

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

DANIEL DOUGLAS VIANA PINHEIRO

**PROPOSTA DE INTERFACE GRÁFICA PARA RASTREAMENTO DE DIABATES
DO TIPO 2 POR MÉTODO NÃO INVASIVO**

São Luís

2018

DANIEL DOUGLAS VIANA PINHEIRO

**PROPOSTA DE INTEFACE GRÁFICA PARA RASTREAMENTO DE DIABATES
DO TIPO 2 POR MÉTODO NÃO INVASIVO**

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Estadual do Maranhão para o grau de bacharelado em Engenharia de Computação.

Orientadora: Áurea Celeste da Costa Ribeiro
Co-Orientador: Denner Araújo Costa

São Luís
2018

Pinheiro, Daniel Douglas Viana.

Proposta de interface gráfica para rastreamento de diabetes do tipo 2 por método não invasivo / Daniel Douglas Viana Pinheiro.– São Luís, 2018.

56 p.

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2018.

Orientador: Profa. Dra. Áurea Celeste da Costa Ribeiro.

1.Diabetes. 2.Rastreamento. 3.Aplicativo. I.Título

CDU: 004.92:616.379-008.64

DANIEL DOUGLAS VIANA PINHEIRO

**PROPOSTA DE INTEFACE GRÁFICA PARA RASTREAMENTO DE DIABATES
DO TIPO 2 POR MÉTODO NÃO INVASIVO**

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Estadual do Maranhão para o grau de bacharelado em Engenharia de Computação.

Trabalho aprovado, São Luís, 9 de julho de 2018:

Profa. Dr^a. Áurea Celeste da Costa Ribeiro
Orientadora

Denner Araújo Costa (Co-orientador)
Avaliador

Espec. Ana Paula Ferreira Costa
Avaliador

LISTA DE ILUSTRAÇÕES

Figura 1: Países com maior número de diabéticos em 2013 (20-79 anos).....	12
Figura 2: Diagrama de Caso de Uso do Sistema	17
Figura 3: Tela Inicial	21
Figura 4: Tela de Cadastro.....	22
Figura 5: Formulário do Coletor.....	23
Figura 6: Tela Inicial com botões.....	24
Figura 7: Corrigindo erros	25
Figura 8: Tela de Confirmação de Dados.....	26
Figura 9: Tela com a resposta do servidor	27
Figura 10: Tela de Autenticação do Firebase.....	28
Figura 11: Enviando dados de teste para o servidor no formato JSON	29
Figura 12: Modelo de processo de engenharia de requisitos	32
Figura 13: Exemplo de Caso de Uso.....	34
Figura 14: Os três princípios do Material Design.....	39
Figura 15: Espaço 3D com eixos x, y e z.	40
Figura 16: Elevação do Material Design.....	41
Figura 17: Sombra no Material Design.....	42
Figura 18: Exemplo do uso de cores.....	43
Figura 19: Exemplo de Ícone de Produto	44
Figura 20: Ícones do sistema.	44
Figura 21: Fonte Roboto	45
Figura 22: Fonte Noto	45
Figura 23: Exemplo de Formato JSON	47
Figura 24: Resposta do servidor com a classificação do paciente	49

LISTA DE SIGLAS

ADA - American Diabetes Association
AVC - Acidente Vascular Cerebral
CAD - Computer Aided Design
DATASUS - Departamento de informática do Sistema Único de Saúde
DM - Diabetes Mellitus
DM1 - Diabetes tipo 1
DM2 - Diabetes tipo 2
DVP - Doença Vascular Periférica
ER - Engenharia de Requisitos
FID - Federação Internacional de Diabetes
HTTP - Hypertext Transfer Protocol
ICA - Independent Component Analysis
IMC - Índice de Massa Corporal
NoSQL - Not Only SQL
OMS - Organização Mundial de Saúde
REST - REpresentational State Transfer
RF - Requisitos Funcionais
RNF - Requisitos Não-Funcionais
SDK - Software Development Kit
SVM - Support Vector Machine
XML - eXtensible Markup Language

AGRADECIMENTOS

Agradeço aos meus pais e familiares pela educação que estive e me guiar pelo caminho correto.

Agradeço aos meus amigos e colegas pelo convívio, amizade e apoio.

A esta universidade, todo seu corpo docente, em especial aos professores por passar seus conhecimentos e contribuindo na minha formação profissional.

RESUMO

O rastreamento de Diabetes Mellitus do tipo 2 é um mecanismo extremamente importante por causa do enorme crescimento da doença no mundo. Pesquisas relatam que há 422 milhões de diabéticos no planeta e grande parte desconhece sua situação. Este problema dificulta seu tratamento e muitos pacientes doentes já manifestam alguma complicação em virtude da sua falta deste nos estágios iniciais da diabetes. Este estudo propõe um aplicativo para o uso de método não invasivo para o rastreamento de diabetes tipo 2, com o objetivo de reduzir os custos que órgãos de saúde dedicam a essa anomalia. A aplicação deve ter recursos de armazenagem de dados off-line, seguir diretrizes de design, deve fazer comunicação com o servidor que irá processar as informações pelo método dos componentes independentes e máquinas de vetor de suporte e vai retornar ao aplicativo se o paciente é pré-diabético, não diabético ou diabético.

Palavras-chave: 1. Diabetes. 2. Rastreamento. 3. Aplicativo.

ABSTRACT

Screening for Type 2 Diabetes Mellitus is an extremely important mechanism because of the enormous growth of the disease in the world. Research reports that there are 422 million diabetics on the planet and most are unaware of their situation. This problem hinders their treatment and many patients, is already compromised by virtue of their lack, in the early stages of diabetes. This study proposes an application for the use of non-invasive method for the development of type 2 diabetes, with the objective of reducing health costs. The application has offline data storage capabilities, follow design guidelines, develop communication with the server that is processed as information by the independent components method and support vector machines and will return to the application if the patient is pre- diabetic, non-diabetic or diabetic.

Keywords: 1. Diabetes, 2. Screening, 3. Application.

SUMÁRIO

1. INTRODUÇÃO	11
1.1 Contextualização.....	11
1.2 Motivação	13
1.3 Questão da Pesquisa.....	14
1.4 Objetivos.....	14
1.4.1 Objetivos Específicos.....	14
1.5 Organização do Trabalho	14
2. METODOLOGIA	15
2.1 Diagrama de Casos de Uso	15
2.2 Cenário.....	17
2.3 Projeto.....	19
2.3.1 Diagnóstico e Rastreamento de Diabetes.....	19
2.3.2 Telas do Aplicativo	20
2.3.3 Utilização do Firebase	27
2.3.4 Comunicação do Cliente-Servidor	28
3. FUNDAMENTAÇÃO TEÓRICA.....	29
3.1 Engenharia de Requisitos	29
3.1.1 Requisitos Funcionais e Não Funcionais	30
3.1.2 <i>Stakeholders</i>	31
3.1.3 Processo da Engenharia de Requisitos	31
3.1.4 Casos de Uso	34
3.1.5 Cenário	34
3.2 Análise de Componentes Independentes	35
3.3 Máquinas de Vetor de Suporte.....	36
3.4 Descrição das bases de dados estudadas	37
3.5 Material Design.....	38

3.5.1 Os três princípios	38
3.5.2 O ambiente do Material Design.....	39
3.5.3 Elevação e Sombra.....	40
3.5.4 Sistema de cores do Material Design	42
3.5.5 Ícones	43
3.5.6 Tipografia.....	45
3.6 Firebase	46
3.7 Formato JSON.....	46
3.8 REST	47
4. RESULTADOS E DISCUSSÃO	49
5. CONSIDERAÇÕES FINAIS	50
5.1 TRABALHOS FUTUROS	50
5.2 RESULTADOS ALCANÇADOS	51
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	52
GLOSSÁRIO	56

1. INTRODUÇÃO

Este Capítulo apresentará a contextualização do trabalho, os motivos para a implementação. Depois sobre a questão da pesquisa, objetivos e a organização do trabalho.

1.1 Contextualização

Diabetes Mellitus (DM) é uma doença resultante da produção insuficiente de insulina, que é um hormônio produzido pelo pâncreas com a função de fazer o metabolismo da glicose para a produção de energia (AMERICAN DIABETES ASSOCIATION, 2007). A falta de insulina ocasiona elevados níveis de glicose sanguínea, que causa prejuízo em diversos sistemas do organismo (LEAHY, 2005), mudando o metabolismo de proteínas, carboidratos e lipídios.

É uma doença que não tem cura e traz diversas complicações quando não é tratada. Ela apresenta sintomas comuns como sede, poliúria, visão turva, perda de peso e poligafia. Quando está em situação grave ocorre a cetoacidose, que na falta de tratamento eficiente, resulta em hálito acético, coma e morte (DAVIDSON, 2001).

Os dados publicados pelos órgãos mundiais são bastante preocupantes pelo aumento de diabéticos pelo mundo, o qual cresce cada vez mais. Segundo a Organização Mundial de Saúde (OMS), em 1980 a 2014, o número de pessoas com diabetes quadruplicou alcançando 422 milhões de pessoas, principalmente em países desenvolvidos. Recentemente, a Federação Internacional de Diabetes (FID) divulgou que 425 milhões de adultos possui a doença e daqui a 28 anos o número poderá chegar a 629 milhões.

Existem três tipos principais de diabetes: o diabetes tipo 1 (DM1), o diabetes tipo 2 (DM2) e o diabetes gestacional. O DM1 origina-se da pouca produção insatisfatória de insulina pelo pâncreas, alguns dos seus sintomas resultam em perda de peso, alterações da visão e fadiga. O DM2 é quando o organismo faz o uso ineficiente da insulina, ou não funciona corretamente, fazendo aumentar a quantidade de glicose no sangue, esse tipo compreende a maioria das pessoas com diabetes do mundo e deriva-se do excesso de peso e falta de atividade física. E o diabetes gestacional ocorre quando uma mulher grávida sem diabetes demonstra níveis elevados de glicose no sangue durante a gravidez (WHO | Diabetes, 2017).

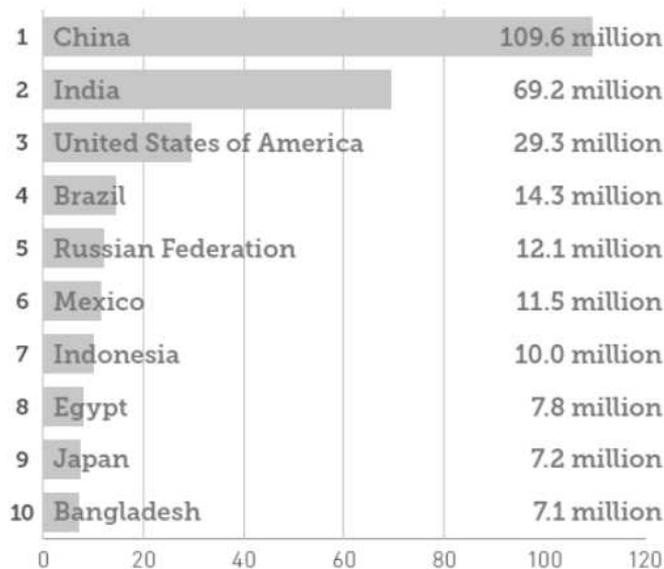
A presença da Diabetes Mellitus nas pessoas está ligada ao envelhecimento da população, da lentidão de avanços terapêuticos da doença e ao atual estilo de vida, marcado pelo sedentarismo e na aglomeração de gordura por maus hábitos de alimentação.

A regularidade das complicações crônicas do DM2 diversifica de acordo com as populações estudadas. Esses pacientes tem a tendência de duas a quatro vezes maior de morrer por doença cardíaca do que aos não diabéticos, e quatro vezes de chance contrair a Doença Vascular Periférica (DVP) e Acidente Vascular Cerebral (AVC). Entre os motivos incluídos na etiologia das complicações crônicas do DM2, são a hiperglicemia, a hipertensão arterial sistêmica, a dislipidemia e o tabagismo (SCHEFFEL, BORTOLANZA, *et al.*, 2004)

O DM traz muitos transtornos sociais vistos como custos indiretos para a sociedade e o indivíduo enfermo, como: morte prematura, incapacidades, absenteísmo, a redução do retorno da educação oferecida ao indivíduo, diminuição da renda do chefe de uma família, aumento de aposentadorias precoces e desemprego.

A Figura 1 abaixo apresenta os dados dos países que possui grande número de diabéticos no ano de 2013, entre as idades de 20 e 79 anos.

Figura 1: Países com maior número de diabéticos em 2013 (20-79 anos).



Fonte: (IDF DIABETES ATLAS, 2013)

Os custos associados a essa anomalia são altos, estimou-se que o diabetes causou pelo menos US \$ 673 bilhões em despesas de saúde em 2015, onde 12% desses gastos foram com adultos (International Diabetes Federation - What is Diabetes).

A ADA (AMERICAN DIABETES ASSOCIATION, 2013) comparou as estimativas dos custos totais do diagnóstico entre os anos de 2007 e 2012 nos Estados Unidos. Em 2007, estima-se que foram gastos U\$ 174 bilhões de dólares e enquanto em 2012, o valor foi U\$ 245 milhões de dólares, representando 41% nos custos totais em 5 anos.

No Brasil, um estudo feito avaliando o impacto sobre o sistema público de saúde brasileiro apresentou que, entre os anos de 1999 e 2001, 2,2% dos gastos do Ministério da Saúde foram por hospitalizações por diabetes (ROSA, SCHMIDT, *et al.*, 2007). O DM2 e a hipertensão arterial estão comprometidos na maior parte dos casos de internações, sendo encarregado por 7,4% das admissões não relacionadas à gestação e por 9,3% de custos com assistência hospitalar (BRASIL, 2009).

1.2 Motivação

Com o aumento exponencial dos custos de diabetes pelo mundo, encontrar uma maneira confiável com poucos gastos para rastreamento de diabetes e que seja acessível para qualquer pessoa de uma classe social, é a principal motivação para a implementação desse trabalho.

Ao invés de utilizar métodos invasivos, ou seja, coletando o sangue do paciente em jejum de 8 horas e depois analisar seu nível de glicemia. Será utilizado métodos não invasivos por serem mais fáceis de serem obtidos, que são dados como altura, idade, pressão sanguínea. Dessa forma, dentro alguns segundos, o resultado já é obtido.

Toda a infraestrutura médica para a coleta do sangue será substituída por um computador apenas entrando com os marcadores não invasivos.

Além dessa abordagem ser de baixo custo e confiável, a identificação precoce da DM2 é a razão primordial para o êxito do tratamento da doença, pois segundo (DIBETICOOL, 2016), o doente irá fazer menos visita no hospital para corrigir os problemas de saúde.

1.3 Questão da Pesquisa

Inicialmente abrimos o tema sobre o Diabetes, o que contribui para o seu crescimento no mundo, suas complicações e como identificar e tratar essa doença.

Os dados coletados das pessoas como o peso, altura, pressão arterial, índice de massa corporal (IMC) são proveitosos para o combate da diabetes. Os mesmos, em pessoas com diabetes são tidos como marcadores da doença porque expõem padrões que a doença manifesta em múltiplos pacientes.

Esses marcadores são divididos em invasivos ou não invasivos. Os Invasivos são descobertos por meio do exame de sangue. Os não invasivos são obtidos por meio de informações como por exemplo altura, pressão arterial, histórico de diabetes na família.

Como implementar uma interface de fácil usabilidade para o rastreamento de diabetes no paciente com a coleta de marcadores não invasivos?

1.4 Objetivos

O objetivo deste trabalho é fazer a implementação do lado *front-end* da aplicação que coleta informações das pessoas para identificar ou rastrear DM2. O projeto de pesquisa fazer a identificação precoce da DM2, utilizando os dados que os agentes de saúde que irão coletar pelos *tablets* com sistema operacional *Android*, na cidade de São Luís.

1.4.1 Objetivos Específicos

- Construção do aplicativo com modernas diretrizes de design;
- A usabilidade do aplicativo deve ser de fácil compreensão;
- Capacitar o armazenamento de dados dentro do dispositivo caso não há internet;
- Fazer a comunicação com o servidor.

1.5 Organização do Trabalho

Este TCC é organizado em seis capítulos:

O **CAPÍTULO 2** relata a toda Metodologia e desenvolvimento do projeto.

O **CAPÍTULO 3** redige a Fundamentação Teórica constituído basicamente de Engenharia de Requisitos, Diagnóstico e Rastreamento de diabetes, Máquinas de

Vetor de Suporte, Análise de Componentes Independentes, Banco de Dados Não Relacional, Material Design, Firebase e REST.

O **CAPITULO 4** apresenta os Resultados e Discussões da atual fase da pesquisa.

O **CAPITULO 5** expõe as Considerações Finais e Trabalhos da proposta do TCC.

O **CAPITULO 6** é reservado para Referencias Bibliográfica do trabalho.

2. METODOLOGIA

A metodologia aplicada neste trabalho dedicou-se ao desenvolvimento do aplicativo para o rastreamento de diabetes do tipo 2 por método não invasivo com interface gráfica utilizando diretrizes de boas práticas, recurso de armazenagem off-line e comunicação ao servidor por meio da API RESTful.

A base de dados com os dados não invasivos dos pacientes vai ser enviado a um servidor que executará técnicas de *Computer Aided Diagnosis* do inglês (CAD), do estudo proposto por (RIBEIRO, BARROS, *et al.*, 2015) e devolverá uma nota para o *tablet* classificando a pessoa como diabética ou não diabética.

Para a implementação do projeto, as seguintes etapas foram feitas:

1. Estudo da bibliografia envolvida;
2. Estudo da documentação do Android;
3. Estudo da documentação do Material Design;
4. Estudo da documentação do Firebase;
5. Experimentos práticos.

2.1 Diagrama de Casos de Uso

Toda a modelagem do sistema está exposta na Figura 2.

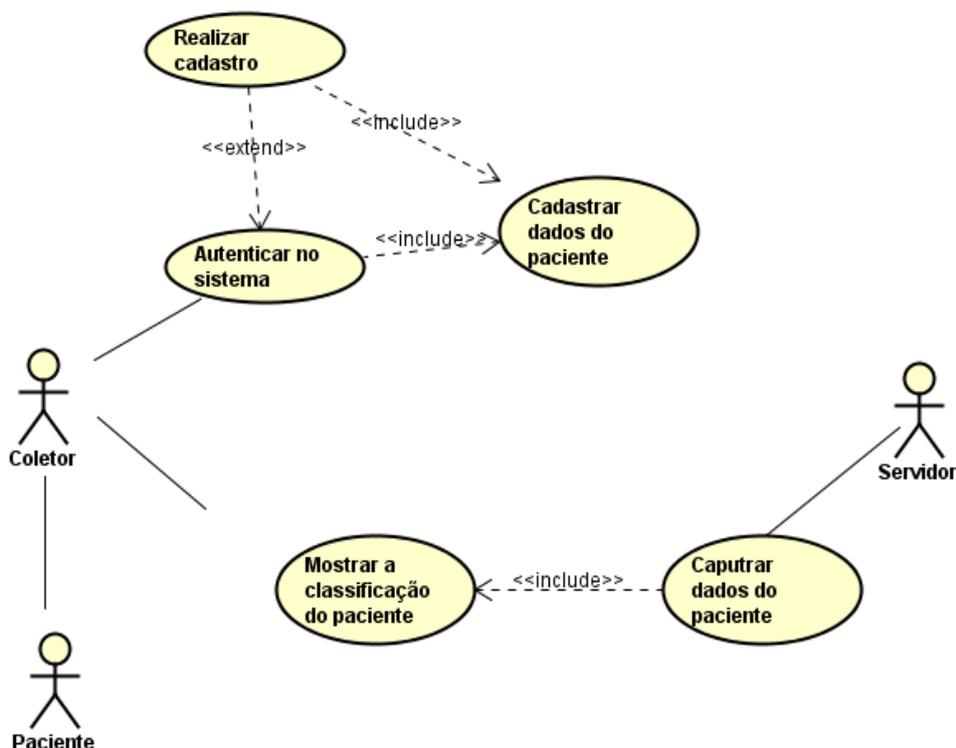
- **Atores**

1. **Coletor:** É o profissional de saúde que irá cadastrar os dados não invasivos do paciente e mostra-lo a sua classificação da diabetes.
2. **Paciente:** É a pessoa que quer saber a sua situação em relação a diabetes. Só deverá acessar o sistema por meio do coletor.
3. **Servidor:** é um software que fica em outro computador onde captura as informações do questionário no aplicativo, armazena no banco de dados, processa o resultado da classificação do paciente e envia a resposta para o aplicativo.

- **Casos de Uso**

1. **Autenticar no sistema:** Antes de utilizar o sistema, o coletor deve se identificar primeiro com e-mail e senha.
2. **Realizar Cadastro:** Caso não tenha autorização, o coletor pode realizar o cadastro no sistema validando seu e-mail e senha.
3. **Cadastrar dados do paciente:** Após a autenticação, será exibido o formulário o qual o coletor deve preencher junto com o paciente. Depois de confirmar os dados, o coletor irá enviá-los para o servidor e esperar a sua resposta com a classificação do paciente.
4. **Capturar dados do paciente:** O servidor irá receber os dados do paciente no formato JSON e em seguida processá-los. Os dados dos pacientes serão armazenados em um registro organizados por tabelas, contendo todas as suas informações, data de criação no banco, data de atualização dos dados no banco.
5. **Mostrar a classificação do paciente:** O servidor irá solicitar a API que usará as técnicas de Análise de Componentes Independentes (ICA) e Máquina de Vetor de Suporte para determinar a situação do paciente. Após a resposta, o servidor mandará a informação para o aplicativo e o coletor irá comunicar a situação.

Figura 2: Diagrama de Caso de Uso do Sistema



Fonte: Criado pelo software Astah UML

2.2 Cenário

- **Autenticar no sistema**

Ator: Coletor

Pré-condição: Possuir e-mail e senha cadastrados no sistema

Fluxo normal:

1. Digitar e-mail correto.
2. Digitar senha corretar.
3. Pressionar botão "Enviar"

Fluxos alternativos:

1. Ao errar e-mail, digitar novamente.
2. Ao errar senha, digitar novamente.
3. Caso o coletor não lembre da senha, deve solicitar o contato do administrador.

Pós-condição: Acessar a página de formulário

- **Realizar Cadastro**

Ator: Coletor

Pré-condição: Não tiver conta no sistema

Fluxo normal:

1. Preencher o formulário corretamente.
2. Confirmar dados e apertar o botão “Cadastrar”

Fluxos alternativos:

1. Corrigir os dados inválidos
2. Pós-condição: Possuir conta no sistema

Pós-condição: Coletor poderá fazer autenticação com e-mail e senha

- **Cadastrar dados do paciente:**

Ator: Coletor

Pré-condição: O profissional já está autenticado no sistema

Fluxo normal:

1. Preencher o formulário junto com o paciente.
2. Verificar o que foi digitado, caso positivo, caso de uso “Enviar para o servidor”

Fluxos alternativos:

1. Caso tenha dados inválidos, digitar corretamente.
2. Caso não tenha internet, esperar a conexão se estabelecer para enviar os dados novamente, não terá perda das informações pois estará armazenado no aplicativo.

Pós-condição: Dados verificados e pronto para serem processados no servidor.

- **Capturar dados do paciente:**

Ator: Servidor

Pré-condição: Os dados enviados pelo aplicativo devem estar consistentes

Fluxo normal:

1. Receber os dados via JSON
2. Formatar esses dados e inserir no banco de dados.

Fluxos alternativos:

1. Caso tenha erro ao conectar ou inserir no banco de dados, retornar essa informação ao aplicativo para o coletor tentar mais tarde.

Pós-condição: Os dados prontos para serem minerados pela API.

- **Mostrar a classificação do paciente:**

Ator: Servidor e Coletor

Pré-condição: Dados recebidos pelo servidor com sucesso

Fluxo normal:

1. Enviar os dados para a API
2. Receber a resposta da situação do paciente e mostrar na tela do aplicativo

Fluxos alternativos:

1. Caso tenha erros, mostrar a mensagem de erro no aplicativo.

Pós-condição: O coletor ter informação da situação do paciente e repassar a informação para ele.

2.3 Projeto

Esta seção apresentará a descrição dos componentes do trabalho: Descrição das bases de dados, a diferença entre diagnosticar e rastrear uma doença e as telas do aplicativo.

2.3.1 Diagnóstico e Rastreamento de Diabetes

Muito confundem o significado de diagnóstico e rastreamento. Diagnóstico é voltada para identificação de uma determinada doença por meio de uma consulta médica. Em um agrupamento de dados, formado de sinais e sintomas, histórico clínico, exame físico e exames complementares é analisado pelo profissional de saúde e sintetizado em uma ou mais doenças.

No rastreamento, quer antecipar o acontecimento da doença não previamente conhecida, pela utilização de testes, exames e outros meios complementares de diagnóstico, os quais podem ser rapidamente aplicados. Estes testes não têm por objetivo ser um diagnóstico e sim identificar suspeitos de uma determinada patologia ou outra condição (WILSON e JUNGNER, 1968).

A OMS, em 1968, divulgou as diretrizes de rastreio, mesmo que seja de muitos anos atrás, continuam sendo aplicadas até hoje. São elas:

1. A condição precisa ser um importante problema de saúde.
2. A condição deve possuir tratamento.
3. Devem haver facilidades de diagnóstico e tratamento.
4. A doença precisa ter um estágio latente.
5. Deve haver um exame diagnóstico para a condição.
6. Este exame deve ser aceitável pelo paciente.
7. A história natural da doença deve ser adequadamente compreendida.
8. Deve haver uma política de tratamento definida.
9. O custo de encontrar os casos deve ser economicamente favorável em relação ao custo do tratamento tardio.
10. A busca de novos casos deve ser contínua, e não um processo "de uma vez por todas".

Pela razão da DM2 ter uma fase assintomática é possível rastreá-la. Com o rastreio previamente antecipado da DM2, seu tratamento pode trazer inúmeros privilégios na qualidade de vida do paciente.

2.3.2 Telas do Aplicativo

Foi utilizado o recurso do "Text Field" do Material Design para o usuário preencher os campos do e-mail e Senha para autenticar no aplicativo. Esse componente já valida o que foi digitado e emite um erro caso seja inserido dados inválidos.

O botão "Enviar" verifica o e-mail e senha do usuário, caso sejam válidos é autorizado o uso do aplicativo. Já o botão "Cadastrar" cria um novo usuário no sistema. A Tela de Autenticação está na Figura 3.

Figura 3: Tela Inicial



Fonte: criado utilizando o software Balsamiq Studio

Na tela de Cadastro, os Text Fields foram empregados com os campos “Nome”, “CPF”, “E-mail”, “Senha” e “Confirmar Senha” para inserir novos usuários de coletores no sistema. Antes de realizar o processo de cadastro, é feita toda a validação dos itens preenchidos e é emitido um erro no campo que foi digitado errado.

O botão “Confirmar” adiciona o novo usuário e o botão “Limpar” apaga o que está escrito em todos os campos.

A Figura 4 apresenta a Tela de Cadastro.

Figura 4: Tela de Cadastro

A smartphone mockup displaying a registration form titled "Cadastro de Coletor". The form has a light blue header with the title. Below the header, there are five input fields: "Digite seu nome", "Digite seu CPF", "Digite seu e-mail", "Digite sua senha", and "Confirme sua senha". Each field has a corresponding label above it. The "senha" fields have a small eye icon to the right of the input field. At the bottom of the form, there are two buttons: "CONFIRMAR" and "LIMPAR". The status bar at the top shows the time 17:08, signal strength, Wi-Fi, and battery icons.

Fonte: criado utilizando o software Balsamiq Studio

Nessa primeira parte da tela de formulário, são inseridos os dados do paciente: Idade, Sexo, PAS (mmHg), PAD (mmHg), Cintura (cm), Peso (kg), Altura (cm), Antecedentes Familiares, Tabagismo, Sedentarismo, Sobrepeso, Infarto, CPF e Nome do Paciente sendo os campos compostos por Text Field para inserção dos dados e Radio Buttons para escolha de opção.

Antes dos dados irem para o servidor, é feita toda a validação sempre alertando ao usuário em qual campo o erro aconteceu. A Figura 5 mostra uma parte do formulário a ser digitado pelo coletor.

Figura 5: Formulário do Coletor



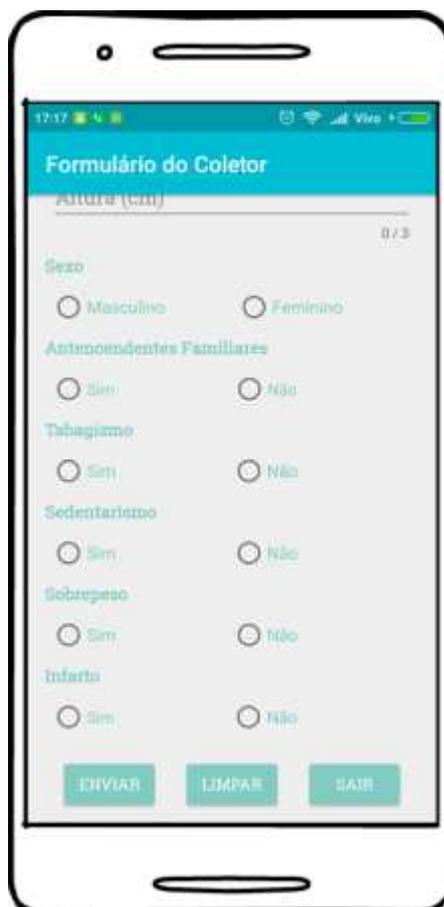
The image shows a smartphone screen displaying a form titled "Formulário do Coletor". The form contains several input fields, each with a character count indicator on the right side. The fields are: "Nome do Paciente" (0 / 50), "CPF do Paciente" (0 / 8), "Idade" (0 / 3), "PAS (mmHg)" (0 / 3), "PAD (mmHg)" (0 / 3), "Cintura (cm)" (0 / 3), and "Peso (kg)". The form is displayed on a white smartphone with a black border.

Fonte: criado utilizando o software Balsamiq Studio

Nessa segunda parte da tela, exposto pela Figura 6, é mostrado o restante do formulário incluindo os botões “Enviar”, para encaminhar os dados para o servidor, o botão “Limpar”, para excluir os dados digitados e o botão “Sair”, para encerrar a sessão do usuário e retornar a tela de Login.

Em cada campo do formulário, há um contador de caracteres que informa a quantidade máxima que deve ser exibida e que verifica se a informação está dentro de um intervalo válido, deixando os dados mais consistentes.

Figura 6: Tela Inicial com botões



The image shows a smartphone screen displaying a form titled "Formulário do Coletor". The form is for data collection and includes several sections with radio button options:

- Sexo**: Masculino, Feminino
- Antecedentes Familiares**: Sim, Não
- Tabagismo**: Sim, Não
- Sedentarismo**: Sim, Não
- Sobrepeso**: Sim, Não
- Infarto**: Sim, Não

At the bottom of the form, there are three buttons: "ENVIAR", "LIMPAR", and "SAIR". The top of the screen shows the time 17:17 and various status icons.

Fonte: criado utilizando o software Balsamiq Studio

A Figura 7 mostra quando é disparado o erro caso o usuário digite uma informação errada, para certificar que todos os dados que serão enviados para o servidor sejam válidos.

Fazendo a validação do lado *front-end* implicará do aumento de desempenho do servidor, pois ele não irá gastar recursos de processamento e memória para verificar se os dados estão corretos, o próprio aplicativo fará essa tarefa.

Figura 7: Corrigindo erros

The image shows a smartphone screen displaying a form titled "Formulário do Coletor". The form contains the following fields and values:

- Nome do Paciente: Daniel (6 / 50)
- CNPJ do Paciente: 56775794 (8 / 8)
- Idade: (0 / 3) with a red error message: "Deve estar entre os valores 18 e 65"
- PAO (mmHg): 40 (1 / 3)
- PAO (mmHg): 80 (2 / 3)
- Colesterol (mg/dl): 58 (2 / 3)
- Press (kg): 90 (2 / 3)

Fonte: criado utilizando o software Balsamiq Studio

A tela da Figura 8 exibe ao usuário todos os dados que ele preencheu no formulário, para ele pode verificar se está tudo com as informações do paciente. É uma forma de evitar dos dados errados irem para o servidor.

O botão "Editar" volta para a tela de formulário para o usuário corrigir as informações e o botão "Confirmar" envia os dados para o servidor e é exibido novamente o formulário outra vez para o próximo paciente.

Figura 8: Tela de Confirmação de Dados



Fonte: criado utilizando o software Balsamiq Studio

Em todas as telas, o conceito de cores primárias e secundárias do Material Design foram aplicados, a fonte utilizada é Roboto de acordo com as métricas de design.

Figura 9: Tela com a resposta do servidor



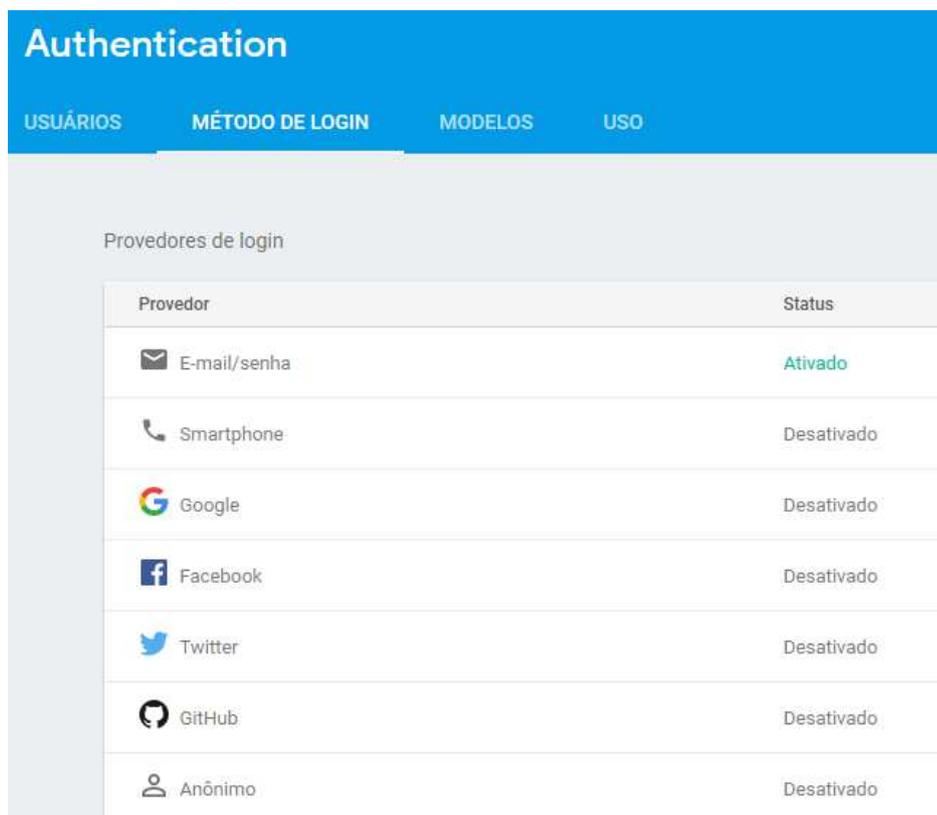
Fonte: criado utilizando o software Balsamiq Studio

Nessa tela da Figura 9, o AlertDialog é gerado com a resposta do servidor, classificando o paciente de acordo com os dados passados para ele. Ao apertar "OK", automaticamente é redirecionado ao formulário para obter novas informações de pacientes.

2.3.3 Utilização do Firebase

O recurso de Autenticação utilizada é apenas para informar o e-mail e a senha do usuário sem precisar associar a uma rede social. As maneiras de autenticação estão listadas na Figura 10.

Figura 10: Tela de Autenticação do Firebase



The screenshot shows the 'Authentication' section of the Firebase console. It features a blue header with the title 'Authentication' and four navigation tabs: 'USUÁRIOS', 'MÉTODO DE LOGIN', 'MODELOS', and 'USO'. Below the tabs, the 'MÉTODO DE LOGIN' tab is active, displaying a table titled 'Provedores de login'. The table has two columns: 'Provedor' and 'Status'. The providers listed are E-mail/senha (Ativado), Smartphone (Desativado), Google (Desativado), Facebook (Desativado), Twitter (Desativado), GitHub (Desativado), and Anônimo (Desativado).

Provedor	Status
 E-mail/senha	Ativado
 Smartphone	Desativado
 Google	Desativado
 Facebook	Desativado
 Twitter	Desativado
 GitHub	Desativado
 Anônimo	Desativado

Fonte: (Firebase, 2014)

O armazenamento off-line já está nativo, ao adicionar o SDK do Firebase na aplicação, quando estiver queda de conectividade, os dados dos pacientes serão mantidos no aplicativo e quando há conexão com a internet, automaticamente as informações serão adicionadas ao servidor. Para demonstrar esse recurso, aplicou-se o recurso do Banco de Dados em Tempo Real do Firebase.

Caso não há conexão com a internet, eles são mantidos dentro do disco. Quando a conectividade é restabelecida, o dispositivo recebe as alterações perdidas e faz a sincronização com o estado atual do servidor.

Para o armazenamento de dados, o servidor assumirá esse papel operando com o Banco de Dados MYSQL. O Banco de Dados do Firebase é apenas um experimento para verificar a função do armazenamento off-line.

2.3.4 Comunicação do Cliente-Servidor

Para enviar dados para o servidor, a requisição POST será feita, porque é projetado para fazer a solicitação ao servidor web que aceita os dados anexados

dentro da mensagem de requisição. Para fazer o teste do *web service* foi utilizado o software Postman, o qual permite realizar requisições HTTP por meio de uma interface simples e intuitiva, auxiliando o teste e depuração de serviços REST.

O formato de dados utilizado é o JSON, porque é a representação que muitas plataformas feitas de diferentes linguagens de programação são entendidas e a comunicação entre eles é bem mais fácil. A Figura 11 mostra o formato dos dados que será enviado pelo servidor após o coletor preencher seu formulário.

Figura 11: Enviando dados de teste para o servidor no formato JSON

```
{
  "nome": "Daniel",
  "cpf": "117213123-12",
  "idade": "57",
  "sexo": "M",
  "pas": "13",
  "pad": "10",
  "cintura": "12",
  "peso": "60",
  "altura": "1.57",
  "antecedentesfamiliares": "1",
  "tabagismo": "0",
  "sedentarismo": "1",
  "sobrepeso": "1",
  "infarto": "0"
}
```

Fonte: Criado pelo software Postman

3. FUNDAMENTAÇÃO TEÓRICA

Este Capítulo descreverá todo o referencial teórico envolvido no trabalho que são a Engenharia de Requisitos, Análise de Componentes Independentes, Máquina de Vetor de Suporte, Material Design, Firebase e REST.

3.1 Engenharia de Requisitos

O autor (LAMSWEERDE, 2000) define ER (Engenharia de Requisitos) como a identificação dos objetivos a serem visados pelo futuro sistema, o procedimento dos objetivos em serviços e restrições, e a atribuição de responsabilidades pelos requisitos resultantes a agentes humanos, dispositivos e software.

A autora (ZAVE, 1997) define ER como:

[...] é o ramo da engenharia de software que se preocupa com os objetivos do mundo real para as funções e restrições aplicáveis a sistemas de software. E também tem o cuidado com o relacionamento destes fatores para especificações precisas do comportamento do software e com sua evolução no tempo e através de famílias de produtos.

Os autores (SOMMERVILLE e KOTONYA, 1998) definiu Engenharia de Requisitos como sendo a maneira de escolher o nome das atividades desenvolvidas, no contexto do ciclo de vida de software, que estão relacionadas com a definição dos requisitos de um sistema.

A ER identifica metas a serem atingidas pelo sistema que será desenvolvido, tal como o procedimento de tais metas em serviços e restrições. O estudo também tem interesse no relacionamento desses fatores para gerar uma especificação do comportamento do software e sua evolução durante o tempo (ZAVE, 1997).

A ER é um ramo enorme e multidisciplinar, seus aspectos sociais e humanos desempenham um importante papel (ZAVE, 1997) e (NUSEIBEH e EASTERBROOK, 2000). A ER foi gerada para determinar todas as atividades envolvidas em descobrir, documentar e manter um conjunto de requisitos para um projeto de sistema de software (SOMMERVILLE e SAWYER, 1997).

Para determinar o ciclo de vida do projeto, aplicou-se as orientações da ER para melhor controle da construção do software. Todos os requisitos foram bem entendidos para não haver erro do desenvolvimento do aplicativo com funções erradas.

3.1.1 Requisitos Funcionais e Não Funcionais

Levantamento de Requisitos é uma etapa do desenvolvimento de sistema dentro da Engenharia de Requisitos. É conversar e entender as necessidades do cliente e conhecer a sua regra de negócio, assim projetar a aplicação que o satisfaça.

A ER determina o processo de definição de requisitos como um processo onde, o que tem a obrigação de elicitar, modelar e analisar. Esse processo lida com diferentes pontos de vista, e usar uma variedade de métodos, ferramentas e pessoal. O resultado desse processo é um modelo, que a partir disso, um documento de requisitos é gerado. Esse processo ocorre em um contexto anteriormente estabelecido que chamamos de Universo de Informação (KAINDL, BRINKKEMPER, *et al.*, 2002).

Os Requisitos são projetados durante as fases iniciais do desenvolvimento do sistema como uma especificação do que deveria ser realizado. São definições de como o sistema deveria se comportar (SOMMERVILLE e SAWYER, 1997).

Há muitas denominações presentes na literatura para o termo requisitos. Uma definição simples dada por (MACAULAY, 1996) que o requisito é apenas algo que o cliente necessita. De acordo com (JACKSON, 1995) requisitos são fenômenos ou propriedades do domínio da aplicação o qual devem exercer, normalmente expressos em linguagem natural, diagrama informal ou outra notação apropriada ao entendimento do cliente e da equipe de desenvolvimento.

A complexidade de um software é determinada por sua funcionalidade (o que o sistema faz) denominada de Requisitos Funcionais (RF) e por exigências gerais que fazem parte do desenvolvimento do software como custo, performance, segurança, confiabilidade, manutenibilidade, portabilidade, custos operacionais entre outros, denominados Requisitos Não-Funcionais (RNF). (CHUNG, NIXON, *et al.*, 1999)

Os requisitos obtidos para construção da aplicação foram adquiridos pela Orientadora, os mesmos são tratados como os objetivos específicos desse trabalho.

3.1.2 Stakeholders

A definição mais utilizada é dada por (FREEMAN, 1984): “Um *stakeholder* em alguma organização é qualquer grupo ou indivíduo que pode influenciar ou ser influenciado pela obtenção dos objetivos da organização”.

O escritor (KRUCHTEN, 2000) define que é qualquer pessoa ou representante de uma organização que possua um *stake* – um grande interesse – na solução de um projeto.

Os *Stakeholders* do projeto são a orientadora e o co-orientador, pois são as pessoas que fizeram a pesquisa sobre o classificador de diabetes e sabem como deve ser a interação desse sistema.

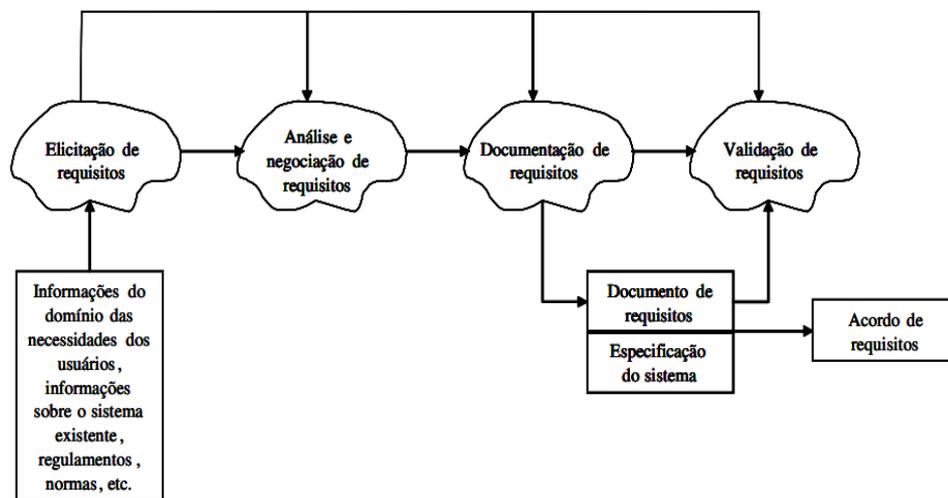
3.1.3 Processo da Engenharia de Requisitos

A engenharia de software é formada através de um conjunto de fases. Cada uma delas tem a possibilidade de envolver métodos, ferramentas e procedimentos, cujos tipos de formação são citados como modelo de engenharia de software

(PRESSMAN, 2002). Ainda segundo o autor, não importando a modelagem de desenvolvimento de software, o processo sempre inclui três fases: definição, desenvolvimento e manutenção.

Nas últimas décadas, diversos métodos de análise e especificação de requisitos foi criada, onde poucas propostas tem o objetivo da sistematização da definição de requisitos de forma menos subjetiva (SANTANDER e CASTRO, 2002). Genericamente, a maior parte dos processos de engenharia de requisitos podem ser definidos em um modelo de atividades de alta granularidade, como apresentado na Figura 12.

Figura 12: Modelo de processo de engenharia de requisitos



Fonte: adaptado de (SOMMERVILLE e KOTONYA, 1998)

As etapas apresentadas no modelo são descritas a seguir (SOMMERVILLE e KOTONYA, 1998):

- **Elicitação de requisitos de software:** onde são detectadas as expectativas e necessidades dos *stakeholders* a respeito do software a ser desenvolvido.
- **Análise e negociação dos requisitos:** logo após de obter os requisitos iniciais, estes serão usados como base para análise de requisitos. Ela separa os requisitos em categorias, percorre as relações entre eles, e classifica a relevância de cada um de acordo com a necessidade do *stakeholders*. Os requisitos são negociados para definir quais serão aprovados.
- **Documentação dos requisitos de software:** os requisitos são documentados de grande nível de detalhe. Geralmente, é construído um documento de especificação de requisitos, o qual todos os *stakeholders* possam entendê-lo.

- **Validação dos requisitos de software:** esta última etapa analisa a especificação de software, de forma a garantir que todos os requisitos foram definidos sem incerteza, inconsistências ou omissão, e que todos os problemas foram detectados e corrigidos.

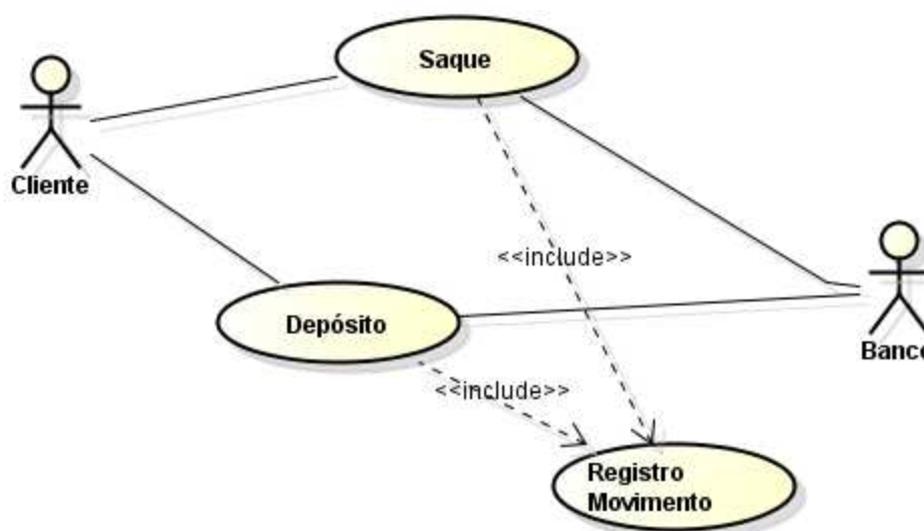
As etapas acima foram realizadas no projeto, desde do protótipo como a criação dos diagramas de casos de uso e suas especificações por cenários. A implementação atendeu o que os *stakeholders* desejavam no sistema.

3.1.4 Casos de Uso

Um Caso de Uso foi inicialmente apresentado em 1987 como um instrumento que acompanhou a técnica *Objectory*, e a aproximação de Casos de Uso para a Engenharia de Software foi criada por (RUMBAUGH, JACOBSON e BOOCH, 1999).

O diagrama de casos de uso demonstra o relacionamento entre atores e casos de uso em um sistema. Nele, os usuários, banco de dados, equipamentos, ou até outros sistemas que possam interagir com o sistema em desenvolvimento, são representados por atores, os casos de uso (ações) são representados por elipses; e as interações entre atores e casos de uso são representadas por uma linha contínua (SOMMERVILLE, 2003). A **Erro! Fonte de referência não encontrada.** apresenta um exemplo de sua modelagem.

Figura 13: Exemplo de Caso de Uso



Fonte: (GUEDES, 2006)

3.1.5 Cenário

Segundo (RUMBAUGH, JACOBSON e BOOCH, 1999), o um cenário é um conjunto ordenado de interação normalmente entre um sistema e um conjunto de atores externos ao sistema. Pode consistir numa sequência concreta de passos de interação ou num possível conjunto de passos de interação.

Um ator é uma função de objeto ou objetos fora de um sistema que comunica com ele e retratam todas as possíveis interações com o sistema (MOISIADIS, 2000)

Outro conceito é dado por (ROSSEN e CARROLL, 2001), um cenário é uma descrição que contém atores, a informação por trás deles, os seus objetivos e sequências de ações e eventos. Pode incluir também os obstáculos e êxitos dos atores. Em determinados sistemas, os cenários podem esconder um dos elementos ou expressá-lo de forma simples ou implícita.

Os cenários podem ser expressos em várias formas e formatos. Por exemplo, narrativas textuais, *storyboards*, protótipos produzidos em várias maneiras. No trabalho, o cenário é descrito por situações. Quando um caso de uso for executado com sucesso, vai para um fluxo, caso contrário, vai para o outro. Além de informar o que os pré-requisitos e o que se espera na realização do caso de uso.

De acordo com (MAIDEN, MINOCHA, *et al.*, 1998), a diferença entre “casos de uso” e “cenários” é que os casos de uso emprega-se uma coleção de ações e regras temporais que comandam como a ação pode ser unida, enquanto, um cenário é uma sequência de eventos ordenados, o qual são determinados o início e o fim dos eventos ou ações de um caso de uso. Contudo, é praticável ter vários cenários para um caso de uso, cada um estabelecido um possível curso de ações através de casos de uso.

3.2 Análise de Componentes Independentes

A técnica *Independent Component Analysis* (ICA) é uma projeção de subespaço largamente empregada, que prepara dados de um espaço de alta dimensão para um espaço de menor dimensão (HYVÄRINEN, KARHUNEN e OJA, 2004).

No atual estudo, as bases de ICA foram calculadas aplicadas por meio do algoritmo FastICA (HYVÄRINEN, KARHUNEN e OJA, 2004), enquanto que outros métodos, como por exemplo, Informax (BARTLETT, MOVELLAN e SEJNOWSKI, 2002), a probabilidade máxima (HYVÄRINEN, 1999) e Jade (RIBEIRO, BARROS, *et al.*, 2015) podem ser usados.

O algoritmo FastICA é o mais comumente utilizados dentro dos algoritmos ICA, criado por Hyvärinen, Karhunen e Oja (HYVÄRINEN, KARHUNEN e OJA, 2004).

Constitui na otimização de uma função de contraste que mede a não Gaussianidade da mistura. A fama do FastICA pode ser referente à sua simplicidade, facilidade de interpretação e flexibilidade para escolher a função de não-linearidade (WEI, 2017).

Há duas versões de algoritmos FastICA: FastICA de uma unidade e FastICA simétrica. A versão unitária retira uma fonte de cada vez até que todas as fontes sejam restauradas. Para impossibilitar que o algoritmo faça a convergência para a mesma fonte duas vezes, um procedimento complementar chamado *deflation* é necessário (DELFOSSE e LOUBATON, 1995). O FastICA de uma unidade tem uma inconveniência comum de todos os esquemas de separação sequencial de fontes: o aumento de erros ao longo da extração contínua de problemas de grande dimensionalidade.

A versão simétrica do FastICA (OJA e ZHIJIAN, 2006) extrai todos os sinais de origem simultaneamente. Nesse trabalho foi aplicado o FastICA simétrica porque a mesma não tem a consequência da desvantagem de aumento dos erros e normalmente é mais estável.

3.3 Máquinas de Vetor de Suporte

Um *support vector machine* (SVM) é uma reprodução matemática de um modelo de classificação tradicionalmente binária. Os problemas são descritos em um espaço e esse classificador calcula uma fronteira (hiper-plano) que divide ambas as classes de forma distribuída. Para identificar novos casos, basta averiguar em que lado da fronteira este é demonstrado. Ainda que a princípio só é praticável uma classificação por meio de uma função linear, ou seja, a fronteira entre classes ser uma linha reta, aperfeiçoamentos mais atualizados tornaram possível a classificação não linear através de funções polinomiais e de base radial (BORGES, 2016)

O grande benefício deste método consiste na capacidade de atingir bons resultados de classificação na maioria dos casos de problemas de classes binárias: os testes feitos (MEYER, LEISCH e HORNIK, 2003), a SVM descobriu os melhores de 19 em 21 testes. Entretanto, não se pode afirmar que este é o melhor meio de classificação, pois depende da forma de estruturação dos conjuntos de dados.

Essa técnica tem grande dependência da função escolhida e do número de classes. De modo que as SVM são classificadores binários, as situações que rodeiam classes ternárias ou superiores são minimizadas a problemas de classes binárias referidos por associação de classificadores. Quando verificados com outros métodos de classificação, a SVM tem um treino lento e requer um considerável recurso computacional (MEYER, LEISCH e HORNIK, 2003).

3.4 Descrição das bases de dados estudadas

Segundo (NGUYEN, 2010), uma base de dados é um conjunto de informações relacionadas entre si, de algum tema definido. Ainda segundo o autor, ela tem a função de gerenciar muitas coleções de informações de forma a simplificar a organização, manutenção e pesquisa de dados.

Com a base de dados em mãos, procura-se padrões buscando descobrir informações consideráveis e retirando-as com o auxílio de computadores. No contexto deste trabalho, buscando por parâmetros que possam auxiliar no diagnóstico da diabetes tipo 2.

No estudo feito por (RIBEIRO, BARROS, *et al.*, 2015) foram usadas três bases de dados, que são pilares centrais aplicado nesse trabalho, são elas: uma base de dados brasileira, uma base de dados PIMA e uma base de dados Africana-Americana.

A primeira base de dados foi disponibilizada pelo DATASUS (Departamento de informática do Sistema Único de Saúde). Ela possui 17 marcadores, mas foram utilizados somente 14, pois de acordo com o estudo feito por (RIBEIRO, BARROS, *et al.*, 2015), são não invasivos que estão na Tabela 1:

Tabela 1: Marcadores do DATASUS

Nº	Marcador	Continua	Discreta
1	Idades (anos)	X	
2	Pressão Sanguínea Sistólica	X	
3	Pressão Sanguínea Diastólica	X	
4	Cintura (cm)	X	
5	Peso (kg)	X	
6	Altura (cm)	X	
7	Antecedentes familiares		X
8	Uso de Tabaco		X
9	Estilo de vida Sedentário		X
10	Sobrepeso		X
11	Infarto		X
12	Outras Coronariopatias		X
13	AVC		X
14	Pé Diabético		X

Fonte: adaptado de (RIBEIRO, BARROS, *et al.*, 2015)

Aonde, os itens de 1 ao 6 são variáveis contínuas e os itens de 7 ao 14 são variáveis discretas. Esses indicadores clínicos foram recebidos de pacientes entre 12 e 100 anos, onde 500 foram classificados como diabéticos e 500 classificados como não diabéticos, 50% para cada caso.

A segunda base de dados (PIMA) é bastante empregada na literatura para testes de sistemas CAD (*Computer Aided Design*) (UCI Machine Learning Repository, 2007) já que é uma base de dados de acesso livre, sua desvantagem é ser desbalanceada e homogênea.

O desbalanceamento ocorre em um conjunto de dados, se existir uma clara desproporção entre a quantidade de exemplos de uma ou mais classes em relação às outras demais, podendo valorizar as classes predominantes de modo que as de menor representação possam ser ignoradas (BARELLA, 2015).

Em relação à homogeneidade, ela é representada por apenas uma população homogênea (índios PIMA), mais exatamente por mulheres com idade mínima de 21 anos. Como a população brasileira é composta pela mistura de raças tornando reconhecimento de padrões nessa base de dados impraticável para nossa população.

A última base, a Africana-Americana, foi disponibilizada da Faculdade de Medicina da Universidade de Virginia. Possui 19 marcadores que foram conseguidos de 403 pacientes de um estudo feito para entender a diabetes e outros fatores de risco cardiovasculares na Virginia central para afro-americanos. Logo 60 pacientes (14,88%) foram classificados como diabéticos e 343 (85,11%) como não diabéticos.

3.5 Material Design

É uma linguagem de *design* criada pela Google, com o objetivo de unificar o visual do Android com os aplicativos do Chrome OS, foi inspirado em materiais e objetos do mundo real e agem conforme são usados.

Os tópicos a seguir descreve as métricas de design que deve estar no projeto de acordo com (Material Design, 2014).

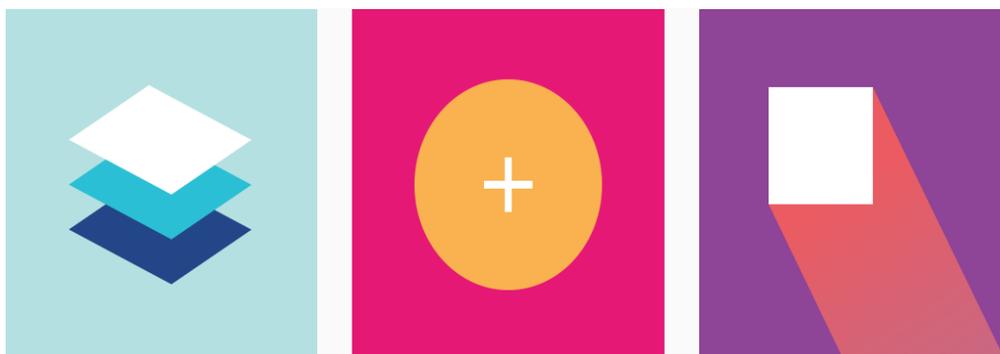
3.5.1 Os três princípios

O Material Design trabalha com três princípios representados na Figura 14:

- **O Material é a Metáfora:** é a teoria que junta o espaço racional e um sistema de movimento. O material é inspirado na realidade tátil, baseado no estudo de papel e tinta, mas tecnologicamente avançado e aberto à imaginação e magia. As superfícies e bordas são provas visuais que tudo é baseado na realidade.

- **Negrito, gráfico, intencional:** os elementos fundamentais do projeto em impressão são: tipografia, grades, espaço, escala, cor e uso de imagens-guia. Eles fazem muito mais do que agradar aos olhos. Estes elementos criam hierarquia, significado e foco. Opções de cores, aparência de borda a borda, tipografia em grande escala, e espaço em branco intencional, criam uma interface gráfica ousada.
- **O Movimento fornece significado:** o movimento respeita e reforça o usuário como o motor principal. Ações primárias do usuário são os pontos que iniciam o movimento. Toda a ação ocorre em um único ambiente, os objetos são apresentados ao usuário sem quebrar a continuidade da experiência, mesmo quando eles se transformam e se reorganizam. O movimento é significativo e apropriado, que serve para concentrar a atenção e manter a continuidade. O *feedback* é sutil, mas claro. As transições são eficientes e claras.

Figura 14: Os três princípios do Material Design



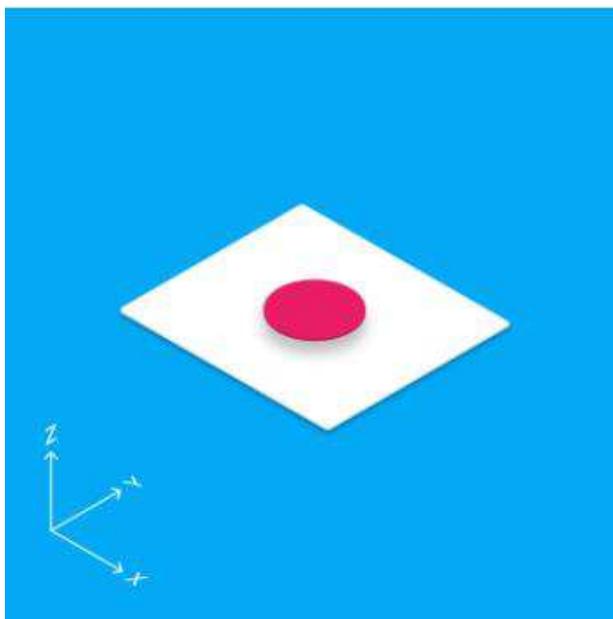
Fonte: (Material Design, 2014)

3.5.2 O ambiente do Material Design

O ambiente do Material Design é um espaço 3D, o que significa que todos os objetos possuem dimensões x, y e z. O eixo z está perpendicularmente alinhado ao plano da tela, com o eixo z positivo se estendendo em direção ao visualizador.

Cada folha de material ocupa uma única posição ao longo do eixo z e tem uma espessura padrão de 1dp, equivalente a um pixel de espessura em telas com uma densidade de pixels de 160. A Figura 15 demonstra um exemplo visual.

Figura 15: Espaço 3D com eixos x, y e z.



Fonte: (Material Design, 2014)

3.5.3 Elevação e Sombra

No mundo físico, os objetos podem ser empilhados ou afixados um ao outro, mas não podem passar um pelo outro. Objetos também lançam sombras e refletem a luz.

O Material Design reflete essas qualidades para formar um modelo espacial familiar aos usuários e pode ser aplicado de forma consistente em aplicativos.

A elevação é a profundidade relativa, ou distância, entre duas superfícies ao longo do eixo z. A ideia de elevação é mostrada na Figura 16.

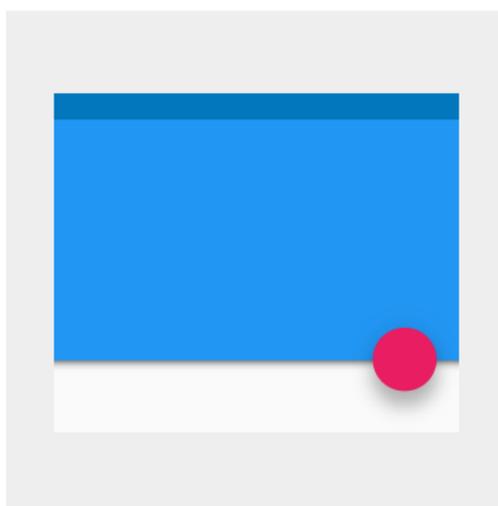
Figura 16: Elevação do Material Design.



Fonte: (Material Design, 2014)

As sombras fornecem pistas visuais importantes sobre a profundidade dos objetos e o movimento direcional. Eles são a única sugestão visual indicando a quantidade de separação entre as superfícies. A elevação de um objeto determina a aparência de sua sombra. A Figura 17 exibe esse conceito.

Figura 17: Sombra no Material Design.



Fonte: (Material Design, 2014)

3.5.4 Sistema de cores do Material Design

Uma cor primária é a cor exibida com mais frequência nas telas e componentes do seu aplicativo. Também pode ser usado para acentuar elementos, se você não tiver uma cor secundária.

Para criar contraste entre elementos, você pode usar tons mais claros ou mais escuros da sua cor primária. O contraste entre tons mais claros e mais escuros ajuda a mostrar a divisão entre as superfícies, como entre a barra de status e a barra de ferramentas.

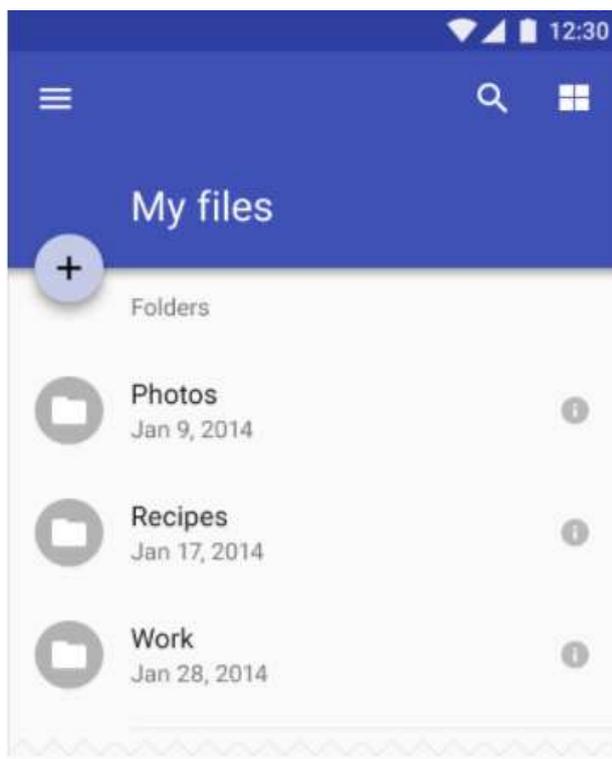
Uma cor secundária é usada para acentuar partes selecionadas de sua UI. Pode ser complementar ou análogo à sua cor primária, mas não deve ser simplesmente uma variação clara ou escura de sua cor primária. Ele deve contrastar com os elementos que o cercam e ser aplicado com moderação como um sotaque.

As cores secundárias são mais utilizadas para:

- Botões, botões de ação flutuante e texto do botão
- Campos de texto, cursores e seleção de texto
- Barras de progresso
- Controles de seleção, botões e controles deslizantes
- Links
- Manchetes

Usar uma cor secundária é opcional. Não é necessário se você usar variações de sua cor primária para acentuar elementos. A Figura 18 mostra um exemplo da utilização de cores em uma aplicação.

Figura 18: Exemplo do uso de cores.

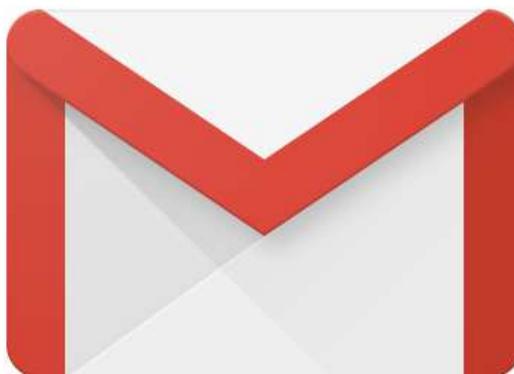


Fonte: (Material Design, 2014)

3.5.5 Ícones

Os ícones de produtos são a expressão visual dos produtos, serviços e ferramentas de uma marca. Simples, arrojados e amigáveis, eles comunicam a ideia central e a intenção de um produto. Enquanto cada ícone de produto é visualmente distinto, todos os ícones de produtos para uma determinada marca devem ser unificados através de conceito e execução. Na Figura 19 aponta o ícone de produto do Gmail que é o serviço de e-mail do Google.

Figura 19: Exemplo de Ícone de Produto



Fonte: (Material Design, 2014)

Um ícone do sistema, ou ícone de UI, simboliza um comando, arquivo, dispositivo ou diretório. Os ícones do sistema também são usados para representar ações comuns, como lixo, impressão e salvar.

O design dos ícones do sistema é simples, moderno, amigável e às vezes peculiares. Cada ícone é reduzido a sua forma mínima, com cada ideia editada para sua essência. Os projetos garantem legibilidade e clareza mesmo em tamanhos pequenos. Alguns ícones do sistema são revelados na Figura 20.

Figura 20: Ícones do sistema.



Fonte: (Material Design, 2014)

Os ícones de produtos são 48dp; os ícones do sistema são 24dp

3.5.6 Tipografia

Roboto é o tipo de letra padrão no Android, foi refinado extensivamente para trabalhar em todo o conjunto mais amplo de plataformas suportadas. É um pouco mais amplo e redondo, dando maior clareza e tornando-o mais otimista. O Alfabeto e os números nessa fonte são demonstrados na Figura 21.

Figura 21: Fonte Roboto

ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 abcdefghijklm
 nopqrstuvwxyz
 1234567890

Fonte: (Material Design, 2014)

O Noto é o tipo de letra padrão para todos os idiomas no Chrome e no Android para todos os idiomas não cobertos pela Roboto. Demonstrações na Figura 22.

Figura 22: Fonte Noto

ABCDEFGHIJKLM	ENGLISH
朝辞白帝彩云间	SIMPLIFIED CHINESE
朝辭白帝彩雲間	TRADITIONAL CHINESE

Fonte: (Material Design, 2014)

3.6 Firebase

É um serviço em nuvem para aplicações *mobile* (Android e iOS) e *web*. É uma plataforma que oferece um *back-end* completo e é composta por SDK e console para criar e gerenciar aplicações. É hospedada e mantida pelo Google em seus *datacenters* (Firebase, 2014).

Algumas das suas funcionalidades são:

- **Analytics:** é um painel que mostra o comportamento dos usuários da aplicação, segmentação demográfica e desempenho de campanha.
- **Autenticação:** tem a capacidade de fazer autenticação de usuários via e-mail, Facebook, GitHub, Google Sign-In e Twitter.
- **Relatório de Dados:** fiscaliza os erros da aplicação em todos os dispositivos e é incluído com o Analytics para averiguar o comportamento dos usuários após as falhas.
- **Database:** é um banco de dados NoSQL para guardar dados JSON
- **Real time:** os dados são guardados no banco de dados em tempo real.
- **Offline:** garante armazenagem de dados na memória cache local, possibilitando o funcionamento da aplicação em estado off-line.

O kit de desenvolvimento de software (SDK) do Firebase possui suporte nas seguintes linguagens de programação: C++, Java, Javascript, Node.js, Objective-C e Swift.

3.7 Formato JSON

Um dos formatos utilizados para comunicação entre plataformas diferentes é o JSON (*JavaScript Object Notation*, ou em português, Notação de Objeto JavaScript). Sua sintaxe é ajustada para representar estruturas de dados obtidas em respostas de um sistema.

O JSON pode ser constituído por quatro tipos primários (*strings*, números, booleanos e nulos) e dois tipos estruturados (objetos e vetores). Um objeto é uma coleção não ordenada de zero ou mais pares chave/valor, onde cada um é uma string. Um vetor é uma sequência ordenada de zero ou mais valores. (CROCKFORD, 2006)

Esse formato foi desenhado com o propósito de ser simples, portátil, textual e um subconjunto do Javascript. O formato é exemplificado na Figura 23.

Figura 23: Exemplo de Formato JSON

```
{
  "type": "menu",
  "value": "File",
  "items": [
    {"value": "New", "action": "CreateNewDoc"},
    {"value": "Open", "action": "OpenDoc"},
    {"value": "Close", "action": "CloseDoc"}
  ]
}
```

Fonte: (CROCKFORD, 2006)

Sua indentação e o uso de espaços em branco é totalmente opcional no JSON. O mesmo exemplo pode ser escrito em uma única linha que não irá ter erro no compilador.

Devido a sua simplicidade, o JSON não mantém referência para elementos e necessita do uso das aspas para manter compatível com a sintaxe do Javascript.

3.8 REST

O autor (SOMMERVILLE, 2011) afirma que “uma *web service* é uma instância de uma ideia mais geral de um serviço”, onde um serviço é uma maneira de oferecer resultado ao solicitante sem que ele seja o consciente da execução. Os serviços foram recomendados para autorizar a integração entre sistemas de plataformas diferentes. Assim surgiu os serviços baseados em REST, os quais usam unicamente o protocolo HTTP para a troca de mensagens e fornecimento de recurso.

REST, *Representational State Transfer*, ou Transferência de Estado Representativo, é um padrão de arquitetura para aplicações web criado por Roy Fielding (FIELDING, 2000). De acordo com o autor, o REST deixa de lado os detalhes de sintaxe de protocolo e de implementação, indo direto ao ponto nas regras de negócio, na interação de seus elementos e suas limitações. Suas propriedades essenciais são comunicação cliente-servidor, comunicação sem estado, uso de cache, interface uniforme e a uso de camadas.

As aplicações REST são livres de plataforma e de linguagem e operam em cima do protocolo HTTP. Entretanto, não há formato particular para representação de dados, podendo utilizar tanto XML quanto JSON (SAUDATE, 2014).

Diante disso, (RICHARDSON e RUBY, 2008) sugere a usabilidade destas aplicações como serviços web, que foi denominado de *web services* RESTful. Eles utilizam a base acessível na web, por meio do protocolo HTTP, o qual deve obedecer às seguintes características:

- Os recursos devem ser associados a um endereço por meio de uma URI (*Uniform Resource Identifier*) com nomes significativos;
- Os serviços não guardam estado, isto é, as informações devem estar dentro da requisição;
- Deve possuir uma interface uniforme e utilizar os verbos HTTP;
- Deve dotar de links que autorizem a navegação por outros serviços relacionados.

No que tange da uniformização da interface, ainda segundo o autor, deve ser usado os métodos GET, PUT, POST, DELETE, HEAD e OPTIONS, do protocolo HTTP. Nas operações CRUD (*Create, Read, Update e Delete*) pode utilizar os verbos deste modo:

- GET para recuperar um recurso;
- PUT para criar um novo recurso;
- POST para atualizar um recurso;
- DELETE para deletar um curso;
- Outros métodos como o HEAD para recuperar metadados e OPTIONS para identificar a que chamadas um recurso suporta.

Conforme cita o (WAGH e THOOL, 2012), o REST oferece menor tráfego de dados, flexibilidade e grande simplicidade no desenvolvimento de serviços.

4. RESULTADOS E DISCUSSÃO

Para uma visão geral do sistema, foi adotado um padrão dentro da Engenharia de Software para a descrição das funcionalidades do sistema chamado UML. Assim qualquer engenheiro que nunca entrou em contato com o projeto, obterá uma grande visualização lógica da construção de um sistema.

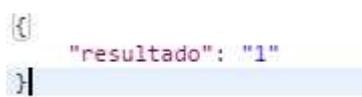
Para aprofundar as informações de um caso de uso, foi utilizado o conceito de Cenário. Dessa maneira, saberemos lidar com todas as condições quando o caso de uso for executado com sucesso ou com falha. Além do que é preciso antes de efetuar o caso de uso e o que é esperado após sua execução.

A base de dados já está devidamente treinada, isto é, utilizou-se alguns dados balanceados para reconhecimento dos padrões, prontos para analisar novos dados e fazer a classificação da situação do paciente. O classificador do trabalho é um sistema externo e faz a comunicação o servidor utilizado RESTful, onde o formato de dados em JSON e o cabeçalho da requisição estão adequadamente configuradas para cada parte do sistema (aplicativo, servidor e sistema externo) com o objetivo de conseguir o êxito da comunicação.

Quanto as telas, utilizou-se o Material Design porque o aplicativo foi construindo dentro da plataforma de desenvolvimento da Google garantindo uma padronização. Além do sistema não ser apenas restrito a tablets, podendo adaptar em qualquer tamanho de tela com apenas um desenvolvimento. Além de fornecer cores vibrantes, conjunto de ícones, efeitos de sombras e entre outros, a interface é totalmente leve para que possa lidar com qualquer hardware do dispositivo. Os elementos são bem interativos para que a usabilidade do sistema tenha uma boa experiência ao usuário.

Ao enviar os dados para o servidor, ele responderá com a classificação do paciente também com o formato JSON da Figura 24. O resultado “1” irá exibir a mensagem de “Diabético” e resultado “2” irá mostrar “Não Diabético”.

Figura 24: Resposta do servidor com a classificação do paciente



```
{
  "resultado": "1"
}
```

Fonte: Criado pelo software Postman

5. CONSIDERAÇÕES FINAIS

Quanto ao objetivo de criar layout agradável ao usuário, ele foi atingindo. Pois fornece um olhar limpo e minimalista ao projeto, fazendo a aplicação tender a ter tamanho menores de arquivos, por sua vez, levam menor tempo para serem carregados e com menor uso de dados.

A diabetes pode ser identificada em suas fases iniciais, verificando variáveis clínicas em paciente sem analisar o que foi demonstrado por testes bioquímicos. Existem outros métodos para diagnosticar a doença previamente, entretanto, foram concebidas em laboratórios médicos.

Isso fez surgir uma pergunta se as doenças podem ser determinadas sem qualquer exame físico da pessoa e sem desperdiçar tempo aguardando que a doença agrave.

Além do diabetes tipo 2, esses métodos podem ser utilizados a outras doenças como o câncer, o qual é uma das grandes causas de morte entre as pessoas por causa da ausência do conhecimento prévio. Como consequência, pode ocasionar em graves sequelas, tanto social tanto econômico.

5.1 TRABALHOS FUTUROS

As melhorias a ser desenvolvidas para o trabalho são:

- Desenvolvimento do aplicativo utilizando um framework de criação de aplicações híbridas ou nativas, permitindo um código para operar nas plataformas Android e iOS.
- Expandir para identificação ou rastreamentos de anomalias como câncer.

5.2 RESULTADOS ALCANÇADOS

- Fomento para esta pesquisa do edital FAPEMA (Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão) universal 31/2016;
- Fomento para a pesquisa do RH Laboratório de Análises Clínicas;
- Envio para o edital CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e FINEP (Financiadora de Estudos e Projetos);
- Preparação para congresso internacional;
- Parceiras: Hospital Bandeirantes de São Paulo, RH Laboratório de Análises Clínicas, Lab One, M Life Eirele, Socorrão II, CNPq, FAPEMA, So Cloud;
- Criação de *startup* pela clínica M Life Eirele;
- Apoio e embasamento para a lei federal do agente comunitário de saúde pelo seminário da Câmara Federal no Maranhão.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AMERICAN DIABATES ASSOCIATION. Diagnosis and Classification of Diabetes Mellitus. **Diabetes Care**, v. 30, p. 6, Janeiro 2007.

AMERICAN DIABATES ASSOCIATION. Economic Costs of Diabetes in the U.S. in 2012. **Diabetes Care**, v. 36, n. 4, p. 1033-1046, Abril 2013.

BARELLA, V. **Técnicas para o problema de dados desbalanceados em classificação hierárquia**. USP - São Carlos. [S.I.], p. 103. 2015.

BARTLETT, M.; MOVELLAN, ; SEJNOWSKI, T. Face recognition by independent component analysis. **IEEE Transactions on neural networks**, v. 13, n. 6, p. 1450-1464, 2002.

BORGES, L. **Um estudo da Diabetes Mellitus e Hipertensão Arterial baseado em técnicas de Data Mining aplicadas a dados da Administração Regional de Saúde do Centro**. ISEC Engenharia. Coimbra, p. 184. 2016.

BRASIL. **Glossário Temático Economia da Saúde**. 2. ed. Brasília: Editora do Ministério da Saúde, 2009. 60 p.

CHUNG, L. et al. **Non-Functional Equirements in Software Engineering**. 1. ed. [S.I.]: Springer Science & Business Media, v. 5, 1999.

CROCKFORD, D. Json specification, 2006. Disponível em: <<http://www.ietf.org/rfc/rfc4627.txt>>.

DAVIDSON, M. B. **Diabete mellitus diagnóstico e tratamento**, Rio de Janeiro, 2001. 389.

DELFOSSÉ, N.; LOUBATON, P. Adaptive blind separation of independent sources: a deflation approach. In: _____ **Signal processing**. 1. ed. [S.I.]: [s.n.], v. 45, 1995. p. 59-83.

DIBETICOOL, W. Diagnóstico precoce: como descobrir o diabetes o quanto antes? | Diabeticool. **Diabeticool**, 2016. Disponível em: <<http://www.diabeticool.com/diagnostico-precoce-como-descobrir-o-diabetes-o-quanto-antes/>>. Acesso em: 10 Novembro 2017.

FIELDING, R. **Architectural styles and the design of network-based software**. University of California. Irvine. 2000.

FIREBASE , 2014. Disponível em: <<https://firebase.google.com/>>. Acesso em: 28 Dezembro 2017.

FREEMAN, R. E. **Strategic Management: A Stakeholder Approach**. 1. ed. Boston: Harpercollins College Div, 1984.

GUEDES, G. **UML: Uma abordagem prática**. São Paulo: Novatec, 2006.

HYVÄRINEN, A. Fast and robust fixed-point algorithms for independent component analysis. **IEEE transactions on Neural Networks**, v. 10, n. 3, p. 626-634, 1999.

HYVÄRINEN, A.; KARHUNEN, J.; OJA, E. **Independent component analysis**. [S.l.]: John Wiley & Sons, 2004.

IDF DIABETES ATLAS, Brussels, n. 6, 2013.

INTERNATIONAL Diabetes Federation - What is Diabetes. **International Diabetes Federation**. Disponível em: <<https://www.idf.org/about-diabetes/what-is-diabetes.html>>. Acesso em: 19 2017 Novembro.

JACKSON, M. **Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices** (ACM Press). 1. ed. Massachusetts: Addison-Wesley Professional, 1995.

KAINDL, H. et al. Requirements Engineering and Technology Transfer. **Requirements Engineering Journal**, London, v. 7, p. 113-123, 2002.

KIM, J. et al. Effective representation using ICA for face recognition robust to local distortion and partial occlusion.. **IEEE transactions on pattern analysis and machine intelligence**, v. 27, n. 12, p. 1977-1981, 2005.

KRUCHTEN, P. **The Rational Unified Process: An Introduction**. 1. ed. [S.l.]: Addison-Wesley Professional;, 2000.

LAMSWEERDE, A. V. **Requirements engineering in the year 00: a research perspective**. International Conference on Software Engineering. Limerick: [s.n.]. 2000. p. 5-19.

LEAHY, J. L. Archives of Medicine Research. **Pathogenesis of type 2 diabetes mellitus**, 2005. 197-209.

MACAULAY, L. **Requirements Engineering**. 1. ed. [S.l.]: Springer-Verlag London, 1996.

MAIDEN, N. et al. **CREWS-SAVRE: Systematic scenario generation and use**. Third International Conference on. [S.l.]: IEEE. 1998. p. 148-155.

MATERIAL Design, 2014. Disponível em: <<https://material.io/>>. Acesso em: 28 Dezembro 2017.

MEYER, ; LEISCH, F.; HORNIK, K. The support vector machine under test. In: _____ **Neurocomputing**. 1. ed. [S.l.]: Elsevier, v. 55, 2003. p. 169-186.

MOISIADIS, F. **Prioritizing Scenario Evolution**. 4th International Conference on Requirements Engineering (ICRE'00). [S.l.]: [s.n.]. 2000. p. 85.

NGUYEN, T. O que é uma base de dados e para que serve? Tudo o que precisas para ter nota positiva. **Tudo o que precisas para ter nota positiva**, 2010. Disponível em: <<https://ricardo2aoc.wordpress.com/o-que-e-uma-base-de-dados-e-para-que-serve/>>. Acesso em: 2015 Novembro 2017.

NUSEIBEH, ; EASTERBROOK, S. **Requirements engineering: a roadmap**. International Conference on Software Engineering Proceedings of the conference on The future of Software engineering 2000. Limerick: [s.n.]. 2000. p. 35-46.

OJA, E.; ZHIJIAN, Y. The FastICA algorithm revisited: Convergence analysis. **IEEE Transactions on Neural Networks**, v. 17, n. 6, p. 1370 - 1381, Novembro 2006.

PRESSMAN, R. S. **Engenharia de software**. 5. ed. [S.l.]: McGraw-Hill, 2002.

RIBEIRO, Á. et al. Diabetes classification using a redundancy reduction preprocessor. **Research on Biomedical Engineering**, São Luís, v. 31, n. 2, p. 97-106, 2015.

RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: "O'Reilly Media, Inc.", 2008.

ROBINSON, I.; WEBBER, J.; EIFREM, E. **Graph databases**. [S.l.]: O'Reilly Media, Inc., 2013.

ROSA, R. D. S. et al. Interações por Diabetes Mellitus como diagnóstico principal na Rede Pública do Brasil, 1999 - 2001.. **Revista Brasileira de Epidemiologia**, v. 10, n. 4, 2007.

ROSSEN, B.; CARROLL, J. Scenarios, objects, and points of view in user interface design. In: _____ **Object modeling and user interface design**. [S.l.]: Addison-Wesley, 2001. p. 39-69.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. 1. ed. [S.l.]: Addison Wesley , 1999.

SANTANDER, V.; CASTRO, J. Integrating Use Cases and Organizational Modeling. **16º Simpósio Brasileiro de Engenharia de Software**, Gramado, p. 16, Janeiro 2002.

SAUDATE, A. **REST: Construa API's inteligentes de maneira simples**. [S.l.]: Casa do Código, 2014. 364 p.

SCHEFFEL, R. S. et al. Prevalência de complicações micro e macrovasculares e de seus fatores de risco em pacientes com diabetes melito do tipo 2 em atendimento ambulatorial. **Rev Assoc Med Bras**, Porto Alegre, v. 50, n. 3, p. 263-267, 2004.

SOMMERVILLE, I.; KOTONYA, G. **Requirements Engineering: Processes and Techniques** (Worldwide Series in Computer Science). 1. ed. [S.I.]: Wiley, 1998.

SOMMERVILLE, I. **Engenharia de Software**. [S.I.]: Addison Wesley, 2003.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering: A Good Practice Guide**. 1. ed. [S.I.]: Wiley, 1997.

UCI Machine Learning Repository. **UCI Machine Learning Repository: About**, 2007. Disponível em: <<http://archive.ics.uci.edu/ml/about.html>>. Acesso em: 15 Novembro 2017.

WAGH, K.; THOOL, R. A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host. **Journal of Information Engineering and Applications**, v. 2, n. 5, 2012.

WEI, T. A study of the fixed points and spurious solutions of the deflation-based fastica algorithm. **Neural Computing and Applications**, v. 28, n. 1, p. 13-24, 2017.

WHO | Diabetes. **World Health Organization**, 2017. Disponível em: <<http://www.who.int/mediacentre/factsheets/fs312/en/>>. Acesso em: 19 Novembro 2017.

WILSON, G.; JUNGNER, G. **Principles and Practice of Screening for Disease**. [S.I.]: [s.n.], 1968.

ZAVE, P. Classification of Research Efforts in Requirements Engineering. **ACM Computing Surveys**, v. 29, n. 4, p. 7, 1997.

GLOSSÁRIO

Text Field – Campo de Texto do Material Design que permite o usuário inserir, editar e selecionar texto. Geralmente residem em formulários, caixas de diálogo e pesquisas. Sua funcionalidade é validar a entrada de dados, ajudar os usuários a corrigir erros, preencher automaticamente as palavras e fornece sugestões.

Radio Buttons – Permitem a seleção de uma única opção de um conjunto.

Front-end – Lida com a parte da aplicação que interage diretamente com o usuário, o desenvolvedor deve se preocupar com sua experiência.

Banco de Dados – São coleções organizadas de dados que se relacionam de forma a criar alguma informação e dar mais eficiência durante uma pesquisa ou estudo.