, criar opção para selecionar a modalidade da atividade praticada, por exemplo: Corrida, Natação, Musculação etc;
- Criar opção para definir alarmes para despertar nos horários programados pelo usuário, com o intuito de lembrá-lo de aplicar a insulina;
- No menu glicemia, criar um campo para os tipos de glicemia: Em jejum, Pré-prandial e Pós-prandial;
- No menu relatórios, criar uma opção de filtragem de acordo com o tipo de glicemia;
- Geração *backup* em formato *CSV*.

6.1.2 Melhorias e Otimizações

Foram priorizadas algumas das melhorias sugeridas em detrimento de outras, devido à quantidade de mudanças e ao escopo do trabalho ser muito amplo. Assim sendo, são apresentadas a seguir as telas com as mudanças implementadas no aplicativo.

A Figura 6.1 apresenta a nova tela inicial do aplicativo Doce Desafio. Nesta nova versão, o aplicativo pode ser utilizado por múltiplos usuários que serão identificados por uma conta de usuário com *login* e senha.

Logo do aplicativo: Um personagem amarelo com braços levantados, sobreposto a um hexágono amarelo.

Usuário

Senha

Entrar

Criar nova conta Pular

manter conectado

Figura 6.1: Tela de *Login*.

A Figura 6.2 apresenta a mudança implementada na tela de metas, na seção de últimos dados armazenados no aplicativo. Observa-se que na figura da esquerda, os dados de glicemia e refeição não especifica seus tipos, como se observa na insulina. Posteriormente a mudança, como pode ser visto na figura da direita, passam a especificar seus tipos, como em glicemia sendo o tipo em jejum e refeição sendo o tipo café da manhã.

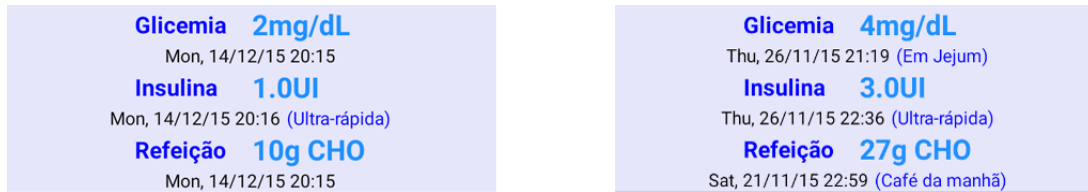


Figura 6.2: Tela de Últimos Dados.

A Figura 6.3 apresenta as mudanças implementadas no menu de restauração/*backup* do aplicativo, localizado na parte inferior da tela. Observa-se que, antes das alterações, não era possível realizar *backup* em formato CSV (*Comma Separated Values*), o que se tornou possível atualmente. Ao clicar na opção 'Gerar CSV' é criado um arquivo com formato CSV para cada opção do menu (Refeição, Glicemia, Insulina e Atividades Físicas) contendo todos seus respectivos dados, e esse arquivo é armazenado na pasta Doce Desafio do *smartphone*, podendo, posteriormente, ser importado para um software como Excel ou semelhante. Também é disponibilizada opção para desconectar/fazer *logout* da conta acessada e opção para verificar se há dados em espera por sincronização com a aplicação web.



Figura 6.3: Tela de Backup.

A Figura 6.4 apresenta as alterações implementadas na tela de atividades físicas. Como proposta para esta mudança, solicitou-se que as opções de intensidade dos exercícios fosse ampliada, pois somente as três opções existentes era muito restritivo. Além disso, solicitou-se também a incorporação de opções de modalidades para caracterizar melhor a atividade física.

Na tela a direita da Figura 6.4 pode-se observar a criação da lista contendo as modalidades, e logo abaixo uma lista contendo as intensidades, que agora podem ser: Muito leve, Leve, Moderado, Intenso e Muito intenso.



Figura 6.4: Tela de Exercícios.

A Figura 6.5 apresenta a mudança realizada na tela de glicemia, em que foi requisitada a possibilidade de caracterizar o tipo de glicemia, nas alternativas: Em jejum, Pré-prandial ou Pós-prandial. Na tela da direita, observa-se a lista que possibilita essa caracterização.

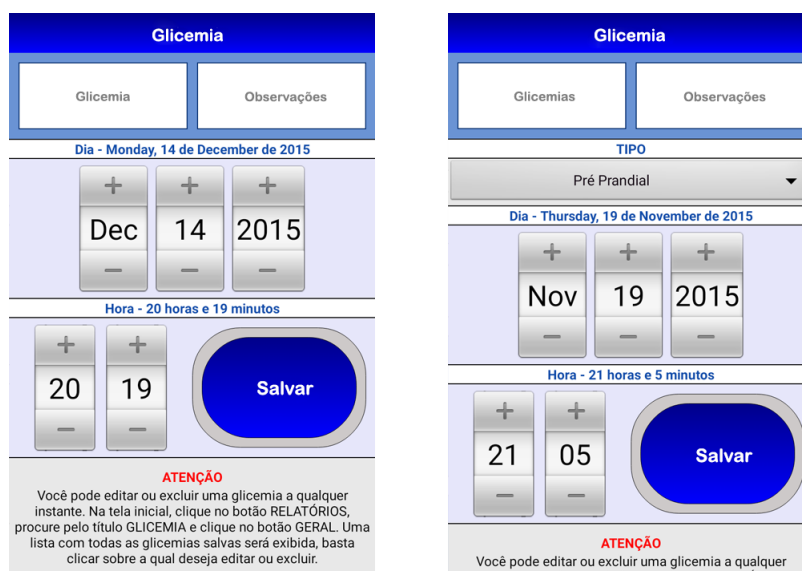


Figura 6.5: Tela de Glicemia.

Na Figura 6.6 é apresentada a mudança na tela de refeição, em que a sugestão de insulina, dependendo da quantidade de carboidratos da refeição, estava localizada de forma pouco intuitiva. Por isso, visando uma melhoria na usabilidade, a opção foi deslocada para a tela seguinte, de forma que possa ser usada com mais facilidade.

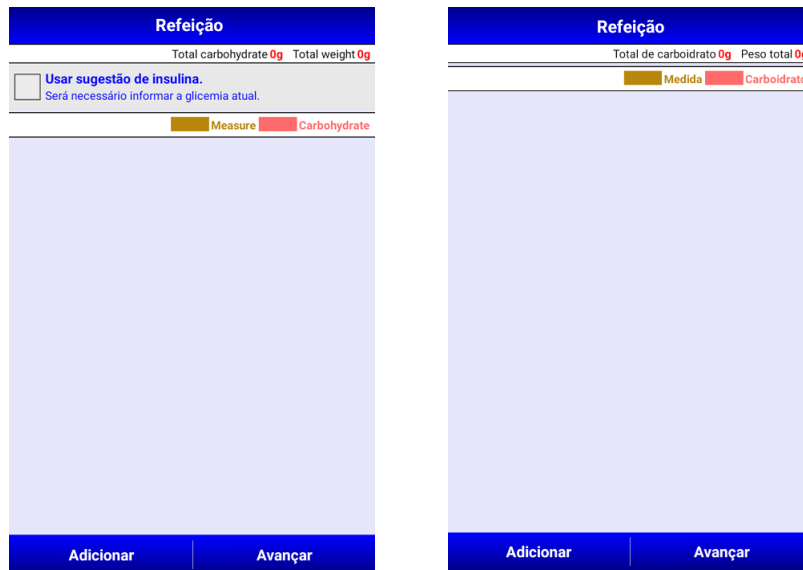


Figura 6.6: Tela de Refeição.

Na Figura 6.7 é apresentada a ampliação das opções de tipo de refeição. Essa mudança foi solicitada porque é indicado que pessoas com diabetes tenham um número de refeições bem distribuído ao longo do dia. As opções disponíveis agora são: Café da manhã, Colação, Almoço, Lanche, Jantar e Ceia, como visto na Figura 6.8.

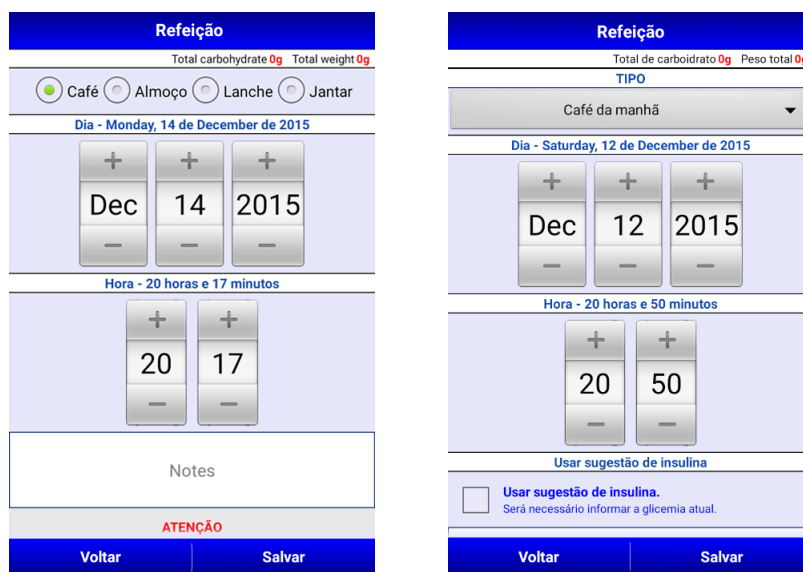


Figura 6.7: Tela de inclusão de Refeição.



Figura 6.8: Tela com Lista de Opções.

A Figura 6.9 apresenta as mudanças realizadas na tela de relatórios, em que foi adicionada a opção de relatórios por tipo, na seção de glicemia, e também adicionada a opção de relatórios por tipo, na seção de atividades físicas. Essas opções possibilitam o aplicativo filtrar os relatórios a partir dos tipos de glicemia, conforme mostrado a Figura 6.10, ou dos tipos de atividades físicas.



Figura 6.9: Tela de Relatórios.

Glicemia - Tipo	
Em Jejum	
Medida	4 mg/dL
Obs: teste 2	
Tipo:Em Jejum	Thu, 26/11/15 21:19
Medida	4 mg/dL
Obs: d	
Tipo:Em Jejum	Sun, 22/11/15 00:37
Medida	5 mg/dL
Obs: f	
Tipo:Em Jejum	Sun, 22/11/15 00:34
Medida	5 mg/dL
Obs: h	
Tipo:Em Jejum	Sun, 22/11/15 00:32
Medida	5 mg/dL
Obs: gh	
Tipo:Em Jejum	Sun, 22/11/15 00:25

Figura 6.10: Tela de Relatórios por Tipo de Glicemia.

6.1.3 Banco de Dados

O banco de dados do aplicativo teve de sofrer algumas mudanças e ser readaptado para permitir a integração com a aplicação web.

Como mostra a Figura 6.11, o banco de dados possui onze tabelas e dois relacionamentos.

Foram criadas duas novas tabelas, sendo elas 'login' e 'SyncREST'. A tabela 'login' foi criada para proporcionar a utilização do aplicativo por diferentes usuários e de modo que os dados do usuário ficassem vinculados a sua conta e pudessem ser acessados também na versão web. A tabela 'SyncREST' foi criada para permitir a sincronização dos dados quando o aplicativo estiver sendo utilizado no modo *offline*, assim os dados inseridos recebem uma sinalização de espera por sincronização e assim que uma conexão com a Internet for estabelecida, eles são sincronizados com o banco de dados remoto da aplicação web.

Não foi alterado o número de relacionamentos em relação a versão anterior do banco de dados do aplicativo, pois a lógica de funcionamento permaneceu a mesma. Quando um usuário realiza o *login* no aplicativo, todos os seus dados são importados do banco de dados central da aplicação web e armazenados temporariamente no aplicativo, quando se desconecta da conta, todos seus dados armazenados nas tabelas são excluídos até que se realize outro *login*. Dessa forma, a lógica de relacionamento permanece de um para um, em que todas as informações serão vinculadas a apenas um usuário.

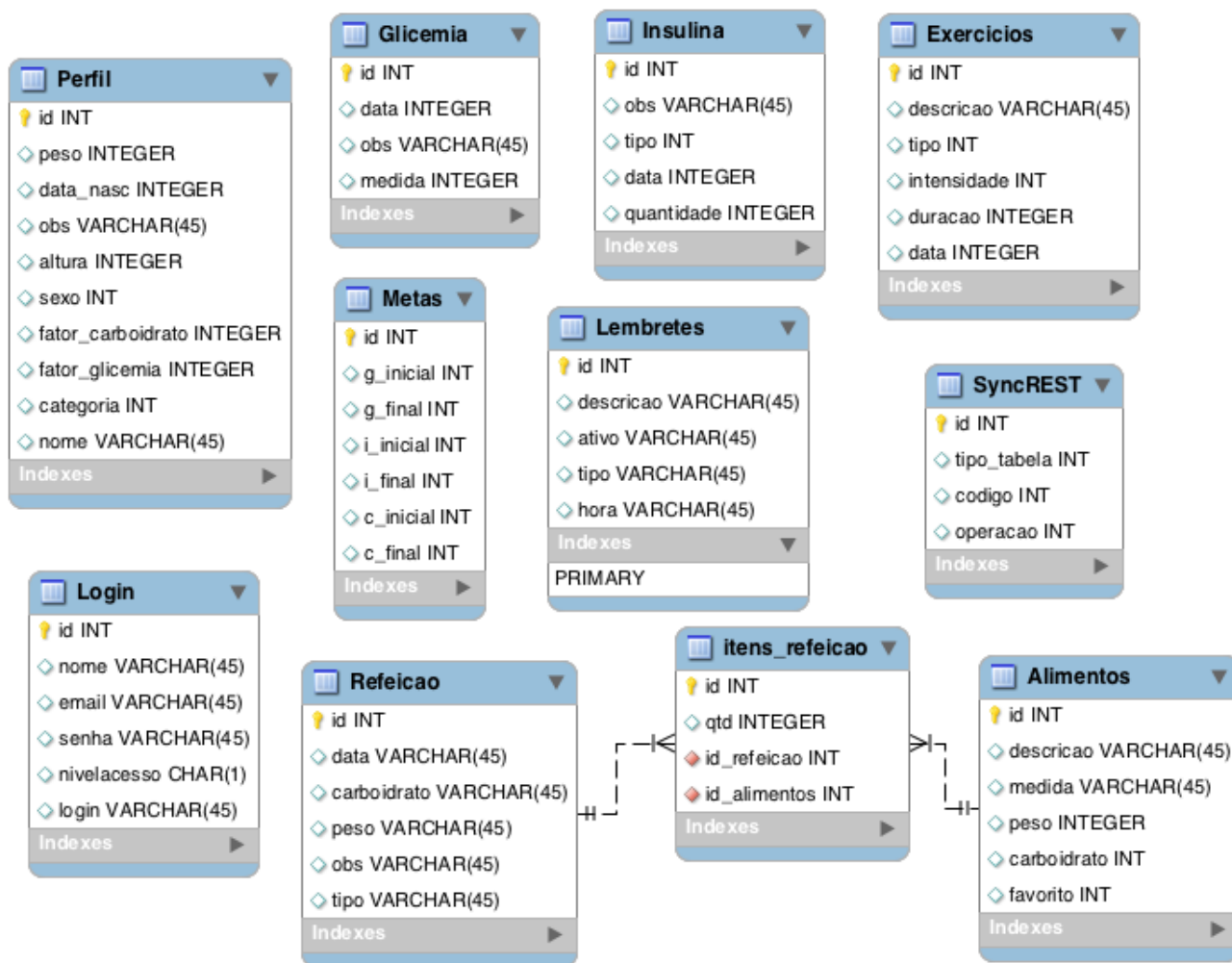


Figura 6.11: Mudanças no Banco de Dados do Aplicativo.

6.2 Aplicação Web

A motivação para desenvolver uma aplicação web partiu do levantamento de melhorias feito pela equipe do Projeto Doce Desafio, assim como, das sugestões de trabalhos futuros propostas por Edison Carvalho e Jair Fernandes [16], em que se constatou a necessidade de um *website* integrado e sincronizado com o aplicativo, de modo que as informações encontradas no aplicativo pudessem ser, posteriormente, acessadas tanto pelos profissionais de saúde quanto pelos responsáveis do projeto Doce Desafio.

Percebeu-se que os dados dos usuários se mantinham apenas localmente no *smartphone* do usuário, não havendo a possibilidade de apresentar esses dados de maneira acessível para a equipe de saúde que acompanhava os diabéticos. Assim, surgiu a ideia de desenvolver uma aplicação web, facilitando o contato dos médicos com os pacientes e usuários do aplicativo.

Para tal, pesquisamos e estudamos sobre possíveis tecnologias e linguagens que auxiliassem nessa tarefa. Como fruto desse esforço, optou-se por desenvolver o sistema web

com a tecnologia Java EE e JSP (*Java Server Pages*), persistindo os dados no banco de dados *PostgreSQL* e permitindo a sincronização entre aplicativo e *website* por meio de *web services*.

Na aplicação web, foram criados níveis de acesso, sendo uma opção para os administradores e outra para os usuários. Como usuário, as permissões são apenas para visualização dos dados até então já cadastrados, ou seja, as tabelas de alimentos, atividade física, glicemia e insulina, com um diferencial de filtros. Com o nível de acesso de administrador, é possível realizar o gerenciamento do sistema, podendo esse, além de visualizar os dados de todos os usuários, realizar o cadastro de novos usuários, listagem dos usuários cadastrados, a edição e a exclusão dos cadastros.

6.2.1 Estrutura da Aplicação Web

A aplicação web possui duas visões diferentes de acordo com o tipo de usuário, este sendo administrador ou usuário comum.

A Figura 6.12 apresenta a tela inicial do sistema. A tela consiste basicamente do nome do projeto na parte superior da tela, a logo do projeto Doce Desafio e os campos para se introduzir o usuário e senha para ter acesso ao sistema.

Para ter acesso ao sistema, o indivíduo deverá ser um administrador, ou seja, um acompanhante médico do indivíduo ou um coordenador devidamente capacitado e indicado pelo projeto Doce Desafio, ou poderá ser um usuário comum, que é o indivíduo que possui o aplicativo instalado em seu *smartphone* e possui dados armazenados nesse. Os usuários serão cadastrados no sistema pelo usuário administrador do sistema.



Figura 6.12: Tela de *Login*.

- Visão dos Usuários

A Figura 6.13 mostra a página apresentada após o acesso de um usuário comum (sem permissão de administrador). A tela contém uma mensagem de boas vindas e informações do perfil que acessou o sistema na parte superior, na faixa verde. O sistema apresenta um menu lateral com as opções de visualização das informações contidas no aplicativo, sendo essas alimentos, exercícios, glicemia, insulina e refeição.

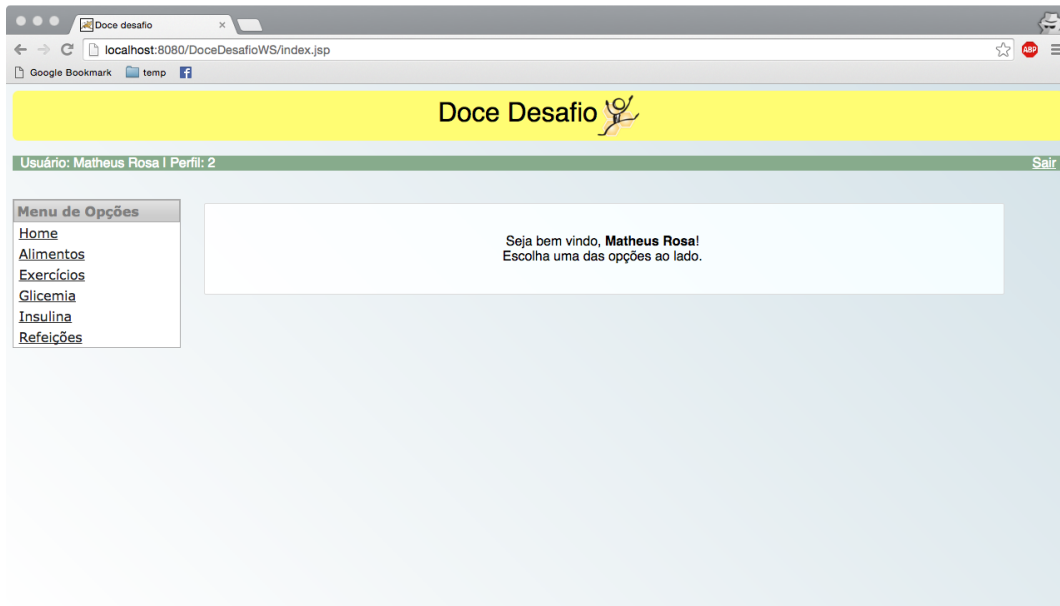


Figura 6.13: Tela de Boas Vindas.

A Figura 6.14 apresenta a página de registro de exercícios acessada por um usuário comum, ela apresenta uma tabela com todos os dados inseridos por esse usuário no aplicativo. Na primeira coluna tem-se o código da atividade, seguido pela sua descrição, o tipo (podendo ser aeróbico ou anaeróbico), as modalidades, a intensidade, a duração da atividade e a data.

Acima da tabela são exibidos campos dedicados aos filtros, permitindo pesquisar as atividades por tipo e por modalidade. O usuário possui a opção de excluir os itens cadastrados no ícone da lixeira.



Figura 6.14: Tela de Exercícios.

A Figura 6.15 apresenta a página de registro de insulinas acessada por um usuário comum. Essa tela apresenta todas as insulinas registradas por esse usuário no aplicativo. Na tabela é possível observar as colunas contendo o código da insulina, o tipo da insulina (podendo ser Basal e Ultra-rápida), seguido pela quantidade, descrição e data.

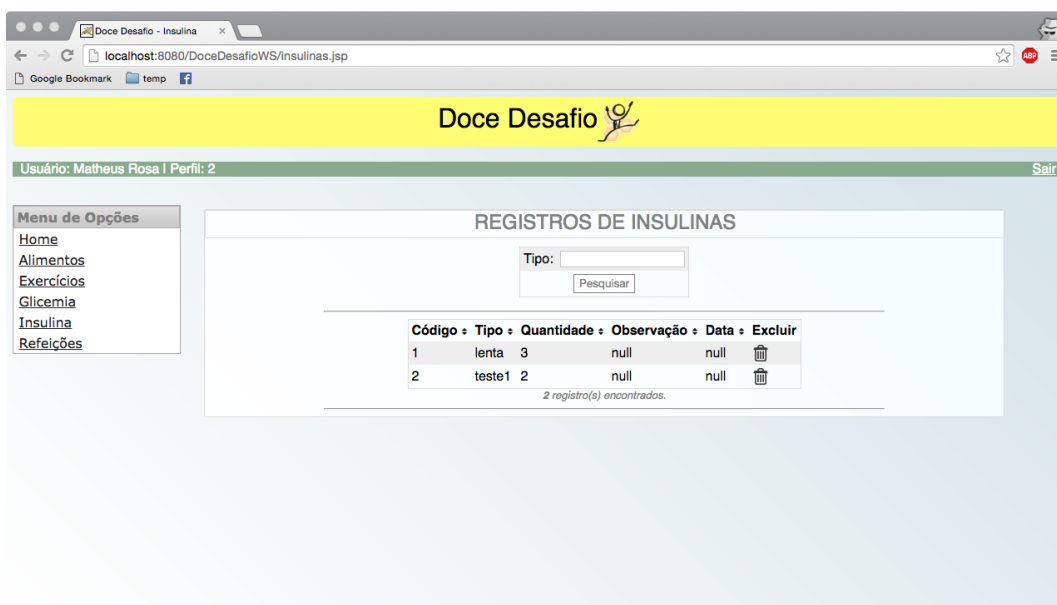


Figura 6.15: Tela de Insulinas.

A Figura 6.16 apresenta a tela de registro de glicemia quando acessada por um usuário comum. Nela pode-se observar a tabela contendo todos os registros de glicemia feitos por esse usuário no aplicativo.

A primeira coluna exibe o código do item, seguido pelo tipo, podendo ser pré-prandial ou pós-prandial, a data do registro, a medida, e por fim, as observações. Acima da tabela é exibido o campo para filtro da glicemia por tipo. O usuário tem a opção de excluir o item no ícone da lixeira.



Figura 6.16: Tela de Glicemias.

A Figura 6.17 apresenta a página de registro de refeições quando acessada por um usuário comum. Nela pode se observar a tabela contendo as refeições indicadas pelo usuário no aplicativo. Na primeira coluna é apresentado o tipo da refeição, podendo ser café da manhã, colação, almoço, lanche, janta e ceia, seguido pelas colunas contendo a observação a respeito da refeição, o peso da refeição, os carboidratos e a data.



Figura 6.17: Tela de Refeições.

- Visão do Administrador

Os usuários que acessam o sistema com o perfil de administrador possuem mais privilégios e permissão para realizarem mais operações que um usuário comum, sendo estas funções específicas para gerenciar o sistema, como cadastrar, alterar e listar usuários. Os usuários que acessam com perfil de usuário comum não possuem permissão para acessar essas funções.

Na Figura 6.18 é apresentada a tela inicial para os usuários que acessam o sistema com perfil de administrador. No menu lateral, é exibida uma nova seção chamada Administrativo com as opções de "Cadastrar Usuário" e "Listar Usuários".

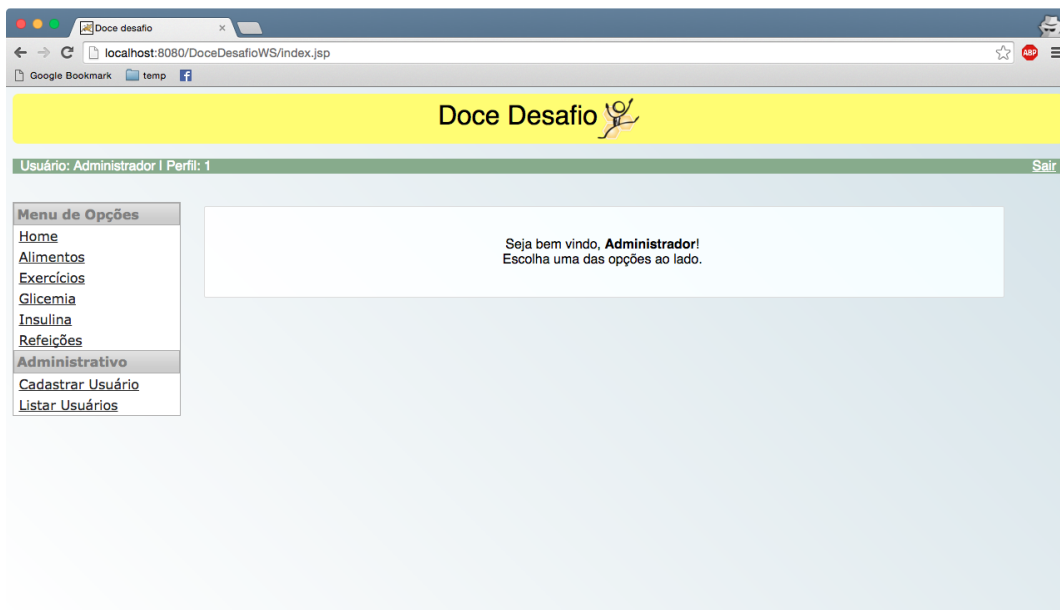


Figura 6.18: Tela do Administrador.

A página de alimentos apresentada na Figura 6.19, quando acessada como administrador, apresenta a tabela contendo os alimentos inseridos por todos os usuários, o que não é possível de ser feito com o perfil de usuário.

A diferença está na primeira coluna, na qual se apresenta o nome do paciente que ingeriu os alimentos, por sua vez, na segunda coluna é apresentado o código do alimento, já na terceira o nome do alimento cadastrado no banco de dados, a quarta contém a quantidade ingerida pelo paciente, a quinta possui o peso e a sexta coluna exibe a quantidade de carboidratos que o alimento possui.



Figura 6.19: Tela do Administrador para Consulta de Alimentos.

O sistema também possibilita a filtragem de dados, que pode ser feita tanto pelo nome do usuário como pelo nome do alimento. A Figura 6.20 apresenta um exemplo de filtro pelo nome do usuário.



Figura 6.20: Tela do Administrador usando Filtro de Dados.

A Figura 6.21 apresenta a página de exercícios quando acessado por um usuário administrador. Observa-se na tabela, todos os dados de todos os usuários do sistema.

A tabela apresenta as colunas usuário, código do exercício, descrição do exercício, tipo, modalidade, intensidade, duração e data. O administrador possui a opção de excluir o item no ícone da lixeira. Na parte superior também é possível filtrar os dados, sendo pelo nome do usuário, pelo tipo da atividade praticada ou pela modalidade.



Figura 6.21: Tela do Administrador para Consulta de Exercícios.

A Figura 6.22 apresenta a página de insulinas quando acessada por um usuário administrador. Nessa tela observa-se a tabela contendo todas as insulinas registradas por todos os usuários.

A primeira coluna mostra o nome do usuário que registrou a insulina, nas colunas seguinte são apresentados o código, o tipo da insulina aplicada, a quantidade aplicada, a observação e a data, respectivamente. É possível filtrar os dados pelo nome do usuário e por tipo, na parte superior da tabela.

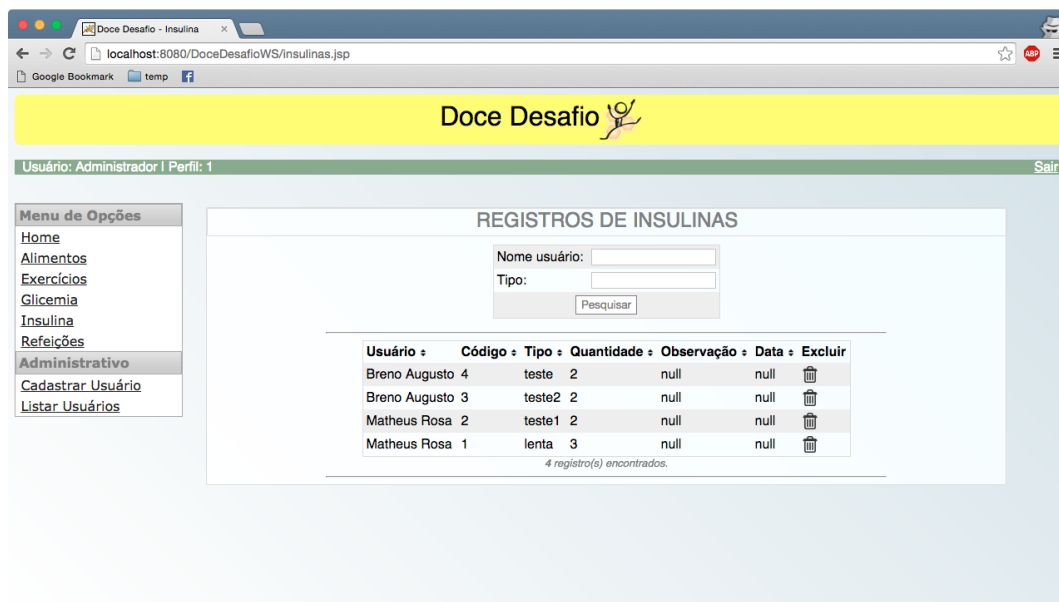


Figura 6.22: Tela do Administrador para Consulta de Insulinas.

A Figura 6.23 apresenta a página de glicemias quando acessada por um usuário administrador. Nessa tela observa-se a tabela contendo as glicemias de todos os usuários do

aplicativo. A primeira coluna especifica o nome do usuário, seguindo o código da glicemia, o tipo da glicemia, podendo essa ser pré-prandial ou pós-prandial, a data que a glicemia foi registrada, a medida da glicemia e por fim as observações.

Na parte superior da tela é possível filtrar os dados da tabela, esse filtro pode ser aplicado tanto pelo nome do usuário quanto pelo tipo da glicemia.



Figura 6.23: Tela do Administrador para Consulta de Glicemias.

A página de refeição apresentada na Figura 6.24, quando acessada como administrador possui a tabela contendo as refeições que foram registradas por todos usuários, por sua vez na segunda coluna é apresentado o tipo da refeição, podendo ser café da manhã, colação, almoço, lanche, janta ou ceia; na terceira uma observação da refeição, a quarta contém o peso, a quinta possui a quantidade de carboidratos que a refeição totalizou e por último a data em que a refeição foi ingerida.

O sistema também possibilita a filtragem de dados, que pode ser feita tanto pelo nome do usuário como pelo tipo da refeição. O usuário administrador pode também excluir os itens da tabela, pelo ícone de lixeira.



Figura 6.24: Tela do Administrador para Consulta de Refeições.

A tela apresentada na Figura 6.25 apresenta os campos necessários para se cadastrar um novo usuário no sistema, são eles: o nome do usuário, o *login* que o usuário preferir, a senha desejada para o usuário, o email e o perfil de acesso ao sistema, podendo este ser administrador ou usuário.

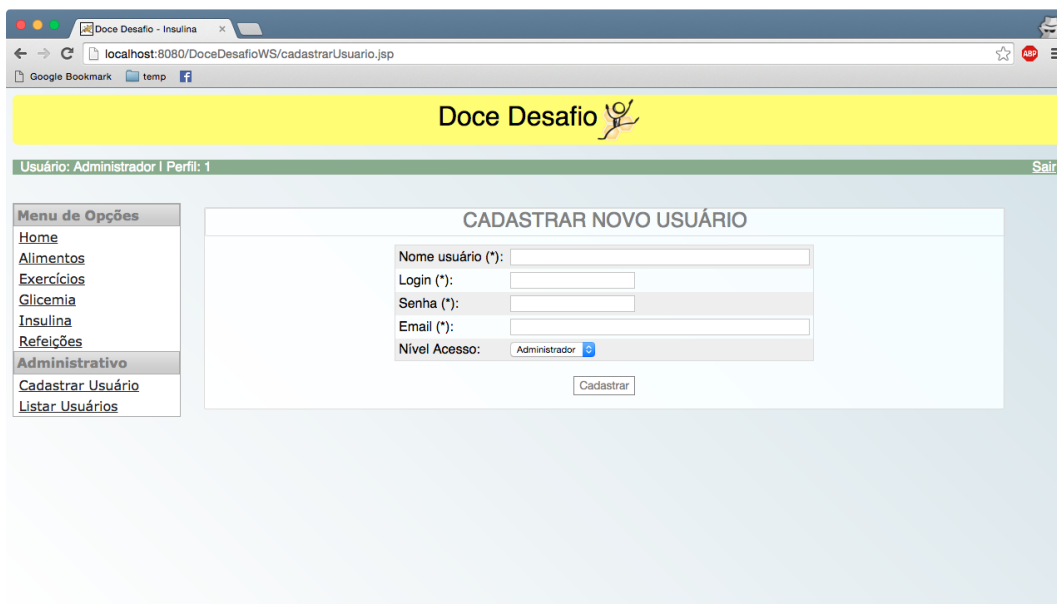


Figura 6.25: Tela do Administrador para Cadastro de Usuários.

O sistema possibilita a edição dos dados do usuário já cadastrado, podendo alterar qualquer um dos campos, sendo eles: nome, *login*, senha, email ou perfil de acesso ao sistema, conforme apresentado na Figura 6.26

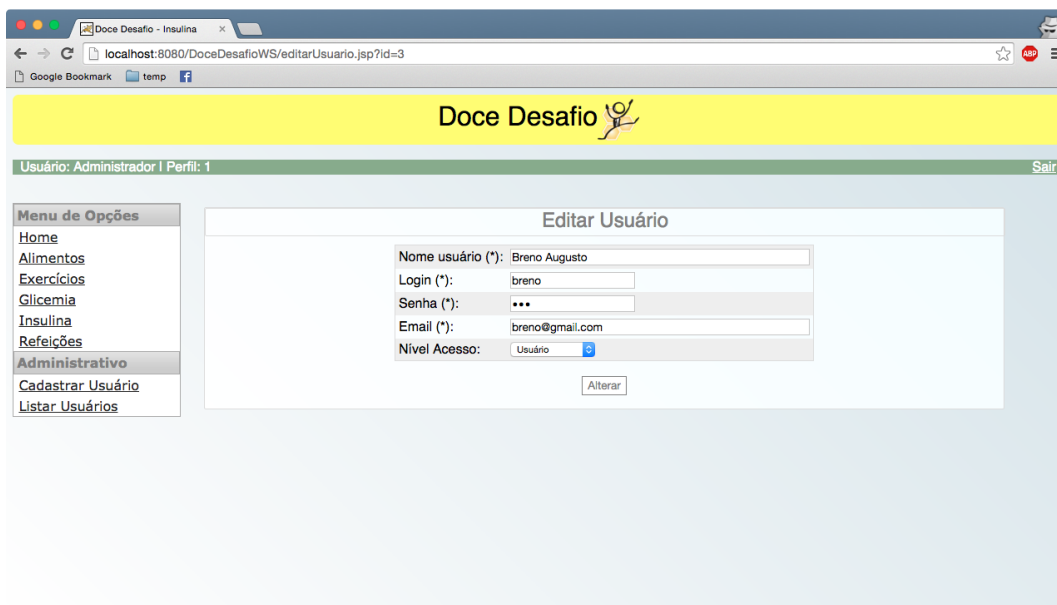


Figura 6.26: Tela do Administrador para Alteração de Usuários.

A tela exibida na Figura 6.27 apresenta a lista de usuários cadastrados, podendo ser feita essa listagem pelos campos do nome do usuário, pelo *login*, pelo email ou pelo nível de acesso. Caso todos os campos sejam deixados em branco, o sistema apresenta todos os usuários cadastrados no banco de dados.

Ao lado direito de cada usuário, são apresentadas as opções de alteração (ícone de lápis) ou remoção (ícone da lixeira).

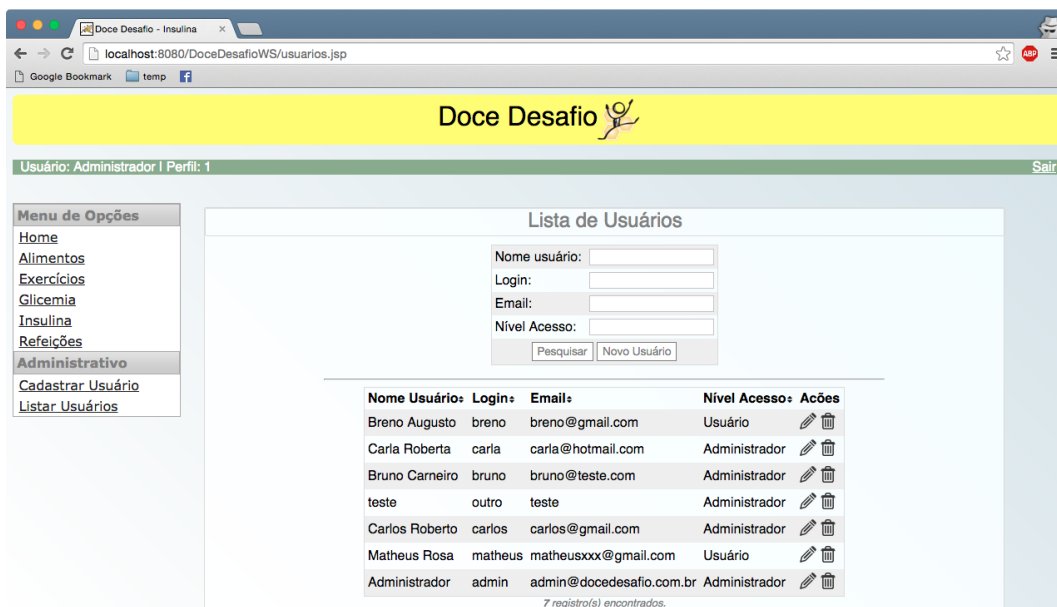


Figura 6.27: Tela do Administrador para Listagem dos Usuários.

6.2.2 Banco de Dados

Para o armazenamento dos dados utilizados pela aplicação web, utilizou-se o banco de dados PostgreSQL, pela sua confiabilidade e grande adoção no mercado.

Uma estrutura semelhante a do banco de dados utilizado no aplicativo foi criada para a aplicação web, com adição de uma nova tabela, sendo esse banco de dados composto por onze tabelas e dez relacionamentos como mostra a Figura 6.28.

Como o banco de dados da aplicação web suporta o armazenamento dos dados de diferentes usuários, a lógica dos relacionamentos é de muitos para muitos, assim as tabelas referentes aos dados inseridos precisaram se relacionar com a tabela 'login' para que ficassem vinculadas a conta de cada usuário.

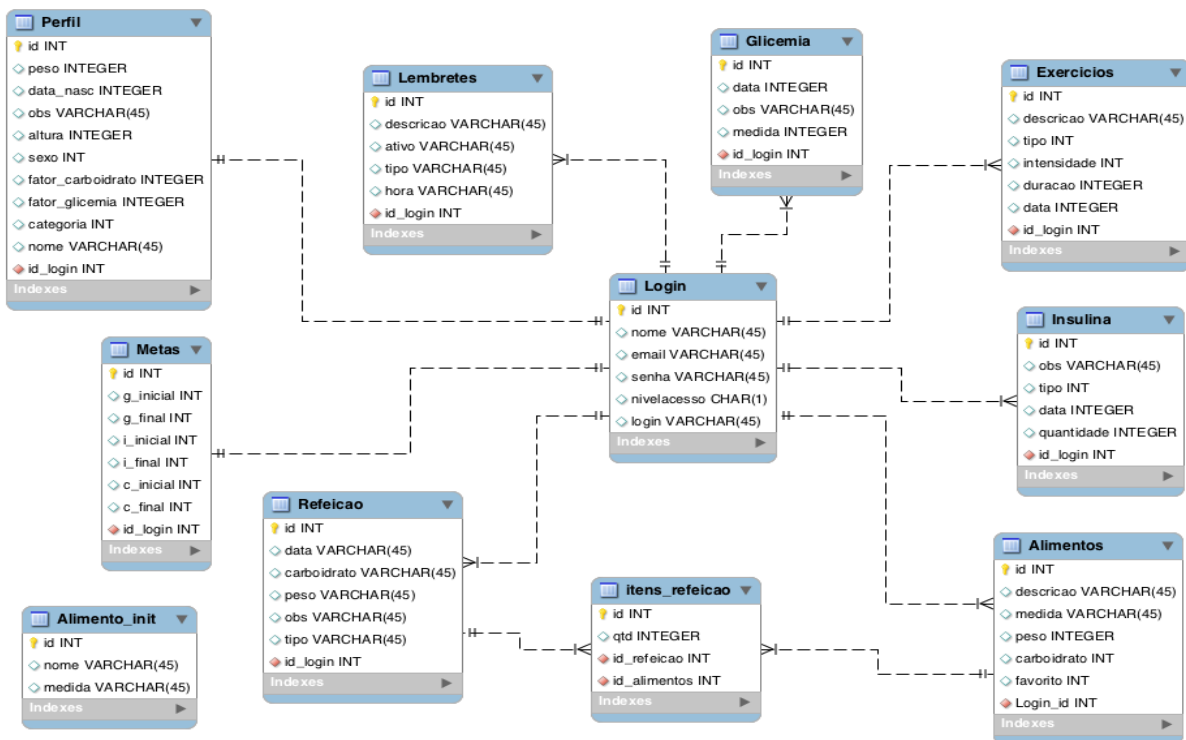


Figura 6.28: Banco de Dados Utilizado na Aplicação Web.

A nova tabela 'alimento-init' é composta de um conjunto de dados previamente definidos relativos a alimentos e seus valores nutricionais, de acordo com a Sociedade Brasileira de Diabetes. Esses dados são disponibilizados assim que uma conta de usuário é criada e são sincronizados com o aplicativo assim que o usuário faz o login. Alimentos inseridos pelo próprio usuário são inseridos na tabela "alimento".

6.3 Integração via *Web Services*

Para realizar a integração do aplicativo com a aplicação web, foi necessário implementar um mecanismo de sincronização que permitisse que, assim que os dados fossem

registrados no aplicativo, eles também estivessem disponíveis na aplicação web, para isso utilizou-se a tecnologia *web services*.

Esse sincronização pode ser feita pelo aplicativo tanto *online* quanto *offline*, especificando cada uma delas:

- Online: Para que essa sincronização seja possível, o *smartphone* deve estar conectado a uma rede com acesso a internet, de modo que os dados possam ser transferidos para o banco de dados remoto da aplicação web. No código do aplicativo existe uma tarefa que é executada por todas as telas do aplicativo, nela é verificada a conexão com a internet e os serviços que serão requisitados ao servidor. Esses serviços são divididos em operações de inserção, atualização e exclusão de tabelas. Caso algum desses serviços sejam acionados pelo aplicativo, é realizado um espelhamento dessa ação no banco de dados global da aplicação web.

Existe também uma classe de conversão de dados, onde nesse é interpretado os objetos recebidos via web pela conectividade REST e os traduzem em objetos do aplicativo *Android*.

- Offline: Quando o *smartphone* não estiver conectado a internet, os dados permanecerão armazenados no aplicativo com uma sinalização de espera por sincronização, e assim que uma conexão com a internet estiver disponível o aplicativo realiza a sincronização automática e envia os dados para o banco de dados remoto. Esses dados são gravados na tabela 'SyncREST' do aplicativo, que possui os campos id, tipo da tabela (glicemia, alimentos, exercícios, insulina, refeição) que houve a alteração, código do dado modificado, e da operação feita que é simbolizada como '1' para inclusão, '2' para alteração e '3' para exclusão. Quando se retorna a conectividade com a internet, essa tabela é percorrida, realizando as operações nos dados com códigos registrados em suas respectivas tabelas. A tabela 'SyncREST' pode ser vista na figura 6.29.

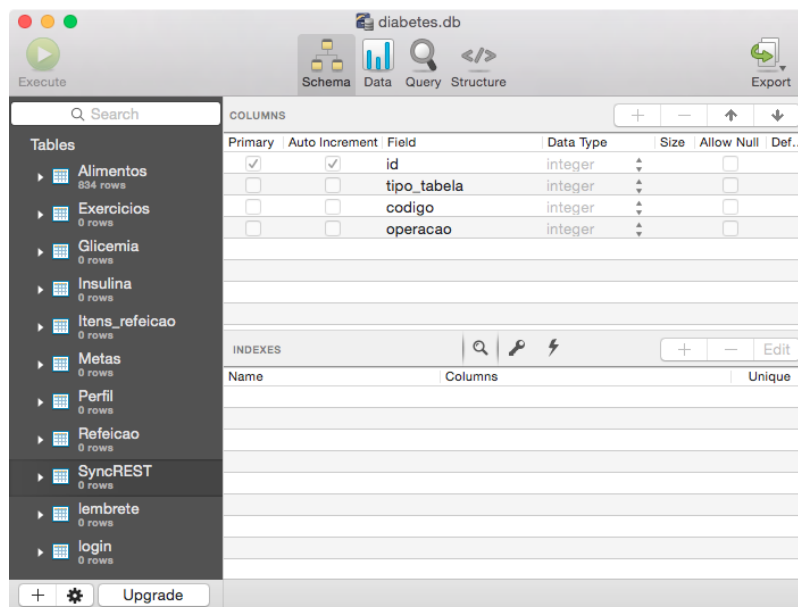


Figura 6.29: Tabela para sincronização offline.

Para a aplicação web os dados são sincronizados da seguinte forma:

6.4 Considerações Finais

Neste capítulo foi abordada a proposta deste trabalho: o desenvolvimento do sistema integrado Doce Desafio, composto por uma versão do aplicativo Doce Desafio que contém melhorias e novas funcionalidades e uma aplicação web. Foi também apresentada a metodologia utilizada para realizar a integração do sistema por meio da tecnologia *web services*.

Foram descritas e exibidas as mudanças gráficas e lógicas implementadas na segunda versão do aplicativo Doce Desafio, apresentada a estrutura da aplicação web juntamente com todas suas páginas e funcionalidades e também a lógica de sincronização dos dados do sistema.

O sistema buscou atender às necessidades levantadas nas entrevistas com os usuários do aplicativo e as solicitações feitas pela equipe responsável pelo projeto Doce Desafio, através de uma interface simples e intuitiva, que seja de fácil utilização por qualquer pessoa.

Capítulo 7

Conclusão e Trabalhos Futuros

7.1 Conclusão

Neste trabalho foi proposto e implementado o Sistema Integrado Doce Desafio, composto pela segunda versão do aplicativo *Android* e uma aplicação web, que se integram via *web services*. Ambos foram voltados para portadores de diabetes tipo 1 e profissionais de saúde que atuam no tratamento da doença, a fim de auxiliar em seu controle e tratamento. Foram expostas e detalhadas as melhorias e otimizações desenvolvidas para o aplicativo *Android*, assim como o desenvolvimento por completo da aplicação web e de sua integração com o aplicativo através da tecnologia *web services*.

Com a nova versão do aplicativo, pode-se obter dados a respeito do tipo de glicemia e do tipo de exercício, possibilitando um controle mais detalhado dessas opções. A nova versão possibilita também a exportação dos dados em formato CSV, o que ajuda na redundância desses dados para *backups* de segurança, contado com a vantagem de que o formato CSV é reconhecido por diversos programas como por exemplo, Excel, LibreOffice e Planilhas Google. Por fim, os dados armazenados no aplicativo, agora podem ser acessados online, o que auxilia o corpo clínico no acompanhamento dos pacientes.

O sistema web beneficia duas partes, os responsáveis pelo paciente, que possuem uma forma de acompanhar todos os usuários do aplicativo, pois com contas de administrador terão privilégios como, cadastrar novos usuários, deletar usuários, visualizar os respectivos dados de qualquer usuário. Assim como, os usuários que não possuem privilégio de administrador, terão uma forma de manter todos os seus dados seguros e disponíveis, possibilitando o acesso aos dados em qualquer computador com acesso a internet.

O trabalho se embasou nas características da diabetes *mellitus* e como seus sintomas e tratamento interferem diretamente na rotina dos indivíduos portadores da doença. Com essas informações, pôde-se compreender melhor as dificuldades desses pacientes e como o sistema web poderia ser uma ferramenta de auxílio no controle da doença.

Do ponto de vista técnico computacional, é importante ressaltar que o sistema Doce Desafio foi construído por meio de ferramentas de código livre ou *open-source*. Ele é portátil e permite a agregação de novos módulos futuramente, visto que o código foi mantido extensível e modularizado.

Finalizado o desenvolvimento deste trabalho, obteve-se um sistema robusto e complexo tecnicamente, que demandou grandes conhecimentos, envolvendo programação para dispositivos móveis, programação *web* e implementação da arquitetura SOA por meio de

web services, tendo sido um grande desafio o aprendizado de todos esses conceitos e sua aplicação prática.

Não foi possível realizar testes de validação com os usuários finais do sistema, de forma que pudesse ser atestada sua eficácia no atendimento aos requisitos especificados e verificado seu funcionamento em diferentes condições de uso. Espera-se que o sistema desenvolvido possa auxiliar os usuários portadores de diabetes, beneficiando-os no controle de sua saúde e propiciando um recurso adicional para o seguimento de seu tratamento pelos profissionais de saúde, resultando em um acompanhamento mais próximo e periódico de seus pacientes.

7.2 Trabalhos Futuros

O sistema integrado Doce Desafio evoluiu de um único aplicativo para uma solução mais completa, um conjunto de softwares, incluindo uma aplicação web robusta. Tanto o aplicativo quanto a aplicação web ainda possuem aspectos que podem ser evoluídos. A seguir são elencadas algumas sugestões de melhorias e evoluções para trabalhos futuros:

- Com a criação de novas versões do aplicativo, seria interessante a concepção de um novo *layout*, podendo-se criar uma identidade visual que conectasse o sistema web, o aplicativo e a sede física do projeto.
- Atualmente, a aplicação web é uma ferramenta auxiliar para os profissionais de saúde acompanharem os pacientes com diabetes tipo 1, possibilitando filtrar os dados inseridos e criar relatórios e gráficos para visualizar o progresso do paciente. Contudo, a aplicação web não possibilita inserir novos dados ou editar os existentes, o que poderia torna-la mais robusta.
- Apesar do sistema ter prezado pela usabilidade, não existe um manual que especifique e ensine a utilizar suas funcionalidades; portanto, a criação de um manual e vídeos demonstrativos auxiliaria os usuários iniciantes a utilizarem os recursos disponíveis.
- Disponibilizar o aplicativo Doce Desafio de forma gratuita, por meio da loja de aplicativos do Android, *Play Store*[8].

Referências

- [1] Estimativas e projeções mundiais de dm no periodo de 1994-2010. <http://dtr2001.saude.gov.br>, 2010. [Online; acessado 6-Janeiro-2016]. 6
- [2] About sqlite. <http://www.sqlite.org/about.html>, 2013. [Online; acessado 6-Janeiro-2016]. 37
- [3] How android was born? ... and today, what? <http://www.android.com/history/>, 2014. [Online; acessado 6-Janeiro-2016]. 17
- [4] Adndroid dashboards. <http://https://source.android.com/devices>, 2015. [Online; acessado 6-Janeiro-2016]. vii, 18, 19, 21
- [5] Gartner says worldwide smartphone sales recorded slowest growth rate since 2013. <http://www.gartner.com/newsroom/id/3115517>, 2015. [Online; acessado 6-Janeiro-2016]. ix, 16
- [6] International diabetes foundation atlas. <http://www.diabetesatlas.org>, 2015. [Online; acessado 6-Janeiro-2016]. 7
- [7] Pesquisa nacional de saude. <http://www.ibge.gov.br/home/estatistica/populacao/pns/2013>, 2015. [Online; acessado 6-Janeiro-2016]. 6
- [8] Play store. <https://play.google.com/store>, 2015. [Online; acessado 6-Janeiro-2016]. 35, 61
- [9] Sanofi diabetes prevenção e causas. <http://www.starbem.com.br/Diabetes/PrevencaoCausas>, 2015. [Online; acessado 6-Janeiro-2016]. 11
- [10] Sociedade brasileira de diabetes. <http://www.diabetes.org.br/para-o-publico/tudo-sobre-diabetes>, 2015. [Online; acessado 6-Janeiro-2016]. 9, 10, 11
- [11] Web services architecture. <http://www.w3.org/TR/ws-arch/>, 2015. [Online; acessado 6-Janeiro-2016]. 30
- [12] World wide web consortium (w3c). <http://www.w3.org>, 2015. [Online; acessado 6-Janeiro-2016]. 17, 31
- [13] American Diabetes Association. Aspectos Práticos de Insulina, um Manual para Prescritores. *Diabetes Care 2008*, 31(1)(xiii,17,19,20):12–54, 2003. 1, 8
- [14] American Diabetes Association. Standards of Medical Care in Diabetes - 2008. *Diabetes Care 2008*, 31(1)(xi,9,10,11,12):12–54, 2008. 6

- [15] Len Bass. *Software Architecture*. Pearson, 2012. 15
- [16] Edison M. de Carvalho e Jair R. Fernandes. Doce Desafio - Aplicativo para Dispositivos Móveis para o Controle e a Monitoração do Diabetes Tipo 1, 2013. Monografia, Departamento de Ciência da Computação, Universidade de Brasília. vii, 1, 2, 12, 35, 36, 37, 38, 46
- [17] Ken Dunham, Shane Hartman, Manu Quintans, Jose Andre Morales, and Tim Strazere. *Android Malware and Analysis*. CRC Press, 2014. 19
- [18] Maria L. Zanetti e Isabel Amélia C. Mendes. Análise das Dificuldades Relacionadas às Atividades Diárias de Crianças e Adolescentes com Diabetes Mellitus tipo 1: Depoimento de Mães. *Latino-am Enfermagem*, 9(6):25–30, 2001. 7
- [19] Daniela B. Marcelino e Maria D. Barross Carralho. Reflexões sobre Diabetes Tipo 1 e sua Relação com o Emocional. *Psicologia: Reflexão e Crítica*, 18(1):72–77, 2005. 5, 7, 10
- [20] Daniela B. Marcelino e Maria D. Barross Carralho. Avanços Farmacológicos no Tratamento de Diabetes tipo 2. *Brazilian Journal of Surgery and Clinical Research*, 8(3):72–78, 2014. 12
- [21] Nikola Milanovic e Miroslaw Malek. Current Solutions for Web Service Composition. *ACM*, 8(6):51–59, 2004. 29
- [22] Harvey M. Deitel e Paul Deitel. *Java: Como Programar*. Pearson, 2010. 22, 24
- [23] Thomas Erl. *Soa - Princípios de Design de Serviços*. Pearson, 2009. vii, 26, 27, 30
- [24] Thomas Erl. *SOA with REST: Principles, Patterns Constraints for Building Enterprise Solutions with REST*. Prentice Hall, 2012. 33
- [25] Jorge L. Gross, Sandra P. Silveiro, Joíza L. Camargo, Angela J. Reichelt, and Mi-rela J. de Azevedo. Diabetes Melito: Diagnóstico, Classificação e Avaliação do Controle Glicêmico. *Arq Bras Endocrinol Metab*, 46(1):16–26, 2001. 7, 11
- [26] Dr. James R. Garvin III. *Power to Prevent*. National Diabetes Education Program, 2005. 8, 9, 10
- [27] Nicolai M Josuttis. *Soa na Prática*. Alta Books, 2008. 26
- [28] Hilary King, Ronald E. Aubert, and Willian H. Herman. Global Burden of Diabetes, 1995-2015: Prevalence, numerical estimate, and projections. *American Diabetes Association*, 21(9):1414–1431, 1998. vii, 7
- [29] José Carlos Lazzeri. *Arquitetura Orientada a Serviços: Fundamentos e Estratégias*. Ciência Moderna, 2009. 28
- [30] Antonio C. Lopes. *Diagnostico e Tratamento*. Manole, 2006. 1, 5, 12
- [31] Eric Newcomer. *Understanding Web Services - XML, WSDL, SOAP, and UDDI*. Addison Wesley, 2002. 30, 31, 32

- [32] João E. Nicoluzzi, Carlos W. Mormanillo, and João Carlos D. Repka. Transplante Simultâneo de Pâncreas-Rim em Portador de Diabetes Mellitus Tipo 1 com Insuficiência Renal Crônica. Experiência Inicial do Hospital Angelina Caron. *Arq Bras Endocrinol Metab*, 47(3):243–247, 2003. 6, 8
- [33] Zimmert P., Alberti KG., and Shaw J. Global and Societal Implications of the Diabetes Epidemic. *Nature* 2001, Dec 13(414(6865)):782–7, 2001. vii, 6
- [34] Lúcio Camilo O. Pereira. *Android para Desenvolvedores*. Brasport, 2012. 17, 19
- [35] Roger S. Pressman. *Engenharia de Software*. Mcgraw-hill, 2007. 14, 15
- [36] Ian F. Sommerville. *Engenharia de Software*. Pearson, 2008. 13, 14, 15
- [37] Aaron E. Walsh. *The Web Services Specification Reference Book*. Prentice Hall, 2002. 30