

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA BIOMÉDICA**

**LEANDRO VASCONCELOS DOS REIS**

**LEITOR ÓPTICO DE LÂMINAS CITOLÓGICAS COM ARMAZENAMENTO EM  
BANCO DE IMAGENS ONLINE PARA AGILIZAR O PROCESSO DE  
VISUALIZAÇÃO EM LABORATÓRIOS DE ANÁLISES**

**DISSERTAÇÃO**

**CURITIBA  
2019**

**LEANDRO VASCONCELOS DOS REIS**

**LEITOR ÓPTICO DE LÂMINAS CITOLÓGICAS COM ARMAZENAMENTO EM  
BANCO DE IMAGENS ONLINE PARA AGILIZAR O PROCESSO DE  
VISUALIZAÇÃO EM LABORATÓRIOS DE ANÁLISES**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Biomédica, da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de “Mestre em Ciências” - Área de Concentração: Física Médica.

Orientadora: Prof.<sup>a</sup>. Dr.<sup>a</sup>. Rosangela Requi Jakubiak

**CURITIBA  
2019**

#### **Dados Internacionais de Catalogação na Publicação**

R375L Reis, Leandro Vasconcelos dos  
Leitor óptico de lâminas citológicas com armazenamento em banco de imagens online para agilizar o processo de visualização em laboratório de análises [recurso eletrônico] / Leandro Vasconcelos dos Reis.-- 2019.  
1 arquivo texto (87 f.) : PDF ; \$c 1,74 MB.

Texto em português com resumo em inglês.  
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Biomédica. Área de Concentração: Física Médica, Curitiba, 2019.  
Bibliografia: f. 81-87.

1. Engenharia biomédica - Dissertações. 2. Processamento de imagens - Técnicas digitais. 3. Laboratórios de patologia clínica - Exames. 4. Citodiagnóstico. 5. Protótipos - Desenvolvimento. 6. Reconhecimento óptico de caracteres. 7. Processamento eletrônico de dados em tempo real. 8. Computação em nuvem. 9. Imagens digitais. 10. Banco de dados em linha. 11. Validação. I. Jakubiak, Rosângela Requi, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Biomédica. III. Título.

CDD: Ed. 23 -- 610.28

**Biblioteca Central do Câmpus Curitiba – UTFPR**  
**Bibliotecária: Luiza Aquemi Matsumoto CRB-9/794**



Ministério da Educação

Universidade Tecnológica Federal do Paraná

Diretoria de Pesquisa e Pós-Graduação

## **TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº117**

A Dissertação de Mestrado intitulada “Leitor óptico de lâminas citológicas com armazenamento em banco de imagens online para agilizar o processo de visualização em laboratório de análises”, defendida em sessão pública pelo(a) candidato(a) Leandro Vasconcelos dos Reis, no dia 07 de fevereiro de 2019, foi julgada para a obtenção do título de Mestre em Ciências, área de concentração Física Médica, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Engenharia Biomédica.

BANCA EXAMINADORA:

Rosangela Requi Jakubiak, Dra –UTFPR

Charlie AntoniMiquelin, Dr– UTFPR

Carlos Henrique Kuretzki, Dr – Universidade Positivo

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 07de fevereiro de 2019.

Carimbo e Assinatura do(a) Coordenador(a) do Programa

## AGRADECIMENTOS

Primeiro agradecimento que eu faço em tudo na minha vida é para Deus, sem Ele em minha vida não sei se poderia ser quem eu sou.

Agradeço com muito carinho,

minha esposa Carolina, que esteve ao meu lado sempre, me apoiando, me incentivando e me ajudando com as estruturas de MDFs e vidros quando necessário;

meus filhos Miguel e Manuela, que entenderam por não estar junto nos finais de semana, e em alguns eventos;

minha mãe por me ajudar, apoiar e orar em todos os momentos;

ao meu tio Silas, que me trouxe para o Paraná, cuidou da minha família e acreditou no meu potencial profissional;

a minha família que sem a ajuda deles, poderia nem estar aqui;

aos amigos e família pastoral Sérgio e Fernanda, que me incentivaram a finalizar o mestrado e por confiar que estou na profissão certa como professor;

ao professor Dr. Charlie por me ajudar em todo o processo, desde o início até o fim. Além de fazer parte para tudo isso acontecer, sem os conselhos dele seria incapaz de finalizar essa etapa. Não sei como descrever em palavras o quão grato sou por tudo que ele me ajudou e proporcionou na minha vida profissional e acadêmica.

a professora Dr<sup>a</sup>. Rosangela que aceitou ser minha orientadora, e também me ajudou com toda a parte administrativa acadêmica;

ao amigo Geucimar que me ajudou nos momentos mais complicados do projeto. Me incentivando, ensinando e apoiando em tudo que era possível;

ao amigo Ricardo que disponibilizou o laboratório pessoal para poder trabalhar e pelo ânimo que me passou;

Ao professor Dr. Carlos que aceitou o convite de compor a banca avaliadora e contribuiu com correções do documento e do tema abordado;

aos meus professores que ajudaram no processo de aprendizado e principalmente no apoio para terminar a dissertação;

aos colegas de trabalho que estão sempre apoiando uns aos outros a continuar estudando.

## RESUMO

REIS, Leandro Vasconcelos dos. Leitor óptico de lâminas citológicas com armazenamento em banco de imagens online para agilizar o processo de visualização em laboratórios de análises. 87 f. Dissertação: Programa de Pós-Graduação em Engenharia Biomédica - Universidade Tecnológica Federal do Paraná. 2019.

Atualmente os laboratórios clínicos estão ofertando exames nos quais eles não possuem condições de realizá-los, devido a falta de equipamentos ou mesmo por falta de mão de obra especializada, com isso é feita a coleta dos exames em um laboratório e enviado para um segundo laboratório que fará toda a análise. O problema dessa prática é que torna o exame vulnerável a erros nos procedimentos e com demora na resposta ao paciente.

A partir desse entendimento, o projeto desenvolvido tem como objetivo o construir um protótipo para coletar lâminas citológicas, fazendo a conversão em imagens digitais e disponibilização em tempo real através de uma plataforma na nuvem, para laboratórios de análises. Dessa forma, os laboratórios podem eliminar o processo de transporte, diminuindo o tempo levado entre a coleta e o acesso das lâminas, visando melhorar o tempo de retorno dos resultados dos exames aos pacientes e diminuir os erros nos procedimentos dos exames.

O módulo coletor foi desenvolvido totalmente autônomo; em seu interior há um computador embarcado (Raspberry PI) que funciona como um microcomputador normal. Possui uma câmera digital de 5 Megapixels, especial para a Raspberry PI, fazendo a função de coletar as lâminas físicas, transformando-as em imagens digitais através de um sistema local. O sistema local de imagens foi desenvolvido com Javascript e HTML5, e roda sobre o servidor local Node.JS com CLI do Firebase no navegador da Raspberry.

A plataforma que recebe as imagens e os dados dos exames, fica *online* 24 horas por 7 dias da semana. Fica hospedada em um servidor na nuvem que mantém o acesso ao serviço e conecta-se ao Google Firebase. Este por sua vez, armazena e gerencia todos os dados e imagens do sistema.

**Palavras chave:** Coletor de Lâminas, Lâminas Citológicas, Plataforma *online*, Imagens digitais, Transporte de exames, Tempo Real.

## ABSTRACT

REIS, Leandro Vasconcelos dos. Cytological slides optical reader with online image storage to streamline the visualization process in analytical laboratories. 87 f. Dissertation: Postgraduate Program in Biomedical Engineering - Federal Technological University of Paraná. 2019.

At present the clinical laboratories are offering exams in which they are not able to perform them, due to the lack of equipment or even lack of specialized manpower, with which the exams are collected in a laboratory and sent to a second laboratory which will do the whole analysis. The problem with this practice is that it makes the examination vulnerable to procedural errors and delays in patient response.

Based on this understanding, the project aims to build a prototype to collect cytological slides, converting them into digital images and making them available in real time through a cloud platform for analysis laboratories. In this way, the laboratories can eliminate the transport process, reducing the time taken between collection and access of the slides, aiming to improve the time of return of the results of the exams to the patients and to reduce errors in the procedures of the exams.

The collector module was developed completely autonomous; Inside there is an embedded computer (Raspberry PI) that works like a normal microcomputer. It has a 5 Megapixel digital camera, especially for the Raspberry PI, which collects the physical slides, transforming them into digital images through a local system. The local image system was developed with Javascript and HTML5, and runs on the local Node.JS server with Firebase CLI in the Raspberry browser.

The platform that receives the images and the data of the exams, goes online 24 hours by 7 days of the week. It is hosted on a server in the cloud that maintains access to the service and connects to Google Firebase. This in turn stores and manages all data and system images.

**Key words:** Cytological slides, Cytological slides, Online platform, Digital Imaging, Examination transport, Real Time.

## LISTA DE FIGURAS

Figura 1: Modelos dos microscópios .....	20
Figura 2: Resolução da imagem .....	22
Figura 3: Profundidade de cor .....	23
Figura 4: Interpolação na imagem .....	24
Figura 5: Diferença entre <i>zoom</i> óptico e digital .....	25
Figura 6: Ferramenta de Compressão .....	26
Figura 7: Visão Firebase .....	34
Figura 8: Forma de funcionamento do sistema embarcado .....	38
Figura 9: Integração de recursos digitais, lógicos e físicos .....	47
Figura 10: Visão geral do projeto .....	48
Figura 11: Distribuição de laboratórios no território brasileiro .....	49
Figura 12: Modelos de Arduino .....	50
Figura 13: Modelo Raspberry PI 3 .....	51
Figura 14: Câmera de 5 Megapixels Raspberry PI .....	52
Figura 15: Módulo Coletor .....	52
Figura 16: Laboratório Clínico de Análises e Laudos .....	53
Figura 17: Habilitando a câmera Rasp .....	56
Figura 18: Capturar Imagem .....	66
Figura 19: Capturar Imagem com as ações <i>Zoom</i> e <i>Crop</i> .....	67
Figura 20: Cadastrar Imagem .....	67
Figura 21: Banco de imagens .....	68
Figura 22: Aplicando o filtro na imagem .....	68
Figura 23: Capturando Imagem .....	70
Figura 24: Tela de Cadastro .....	71
Figura 25: Imagem já disponível para visualização na Plataforma <i>online</i> .....	72
Figura 26: Imagem visualizada .....	72



## LISTA DE QUADROS

Quadro 1: Fatores críticos para amostras .....	12
Quadro 2: Grupos das amostras - Parte I .....	16
Quadro 2: Grupos das amostras - Parte II .....	17
Quadro 3: Formas de coletas .....	17
Quadro 4: Principais exames citológicos - Parte I .....	18
Quadro 4: Principais exames citológicos - Parte II .....	19
Quadro 5: HTML 5 e suas funções .....	30
Quadro 6: Principais Recursos Firebase Realtime Database .....	35
Quadro 7: Principais Recursos Cloud Storage .....	36
Quadro 8: Filtros da biblioteca Lena.JS .....	37
Quadro 9: Características das memórias eletrônicas .....	39
Quadro 10: Vantagens da nuvem .....	43
Quadro 11: Funções da Nuvem .....	44
Quadro 12: Aplicações mais utilizadas da Nuvem - Específicas .....	46
Quadro 13: Funções do Projeto .....	48
Quadro 14: Validação de modelo .....	50
Quadro 15: Requisições pelo navegador .....	60
Quadro 16: Testes primários dos componentes .....	63
Quadro 17: Testes e validações - Estrutura / Motores .....	64
Quadro 18: Testes e validações - Armazenamento das imagens - online .....	65
Quadro 19: Testes e validações - Visualização das imagens - online .....	66
Quadro 20: Processos do módulo coletor - Protótipo .....	69
Quadro 21: Objetivos Geral e Específicos I .....	77
Quadro 22: Objetivos Geral e Específicos II .....	77
Quadro 23: Objetivos Geral e Específicos III .....	78
Quadro 24: Objetivos Geral e Específicos IV .....	78
Quadro 25: Objetivos Geral e Específicos V .....	79

## LISTA DE TABELAS

Tabela 1: Profundidade de cor .....	23
-------------------------------------	----

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>12</b>
<b>1.1</b>	<b>Caracterização do problema</b> .....	<b>12</b>
<b>1.2</b>	<b>Solução</b> .....	<b>13</b>
<b>1.3</b>	<b>Objetivos</b> .....	<b>14</b>
1.3.1	Objetivo principal .....	14
1.3.2	Objetivos específicos .....	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>15</b>
<b>2.1</b>	<b>Citologia</b> .....	<b>15</b>
2.1.1	Tipos de procedimento para realização dos exames .....	16
2.1.2	Formas de utilização dos exames citológicos .....	18
<b>2.2</b>	<b>Imagens digitais</b> .....	<b>20</b>
2.2.1	Resolução de imagens .....	21
2.2.2	Profundidade de cor .....	22
2.2.3	Interpolação .....	23
2.2.4	Zoom .....	24
2.2.5	Compressão de Imagens .....	25
<b>2.3</b>	<b>Recursos computacionais para processamento de imagens citológicas</b> .....	<b>26</b>
2.3.1	Linguagens de máquina e de baixo nível .....	26
2.3.2	Linguagem de alto nível: estruturadas e orientadas a objetos .....	27
2.3.3	Programação para Sistemas Embarcados .....	28
2.3.4	Linguagens de programação .....	28
2.3.5	Servidores e recursos de integração .....	32
2.3.6	Componentes de Hardware .....	38
2.3.6.1	<i>Arquitetura geral de um sistema embarcado</i> .....	39
2.3.6.2	<i>Núcleo de Sistemas Embarcados</i> .....	39
2.3.6.3	<i>Recursos de hardware</i> .....	40
<b>2.4</b>	<b>Computação em nuvem</b> .....	<b>42</b>
2.4.1	Acesso a aplicações .....	44
2.4.2	O papel dos aplicativos na nuvem .....	45
<b>3</b>	<b>DESENVOLVIMENTO</b> .....	<b>47</b>

<b>3.1</b>	<b>Material e métodos .....</b>	<b>54</b>
<b>3.2</b>	<b>Testes e validações específicos .....</b>	<b>62</b>
<b>3.3</b>	<b>Validação do protótipo .....</b>	<b>69</b>
<b>4</b>	<b>DISCUSSÃO .....</b>	<b>73</b>
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>77</b>
<b>5.1</b>	<b>Considerações finais .....</b>	<b>79</b>
<b>5.2</b>	<b>Melhorias futuras .....</b>	<b>80</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>81</b>

## CAPÍTULO 1 INTRODUÇÃO

A utilização de laboratórios para exames de sangue, ginecológicos, dermatológicos, toxicológicos entre outros, cresce de acordo com a demanda da população. Porém alguns cuidados são necessários ao realizá-los. De acordo com o LACEN/PR (2016) os cuidados preliminares são essenciais e para que se possa realizar uma análise, a fim de que se possa liberar um resultado confiável, é necessário que os procedimentos pré-analíticos, ou seja, coleta, conservação e transporte dos materiais biológicos, sejam realizados de acordo com as normas que visam garantir a qualidade das amostras.

O Quadro 1, demonstra fatores críticos de acordo com o manual de Coletas e Envio de amostras LACEN/PR.

Quadro 1: Fatores críticos para amostras

<b>FATORES CRÍTICOS</b>
FALTA DE IDENTIFICAÇÃO CORRETA DOS PACIENTES
TRANSPORTE FEITO DE FORMA INDEVIDA
COLETA FEITA DE FORMA INDEVIDA
PERÍODO DE ENTREGA DA COLETA FORA DO PRAZO
DOCUMENTOS REQUERIDOS MAL PREENCHIDOS

Fonte: (DIAGNÓSTICOS DO BRASIL, 2016)

### 1.1 - Caracterização do problema

Alguns laboratórios ficam limitados na oferta de exames, devido ao custo para aquisição de tecnologia de ponta e manutenção de especialistas em análises. A solução para essa limitação é a coleta e envio das amostras para outros laboratórios que possuem tais tecnologias e especialistas. Com isso o tempo de envio e resposta dos exames aumentam devido aos processos de transporte e procedimentos

internos dos laboratórios de destino. Ao enviar as amostras coletadas dos pacientes para os laboratórios de análises, é necessário anexar o pedido médico preenchido com os dados clínicos, radiológicos, idade e hipóteses diagnósticas. O médico patologista não tem contato com o paciente, portanto depende exclusivamente destas informações para direcionar o tipo de análise, realização de colorações específicas, pesquisa de agentes infecciosos, dentre outros. Para exames de citopatologia geral, informações sobre a história da moléstia atual, neoplasias prévias, bem como, quimioterapia e radioterapia, são imprescindíveis para o êxito da interpretação citopatológica. O pedido médico deve ser completo e redigido de forma clara e legível. (DIAGNÓSTICOS DO BRASIL, 2016)

A conferência do espécime e da requisição é um fator importantíssimo para que não ocorram trocas ou extravios. Ela deve ser feita antes do envio ao Laboratório de Patologia e na chegada do resultado.

O prazo de liberação de exames para citologias em geral é uma média de 15 dias úteis, contando a partir da chegada ao Laboratório de Patologia (análises). (DIAGNÓSTICOS DO BRASIL, 2016)

## **1.2 - Solução**

A demora para a lâmina chegar nos laboratórios de análises e o tempo dos procedimentos internos do laboratório torna o prazo de retorno ao paciente muito longo, A solução proposta neste trabalho visa reduzir o tempo dos exames disponibilizando uma imagem da lâmina para o laboratório de análises, através de sistema computacional de coleta e envio. Com isso, espera-se reduzir o tempo de entrega das amostras, deixando no processo somente o tempo de procedimentos internos do laboratório de análises.

Para disponibilizar as imagens das lâminas ao laboratório de análises, foi criado um módulo coletor de lâminas que transforma em imagem digital e envia os dados para um sistema de banco de imagens online (nuvem). Esse sistema por sua vez, fica disponível ao laboratório de análises para agilizar o processo.

## 1.3 - Objetivos

### 1.3.1 - Objetivo principal

Desenvolver um sistema de leitura de lâminas citológicas com armazenamento feito no banco de imagens *online* para agilizar o processo de análise nos laboratórios clínicos.

### 1.3.2 - Objetivos específicos

- Desenvolver um produto que seja de simples manuseio e fácil instalação.
- Assegurar que as imagens sejam restritas aos laboratórios clínicos.
- Disponibilizar no servidor em banco de dados em tempo real, com possibilidade de acesso remoto simultâneo desde que o usuário tenha direito ao acesso.
- Garantir que o tempo de acesso as lâminas ocorram em tempo aceitável pelo usuário final.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento do leitor óptico demanda conhecimento de assuntos que vão desde questões biológicas como citologia a questões tecnológicas de óptica, visão computacional, computação em nuvem, sistemas embarcados e comunicação com banco de dados em tempo real. Este capítulo, busca explorar estes assuntos de forma a analisar sua aplicação de forma em geral, para que no próximo passo possa ser aplicado de forma específica do trabalho.

#### 2.1 - Citologia

A Citologia (atualmente, denominada de Biologia Celular) é um dos ramos das ciências naturais. Sua história está intimamente relacionada com o desenvolvimento das lentes ópticas e à combinação destas para construir o microscópio composto (do grego *mikros*, pequeno; *skopein*, ato de ver, examinar). (CCB – UFPE, 2017)

Uma de suas subáreas é a Citopatologia, área de atuação que estuda as doenças a partir de observação ao microscópio de células obtidas por esfregaços, aspirações, raspados, centrifugação de líquidos e outros métodos. Praticamente todos os órgãos e tecidos podem fornecer material para o estudo citológico, permitindo exame rápido, pouco invasivo e não-traumático. As amostras celulares são dispostas em lâminas de vidro identificadas e posteriormente coradas. (ABRALAPAC, 2018)

A técnica citológica possui grande importância no diagnóstico de algumas doenças que acometem os seres humanos e os animais. Essa é uma ferramenta fundamental no diagnóstico de tumores, função hormonal e infecções parasitárias. O exame colpocitológico, conhecido como Papanicolau, é utilizado para detectar, nas mulheres, tumores de colo de útero. Seu idealizador, dr. George N. Papanicolaou, estabeleceu em 1942 os conceitos básicos de interpretação citológica e criou um método de coloração citológica que é utilizado, universalmente, até hoje. A citopatologia analisa as células individualizadas, descamadas, expelidas ou retiradas da superfície de órgãos de diferentes partes do organismo. Como os materiais biológicos apresentam diferentes características, devido às distintas formas de



organização e composição, a coleta do material destinado à análise citológica constitui uma etapa fundamental nesse processo. Há métodos específicos para coleta de materiais distintos. Além disso, nessa fase, são definidos os tipos de procedimentos mais adequados à análise dos preparados citológicos. (MOLINARO; CAPUTO; AMENDOEIRA. 2010)

As técnicas precisam cuidar com consideráveis interferências na qualidade final do diagnóstico, como coleta do material, fixação, processamento, coloração e leitura das lâminas citológicas. (MOLINARO; CAPUTO; AMENDOEIRA, 2010)

### 2.1.1 - Tipos de procedimento para realização dos exames

Nos preparados citológicos, a origem das amostras são diversificadas, proveniente de líquidos orgânicos (urina, líquido, líquido ascítico, pericárdico, sinovial), punções aspirativas por agulha fina (pulmão, mama, tireoide, linfonodos, dentre outros), secreções (escarro, abscesso e fístula), lavados cavitários (brônquicos e bronco alveolares, vesiculares) e raspados (cervicovaginal, ocular). Segundo suas características, as amostras são divididas em três grupos como demonstram o Quadro 2 - Parte I e Parte II. (MOLINARO; CAPUTO; AMENDOEIRA. 2010)

Quadro 2: Grupos das amostras - Parte I

CLASSIFICAÇÃO DA AMOSTRA	MÉTODO DE COLETA	ORIGEM DA AMOSTRA
<b>Distensão celular (esfregaço)</b>	Raspagem <i>Swab</i>	Colpo citologia Olhos Lavado brônquico
	<i>Imprint</i> ou decalque	Lesões cutâneas Biópsias Peças cirúrgicas
	Punção aspirativa	Sangue Lavado brônquico Líquor espinhal
<b>Amostras pastosas</b>	Expectoração	Escarro
	Punção ou drenagem	Abscessos Massas necróticas

Fonte: (MOLINARO; CAPUTO; AMENDOEIRA, 2010)

Quadro 2: Grupos das amostras - Parte II

CLASSIFICAÇÃO DA AMOSTRA	MÉTODO DE COLETA	ORIGEM DA AMOSTRA
<b>Amostras líquidas</b>	Espontânea ou por cateter	Urina
	Escovação	Líquido sinovial
	Escovação ou lavado	Líquido peritoneal ou ascítico
	Punção	Líquido pleural Líquido peritoneal ou ascítico Líquido pericárdico Lavado brônquico alveolar Lavado vesical Líquido estomacal Lavado brônquico Líquido sinovial

Fonte: (MOLINARO; CAPUTO; AMENDOEIRA, 2010)

Molinaro, Caputo e Amendoeira (2010), ainda complementam dizendo que a natureza da amostra (líquida, pastosa ou sólida) irá definir a forma de coleta e o preparo do material segundo as etapas da técnica citológica escolhida. No Quadro 3 apresenta a descrição de como cada amostra é colhida.

Quadro 3: Formas de coletas

<b><i>Distensão celular (esfregaço)</i></b>	É feita ao se distender sobre uma lâmina de vidro uma leve camada de fluidos corpóreos para o exame ao microscópio.
<b><i>Lavado</i></b>	O material é colhido com o auxílio de um cateter de instilação para lavagem, contendo solução salina, de uma cavidade do organismo..
<b><i>Escovados</i></b>	O material é colhido por esfoliação da superfície de mucosas, utilizando-se uma escova.
<b><i>Impressões teciduais (imprint)</i></b>	Denomina-se impressões teciduais o procedimento em que se coloca a área lesionada do tecido em contato com a superfície de uma lâmina de vidro lisa, de forma semelhante ao procedimento para se obter impressão digital. Esse procedimento é também denominado citologia de decalque.

Fonte: (MOLINARO; CAPUTO; AMENDOEIRA, 2010)

### 2.1.2 - Formas de utilização dos exames citológicos

Exames de citologia podem ser utilizados de duas formas: para o diagnóstico e para o rastreamento.

De acordo com ONCOGUIA (2018), um exame diagnóstico só é utilizado em pacientes que apresentam sinais, sintomas ou alguma outra razão para suspeitar que uma determinada doença (como o câncer) possa estar presente. Um exame de diagnóstico classifica a doença (se presente) de forma precisa. E já no caso de um exame de rastreamento é utilizado para encontrar pessoas que possam ter uma determinada doença, mesmo antes de desenvolver quaisquer sintomas. Espera-se que um exame de rastreamento possa encontrar quase todas as pessoas que tem predisposição a alguma doença, mas nem sempre serve para provar que a doença está presente. Alguns exames de citologia, como o exame de Papanicolaou são usados principalmente para a seleção de pacientes em risco, enquanto outros podem identificar com precisão os cânceres. Quando a citologia mostra câncer, uma biópsia é também realizada para se ter certeza de qualquer achado anormal antes do tratamento ser iniciado.

De acordo com os Quadros 4 - Parte I e Parte II, são demonstrados os principais exames citológicos realizados em laboratórios.

Quadro 4: Principais exames citológicos - Parte I

<b>EXAME</b>	<b>FUNÇÃO</b>
<b>Teste De Papanicolaou</b>	Colpo citologia Oncótica ou Citopatologia Cervicovaginal
<b>Colpo citologia Hormonal</b>	Citopatologia Hormonal Isolada
	Curva Hormonal (Colpo citologia Hormonal Seriada)
<b>Citologia de Lesões Cutâneas Bolhosas</b>	Herpes E Pênfigo Vulgar
<b>Citopatologia aspirativa</b>	Órgãos Diversos

Fonte: (PATOLOGIAMÉDICA, 2018)

Quadro 4: Principais exames citológicos - Parte II

EXAME	FUNÇÃO
<p align="center"><b>Citopatologia oncótica</b></p> <p align="center">(Pesquisa Citopatológica De Células Neoplásicas)</p>	Líquidos Ou Efusões (Pleural, Ascítico, Pericárdico, Sinovial, Líquor)
	Secreção Ou Cistos Mamários
	Raspado Ou Escovado Uretral
	Urina (Citologia Urinária)
	Trato Respiratório (Secreção Nasal, Escarro, Lavado Bronquíolo Alveolar, Escovado Brônquico)
	Trato Gastrointestinal
	Lavado gástrico (Escovado Gástrico, Secreção Ocular Conjuntival, Raspado De Lesões Em Geral "Imprints")
<b>Cell Block</b> (Bloco Celular)	De Secreções, Líquidos, Aspirados

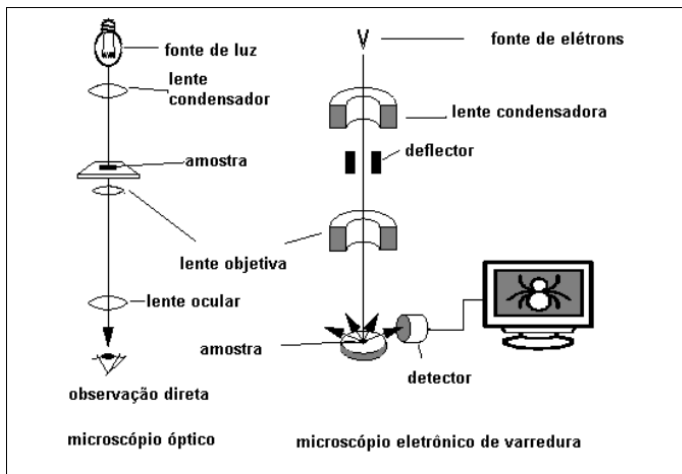
Fonte: (PATOLOGIAMÉDICA, 2018)

### Microscópio ópticos, eletrônicos e digitais

Para realização de exames citológicos, os profissionais da saúde fazem utilização intensa de microscópios que vão desde ópticos a digitais.

De acordo com Carneiro e Junqueira (2012), os microscópios ópticos, também conhecido como microscópio de luz, são aqueles que funcionam com a mescla de partes mecânicas e parte óptica com um sistema de lentes projetando um feixe de luz. Já no caso do microscópio eletrônico, existe o filamento que produz elétrons, esses são acelerados por uma diferença de potencial entre os catodos e anodos, passam por um conjunto de 3 bobinas para serem tratados e por fim são projetados numa tela fluorescente. Os autores ainda dizem que a diferença entre os modelos óptico e eletrônico é o poder resolutivo da imagem; a do microscópio eletrônico é muito maior do que o óptico, sendo que o eletrônico consegue obter imagens em escala de 0,005 nm e já o óptico consegue obter imagens em escala de 0,5 µm. Na Figura 1, apresenta a estrutura de cada modelo de microscópio.

Figura 1: Modelos dos microscópios



Fonte: (OLIANI, 2018)

Por outro lado, um microscópio digital necessita que um microscópio óptico seja equipado com uma câmera digital, porém, sem a necessidade das oculares. Uma imagem da amostra é observada e analisada diretamente em uma tela de um monitor eletrônico. Os microscópios digitais também podem ser microscópios convencionais ou microscópios compostos que tenham oculares, porém, eles também são equipados com uma câmera e salvam informações de *feedback* sobre o *status* do microscópio e os pontos de configuração da câmera. (SCHLAFFER, 2018)

Ao utilizar o microscópio como ferramenta de leitura citológica, o profissional passa a trabalhar com imagens que auxiliarão nos resultados do exame. A leitura microscópica apresenta uma imagem da lâmina ampliada de 10 a 1000 vezes; dependendo do modelo do microscópio óptico esse valor pode ter escala bem maior, sem perder qualidade na imagem. E com isso o resultado da análise fica diretamente ligado a relação da ampliação da lâmina com a qualidade da imagem. (SCHLAFFER, 2018)

## 2.2 - Imagens digitais

De acordo com Gonzalez e Woods (2010), eles definem imagem como uma função bidimensional,  $f(x, y)$ , em que  $x$  e  $y$  são coordenadas espaciais (plano), e a amplitude de  $f$  em qualquer par de coordenadas  $(x, y)$  é chamada de intensidade.

Quando  $x$ ,  $y$  e os valores de intensidade de  $f$  são quantidades finitas e discretas, chamamos de imagem digital.

De acordo com Soares (2010), há dois tipos de imagem digital:

- Imagens de rastreo que são imagens formadas por um conjunto de pontos definidos por valores numéricos, no qual cada ponto representa um pixel. E esse tipo de imagem geralmente são constituídas por imagens fotográficas.
- Imagens vetoriais são originadas a partir de desenhos (retas, pontos, curvas, polígonos simples, etc.). E costumam ser utilizadas para desenhos técnicos de engenharia.

Ao utilizar o microscópio eletrônico com as saídas por imagens digitais do tipo rastreo, o resultado da ampliação final da imagem será uma combinação entre resolução, profundidade da cor, interpolação, *zoom* e compressão. (SOARES, 2010)

A seguir serão discutidos detalhes de cada uma dessas variáveis que podem gerar as interferências na qualidade das imagens digitais.

### 2.2.1 - Resolução de imagens

Gonzalez e Woods (2010) explicam que, a resolução espacial é uma medida do menor detalhe discernível em uma imagem. Quantitativamente, a resolução espacial pode ser expressa em várias formas, sendo que as mais comuns são pares de linha por unidade de distância e pontos (pixels) por unidade de distância. De forma similar, a resolução de intensidade refere-se à menor variação discernível de nível de intensidade na imagem.

De acordo com Galvani (2017), a resolução é a medida de definição da imagem, sendo normalmente expressa para tela/monitor e impressão. Na Figura 2 pode-se notar a mesma imagem com diferentes tipos de resolução.

Figura 2: Resolução da imagem

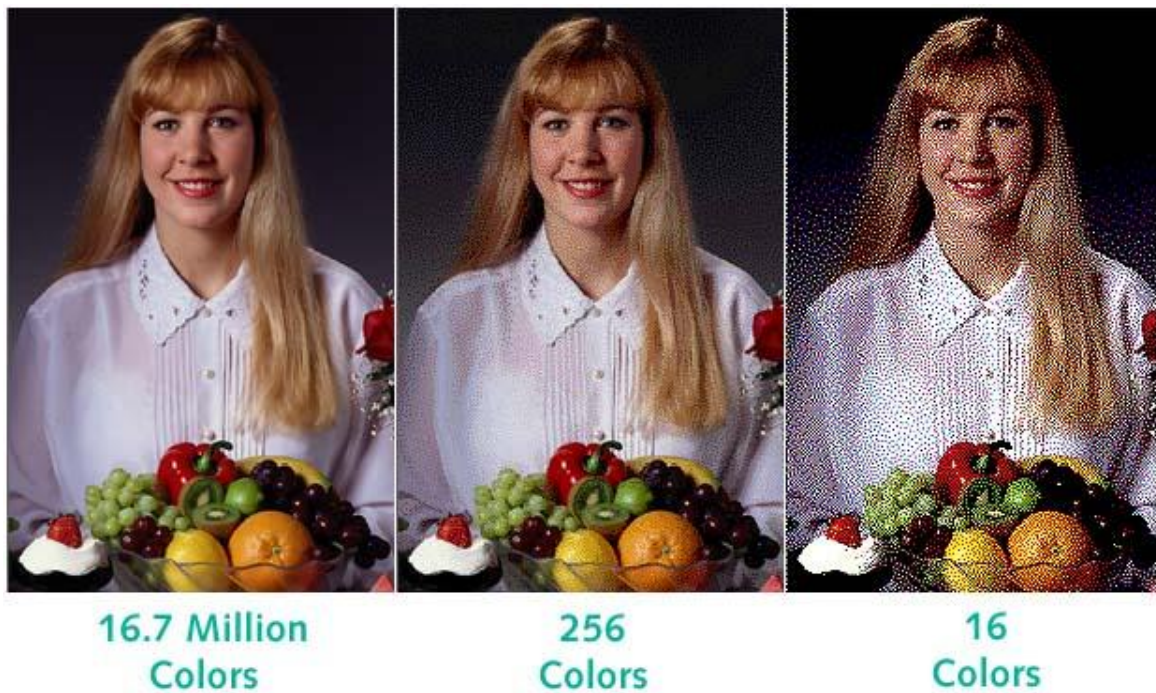


FONTE: (SCURI, 2002)

### 2.2.2 - Profundidade de cor

É um termo da computação gráfica que descreve a quantidade de *bits* usados para representar a cor de um único *pixel* numa imagem *bitmap*. Quanto maior a quantidade da profundidade da cor presente na imagem, maior é a escala de cores disponível. A Figura 3 demonstra a diferença entre a quantidade de *bits* na imagem. A imagem da esquerda está com 24 *bits*, conseguindo atingir até 16 milhões de cores aproximadamente, no meio a imagem aparece com 8 *bits* atingindo 256 cores e na imagem da direita possui 4 *bits*, atingindo 16 cores. A Tabela 1 apresenta o nível de *bits*, a quantidade de cores representada e a sua aplicação. (CORELDRAW, 2012)

Figura 3: Profundidade de cor



Fonte: (FOCUSFOTO, 2014)

Tabela 1: Profundidade de cor

Bits Por Pixel	Número de Cores Disponíveis	Observação
1	2	Monochrome
2	4	CGA
4	16	EGA
8	256	VGA
16	65536	XGA, High Color
24	16777216	SVGA, True Color
32	16777216 + Transparência	-
48	281 Trilhões	-

Fonte: (FOCUSFOTO, 2014)

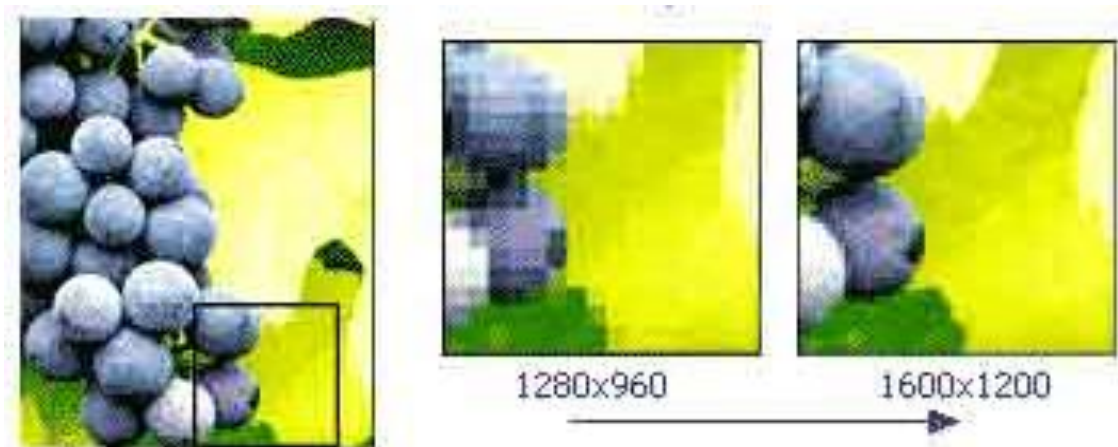
### 2.2.3 - Interpolação

De acordo com Gonzalez e Woods (2010), a interpolação é uma ferramenta básica utilizada extensivamente em tarefas como ampliação, redução, rotação e correções geométricas. Leite (2015), complementa dizendo que são processos



utilizados por *scanners*, câmeras digitais, entre outros periféricos, que, durante o processamento de suas imagens os programas de tratamento que possibilitam aumentar artificialmente o tamanho e resolução da imagem final, adicionando pontos de cores intermediárias entre os já existentes. A Figura 4 demonstra a aplicação da interpolação na imagem.

Figura 4: Interpolação na imagem



Fonte: (MORENO, 2008)

#### 2.2.4 - Zoom

É dado pelo quociente da máxima distância focal pela mínima distância focal. O *zoom* não é capaz de indicar o ângulo de visão de uma dada lente. Lentes com o mesmo fator de *zoom* podem apresentar intervalos de ângulos de visão diferentes e assim produzir resultados diferentes. (BRITO, 2014)

- *Zoom Óptico*: Aproxima as imagens através de um jogo de lentes internas do dispositivo digital. Em função disso, a imagem resultante tem menos probabilidade de ser distorcida e tremida, ou seja, ela é aproximada de maneira real.
- *Zoom Digital*: Processa a imagem em um *software* interno, que simula a aproximação. Portanto é, na verdade, uma ampliação da imagem e não uma aproximação. Com isso, a definição da imagem é reduzida.

A Figura 5 demonstra por Cardoso (2012) a diferença na prática entre a imagem original e a aplicação dos tipos de *zoom* óptico e digital.

Figura 5: Diferença entre *zoom* óptico e digital



Fonte: (CARDOSO, 2012)

#### 2.2.5 - Compressão de Imagens

A compressão de imagens baseia-se na remoção de informação redundante existente nas imagens de forma a armazenar ou transmitir esses mesmos dados de forma eficiente e ela se divide em duas categorias. (CONCI, AZEVEDO, LETA. 2009)

- *Compressão com perdas* - conduz à perda de alguma informação da imagem (que pode ou não ser aparente a olho nu). A imagem descomprimida terá uma qualidade inferior à imagem original.
- *Compressão sem perdas* - quando a compressão, seguida pela descompressão, preserva integralmente os dados da imagem.

Na Figura 6 pode-se perceber a aplicação da técnica de compressão na imagem com formato JPG. Cada passo em que se aplica a compressão, a qualidade da imagem vai piorando.

Figura 6: Ferramenta de Compressão



Fonte: (MOREIRA; SILVESTRE, 2015)

### 2.3 - Recursos computacionais para processamento de imagens citológicas

Existe uma grande diversidade de linguagens de programação, cada uma otimizada para um determinado tipo de aplicação. Elas estão constantemente sofrendo evoluções de forma a facilitar a implementação das soluções. (OLIVEIRA; ANDRADE, 2010)

As linguagens de programação podem ser divididas em linguagens de baixo nível e linguagens de alto nível. (OLIVEIRA; ANDRADE, 2010)

#### 2.3.1 - Linguagens de máquina e de baixo nível

De acordo com Aguilar (2011), as linguagens de máquina são aquelas que estão escritas em linguagens diretamente inteligíveis pela máquina (computador), já que suas instruções são cadeias binárias que especificam uma operação, e as posições de memória implicadas na operação se denominam instruções de máquina.

Quando se tenta visualizar esses códigos em um computador, é aconselhável o uso de um editor hexadecimal, pois se visualizado em um editor de texto normal, os bytes serão representados por caracteres ASCII, tornando o texto uma sequência de símbolos sem lógica. (OLIVEIRA; ANDRADE, 2010)

As linguagens de baixo nível são mais fáceis de utilizar do que as linguagens de máquina, mas também dependem da máquina em particular. As instruções em linguagem de baixo nível são instruções conhecidas como mnemônicas. (AGUILAR, 2011)

Oliveira e Andrade (2010) definem que uma linguagem de programação é dita de baixo nível quando está na camada mais inferior do software. Ou seja, quando ela é a linguagem efetiva do hardware e não seja necessário convertê-la para o entendimento do processador.

### 2.3.2 - Linguagem de alto nível: estruturadas e orientadas a objetos

As linguagens de alto nível foram desenvolvidas devido às dificuldades de programação utilizando linguagem de baixo nível. As linguagens de alto nível são mais fáceis de utilizar que as linguagens de baixo nível e máquina, devido aos seus comandos que lembram a linguagem natural humana. (COCIAN, 2004)

A programação neste caso ainda é realizada com a linguagem de máquina, mas o compilador realiza a conversão. Apesar da praticidade na maioria das vezes, o código gerado pelo compilador fica significativamente maior que o código programado diretamente na linguagem de máquina. (OLIVEIRA; ANDRADE, 2010)

Dentro das linguagens de programação em alto nível, ainda existe distinção das linguagens de programação estruturadas e das orientadas a objeto. (OLIVEIRA; ANDRADE, 2010)

A programação estruturada enfatiza a utilização de unidades separadas de programas, chamadas de módulos, que são ativadas através de comandos especiais. O desenvolvimento de programas deve ser feito de forma descendente, com a decomposição de problema inicial em módulos ou estruturas hierárquicas, de modo a dividir ações complexas em uma sequência de ações mais simples, desenvolvidas de forma mais fácil. (EDELWEISS; LIVI, 2014)

A programação estruturada vem sendo substituída pela programação orientada a objetos, mas ainda é muito utilizada no aprendizado de linguagens de programação, principalmente devido à dificuldade de entendimento da programação orientada a objetos. (OLIVEIRA; ANDRADE, 2010)

Por outro lado, o conceito de orientação a objetos implica na organização de um software em termos de objetos, incorporando suas estruturas, comportamentos e relacionamentos. (OLIVEIRA; ANDRADE, 2010)

A programação orientada a objetos é um paradigma de programação de computadores onde se usam classes e objetos, para representar e processar dados usando programas de computadores. (SANTOS, 2013)

### 2.3.3 - Programação para Sistemas Embarcados

É muito comum a utilização de linguagens de programação de alto nível para a implementação de sistemas embarcados, devido à abstração que ela fornece. Por exemplo, para implementar uma comunicação serial em um microcontrolador, diversas configurações são necessárias para o envio de apenas um caractere. Quando se implementa através de uma linguagem de alto nível, isso tudo é abstraído e é implementado com apenas uma instrução envia (dado). Quando essa instrução é convertida, o compilador se encarrega de realizar todas as configurações necessárias. (OLIVEIRA; ANDRADE, 2010)

Curvello *et al.* (2015) explicam que apesar da multiplicidade de linguagens para sistemas embarcados, deve-se sempre considerar a aplicação fim, não esquecendo dos seus *trade-offs*, como tempo de engenharia e o tempo de entrada do produto no mercado. Nesse sentido, na grande maioria das vezes, a melhor linguagem é aquela com a qual o desenvolvedor está mais familiarizado e a que melhor atende os requisitos solicitados para o sistema a se desenvolver.

### 2.3.4 - Linguagens de programação

#### **I. Linguagem C/C++**

Foi criada, influenciada e testada em campo por programadores profissionais. O resultado é que C dá ao programador o que ele quer em uma linguagem estruturada, além de conseguir eficiência similar aos códigos Assembly. (SCHILDT, 1997)

Segundo Unipan (2004), a filosofia que existe por trás da linguagem C é que o programador sabe realmente o que está fazendo. Por esse motivo, a linguagem C quase nunca “coloca-se no caminho” do programador, deixando-o livre para usar (ou abusar) dela de qualquer forma que queira.

Unipan (2004) ainda define que a linguagem C tornou-se uma das linguagens de programação mais usadas. Flexível, ainda que poderosa, a linguagem C tem sido utilizada na criação de alguns dos mais importantes produtos de software dos últimos anos. Entretanto, a linguagem encontra seus limites quando o tamanho de um projeto ultrapassa um certo ponto. Para resolver este problema, em 1980, enquanto trabalhava nos laboratórios da Bell, em Murray Hill, New Jersey, Bjarne Stroustrup acrescentou várias extensões à linguagem C e chamou inicialmente esta nova linguagem de “C com classes”. Entretanto, em 1983, o nome foi mudado para C++.

E Unipan (2004) finaliza explicando que quando C++ foi inventado, Bjarne Stroustrup sabia que era importante manter o espírito original da linguagem C, incluindo a eficiência, a natureza de nível médio e a filosofia de que o programador, não a linguagem, está com as responsabilidades, enquanto, ao mesmo tempo, acrescentava o suporte à programação orientada a objetos. Assim, o C++ proporciona ao programador a liberdade e o controle da linguagem C junto com o poder dos objetos. As características da orientação a objetos em C++, usando as palavras de *Stroustrup*, “permite aos programas serem estruturados quanto à clareza e extensibilidade, tornando fácil a manutenção sem perda de eficiência”.

## **II. HTML 5, CSS e Javascript**

HTML 5 é a mais recente evolução do padrão que define o HTML, este que consiste numa linguagem para marcação de hipertexto e sua principal característica é a possibilidade de se interligar a outros documentos web (sites) com o uso dos links. (SILVA, 2015)

De acordo com MDN (2017) o termo HTML 5 representa dois conceitos diferentes:

- É uma nova versão da linguagem HTML, com novos elementos, atributos e comportamentos
- É um conjunto maior de tecnologias que permite o desenvolvimento de aplicações e *web sites* mais diversos e poderosos. Este conjunto é chamado HTML 5 & *friends* e muitas vezes abreviado apenas como HTML5.

Criada para ser utilizada pelos desenvolvedores da web, essa linguagem de referências faz ligações a inúmeros recursos, classificados em diversos grupos, baseando-se em suas funções. O Quadro 5, mostra essa ligação. (MDN, 2017)

Quadro 5: HTML 5 e suas funções

<b>FUNÇÃO</b>	<b>DEFINIÇÃO</b>
<b>Semântica</b>	Permite você descrever mais precisamente o seu conteúdo.
<b>Conectividade</b>	Permite uma comunicação com o servidor de formas modernas e inovadoras.
<b>Offline e armazenamento</b>	Permite que páginas web armazenem dados localmente do lado do cliente e opere de forma <i>offline</i> mais eficientemente.
<b>Multimídia</b>	Viabiliza a utilização de áudio e vídeo de forma primorosa na Web Aberta.
<b>Gráficos e efeitos 2D/3D</b>	Viabiliza um leque diversificado de opções de representação gráfica.
<b>Performance e integração</b>	Fornece grande otimização de velocidade e melhor utilização do hardware do computador.
<b>Acesso ao dispositivo</b>	Viabiliza a utilização de diversos métodos e dispositivos de entrada e saída.
<b>Estilização</b>	Permite aos autores a escrita de temas mais sofisticados.

Fonte: (MDN, 2017)

De acordo com Miyagusku (2007), CSS é uma folha de estilos com a principal função de extrair a formatação de uma página do código HTML, separando-a do conteúdo propriamente dito (informações). Além de aumentar o nível de

organização, isso indica que o CSS pode definir, de antemão, a formatação de todos os elementos de uma página ou de várias páginas de um site.

Faria (2018) define a linguagem JavaScript como uma linguagem de programação que permite implementar itens complexos, tais como mostrar conteúdo dinamicamente atualizado, exibir mapas interativos e/ou gráficos 2D/3D animados. Ele também explica que o núcleo da linguagem consiste em alguns benefícios comuns da programação como armazenar conteúdo útil em variáveis, operações com pedaços de texto, executar o código em resposta a determinados eventos e trabalhar com APIs.

No topo da linguagem Javascript as Interfaces de Programação de Aplicativos (APIs) são recursos computacionais construídos para estabelecer padrões de comunicação entre sistemas. Elas geralmente se dividem em duas categorias: (FARIA, 2018)

- **APIs de navegadores** já vem implementadas no navegador, e são capazes de expor dados do ambiente do computador, ou fazer coisas complexas e úteis.
- **APIs de terceiros** não estão implementados no navegador automaticamente, e você geralmente tem que pegar seu código e informações em algum lugar da Web.

Quando é carregada uma página *web* dinâmica no navegador, estão sendo executados o HTML, CSS e JavaScript dentro de um ambiente de execução (a guia do navegador). Isso é como uma fábrica que pega a matéria prima (o código) e transforma em um produto (a página *web*). (FARIA, 2018)

O JavaScript é executado pelo motor de renderização do navegador, depois que o HTML e CSS forem traduzidos e colocados juntos em uma página *web*. Isso assegura que a estrutura e estilo da página já estão no lugar na hora em que o JavaScript for executado. (FARIA, 2018)



### 2.3.5 - Servidores e recursos de integração

De acordo com Oliveira (2017), o modelo de comunicação utilizado na internet é baseado em duas atribuições principais para os elementos envolvidos na comunicação: cliente e servidor. O servidor deve estar sempre disponível, à espera da iniciativa do cliente. O cliente aciona o servidor sempre que precisa fazer comunicação. Esse princípio difere essencialmente dos modelos de comunicação nos quais qualquer participante da rede pode iniciar a comunicação. No modelo cliente-servidor, a comunicação sempre se inicia no cliente.

Oliveira (2017) ainda define que o servidor é um *software* que mantém uma porta de comunicação aberta à espera do cliente. Sua localização, seja pelo seu endereço ou nome, deve ser conhecida por todos os clientes que querem acessá-lo. Um servidor pode receber um grande número de solicitações simultâneas de clientes, por isso, normalmente, executa em um computador de alto desempenho. Por causa da demanda de alto desempenho, os computadores de alto desempenho são geralmente denominados também servidores, apesar de o papel de servidor ser desempenhado por um software. E o cliente também é um *software*, normalmente acionado por um usuário, razão pela qual é comum que tenha interface gráfica amigável. Um navegador *web* como Google Chrome, Mozilla Firefox ou Internet Explorer é um exemplo de cliente. Cabe ao cliente iniciar a comunicação com o servidor, seja acionada diretamente pelo usuário ou de forma automática, em resposta a um evento ou ação externa.

Faria (2018) explica que ao ser visualizada uma página *web* dinâmica, o código do lado do cliente é baixado, executado e exibido pelo navegador. Já os códigos do lado do servidor, por outro lado, são executados no servidor e o resultado da execução é baixado e exibido no navegador.

NodeBR (2016) ainda explica que o Javascript no lado do servidor pode ser um conceito novo para todos que trabalharam exclusivamente com o Javascript no lado do cliente, mas a ideia em si não é tão absurda – porque não usar a mesma linguagem de programação no cliente que você usa no servidor? O lado do servidor na verdade não é diferente do lado do cliente. Claro que não há botões sendo pressionados e não há campos de texto sendo escritos, mas em um nível mais alto, os eventos estão ocorrendo.

Por que a orientação a eventos é ideal para o Node.JS? Javascript é uma excelente linguagem para programação orientada a evento, porque ela permite funções anônimas e encerramentos, e o mais importante, a sintaxe é familiar para quase todos que já programaram na vida. As funções de *callback* que são chamadas quando um evento ocorre podem ser escritas no mesmo lugar onde você captura o evento. Fácil para desenvolver, fácil para manter. Basta escutar um evento, escrever uma função de *callback*, e o Node.JS toma conta de tudo. (NODEBR, 2016)

NodeBR (2016) explica que o servidor Node estabeleceu o objetivo número um que é “fornecer uma maneira fácil para construir programas de rede escaláveis”. Qual é o problema com os programas nos servidores atuais? O gargalo em toda a arquitetura de aplicações *web* (incluindo velocidade de tráfego, velocidade do processador e velocidade da memória) é o número de conexões concorrentes que o servidor pode manipular. O servidor Node resolve esta questão trocando a maneira como a conexão é tratada no servidor. Ao invés de criar uma nova OS *thread* a cada conexão (e alocar a memória anexa a ela), cada conexão dispara um evento executado dentro da *engine* de processos do Node. Ainda alega que um servidor rodando, pode suportar dezenas de milhares de conexões simultâneas.

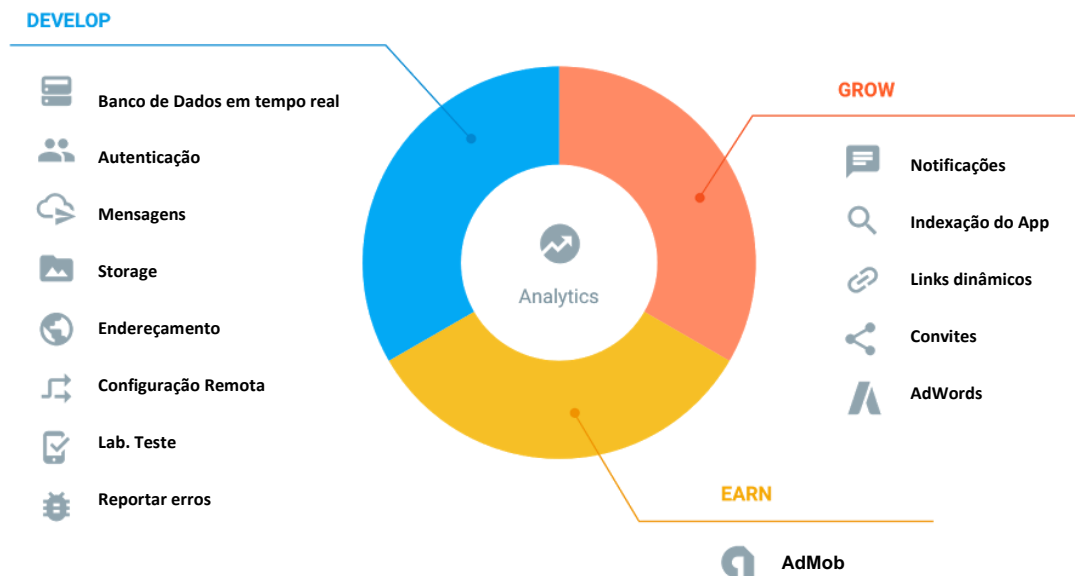
A integração e comunicação entre os diferentes sistemas é realizado através de *web services*. Essa solução possibilita que aplicações desenvolvidas em diferentes tecnologias possam interagir mesmo que sejam escritas em diferentes linguagens e executem em diferentes plataformas. As funções e os dados são definidos e enviados em formato XML. Formatos intermediários, como JSON, CSV ou XML, são usados para que as aplicações disponibilizem e acessem os métodos remotos. E a comunicação é toda encapsulada em requisição HTTP, o que facilita seu acesso e o controle em *firewalls*. (OLIVEIRA, 2017)

A proposta segue tendências já utilizadas em diversas outras tecnologias como RPC, Activex, Corba ou Java RMI, que possibilitam que aplicações distribuídas se comuniquem entre si, trocando informações, chamando métodos ou objetos remotos. *Web services* têm o diferencial de se utilizar da web para realizar o acesso aos métodos remotos e dados remotos. Utilizam, assim, os mesmos princípios e servidores *web*, incluindo comunicação em modo texto e acesso cliente/servidor. (OLIVEIRA, 2017)

## Plataforma de integração Google Firebase

O Firebase é uma plataforma de integração da Google que facilita e torna flexível o desenvolvimento de aplicativos. Pode-se observar na Figura 7 todas as ferramentas disponibilizadas por ela. (MAES, 2018)

Figura 7: Visão Firebase



Fonte: (MAES, 2018).

A plataforma móvel agiliza o desenvolvimento de aplicativos, fornece recursos de alta qualidade, disponibilidade e escalabilidade para os usuários. Composto de características complementares que podem misturar e combinar para atender às necessidades dos desenvolvedores. (MAES, 2018)

O Realtime Database do Firebase é um banco de dados não relacional (NoSQL) que permite a distribuição de conteúdos *cross-platform* e com a possibilidade de trabalho *offline*. Com o Realtime Database não se faz necessária a criação e configuração de servidores ou APIs. É ótimo para validar ideias de *apps* e soluções web pois não requer manutenção de infraestrutura. (CARDOSO, 2018)

O Quadro 6, mostra os principais recursos que a ferramenta Firebase Realtime Database possui e proporciona aos usuários.

Quadro 6: Principais Recursos Firebase Realtime Database

Recurso	Descrição
<b>Em tempo real</b>	Em vez de solicitações HTTP típicas, o Firebase Realtime Database usa a sincronização de dados. Sempre que os dados são alterados, todos os dispositivos conectados recebem essa atualização em milissegundos.
<b>Offline</b>	Os apps do Firebase permanecem responsivos mesmo <i>offline</i> , pois o SDK do Firebase Realtime Database mantém seus dados em disco. Quando a conectividade é restabelecida, o dispositivo cliente recebe as alterações perdidas e faz a sincronização com o estado atual do servidor.
<b>Acessível em dispositivos clientes</b>	O Firebase Realtime Database pode ser acessado diretamente de um dispositivo móvel ou navegador da Web, sem um servidor de aplicativos. A segurança e a validação de dados estão disponíveis por meio de regras de segurança baseadas em expressão do Firebase Realtime Database, executadas quando os dados são lidos ou gravados.

Autor: (FIREBASE, 2018).

O Cloud Storage foi criado para os desenvolvedores de app armazenarem e veicularem conteúdo gerado pelo usuário como, por exemplo, fotos ou vídeos. (FIREBASE, 2018)

O Cloud Storage para Firebase é um serviço de armazenamento de objetos poderoso, simples e econômico criado para a escala do Google. Com os SDKs do Firebase para Cloud Storage, o desenvolvedor utiliza a segurança do Google para fazer o *upload* e o *download* de arquivos nos aplicativos Firebase, independentemente da qualidade da rede. (FIREBASE, 2018)

O Quadro 7, mostra os principais recursos que a ferramenta Cloud Storage possui e proporciona aos usuários.

Quadro 7: Principais Recursos Cloud Storage

Recurso	Descrição
<b>Operações confiáveis</b>	Com os SDKs do Firebase para Cloud Storage, o uploads e downloads são feitos independentemente da qualidade da rede. Os uploads e downloads são mais confiáveis, o que significa que eles são reiniciados no ponto em que foram interrompidos, poupando tempo e largura de banda dos usuários.
<b>Segurança potente</b>	Os SDKs do Firebase para Cloud Storage estão integrados ao Firebase Authentication para fornecer uma autenticação simples e intuitiva para os desenvolvedores.
<b>Alta escalabilidade</b>	O Cloud Storage para Firebase foi projetado para suportar a escala de exabyte que o seu app terá quando se tornar um sucesso.

Autor: (FIREBASE, 2018).

A maioria dos *apps* precisa reconhecer a identidade do usuário. Ter essa informação permite que um *app* salve os dados do usuário na nuvem com segurança e forneça a mesma experiência personalizada em todos os dispositivos do usuário. (FIREBASE, 2018)

O Firebase Authentication fornece serviços de *back-end*, SDKs fáceis de usar e bibliotecas de IU prontas para autenticar usuários no seu *app*. Ele oferece suporte à autenticação por meio de senhas, números de telefone e provedores de identidade federados como Google, Facebook, Twitter e muito mais. (FIREBASE, 2018)

O Firebase Authentication é estreitamente integrado a outros serviços do Firebase e aproveita os padrões do setor, como OAuth 2.0 e OpenID Connect, para que possa ser facilmente integrado ao seu *back-end* personalizado. (FIREBASE, 2018)

### **Biblioteca Lena.JS**

De acordo com Fellipe (2013), ele diz que Lena.JS é uma biblioteca de código aberto com funções para manipulação de imagens utilizando a API do Javascript para o elemento canvas do HTML. Oficialmente, o canvas é “uma tela de bitmap dependente de resolução que pode ser usada para a renderização de elementos gráficos ou outras imagens visuais rapidamente”. Em termos leigos, o canvas é um elemento novo em HTML5, que permite que se desenhem elementos gráficos

usando JavaScript. Ele pode ser usado para renderizar texto, imagens, gráficos, linhas gradientes e outros efeitos dinamicamente. (MICROSOFT, 2013)

O Quadro 8 mostra como é aplicado cada filtro na imagem.

Quadro 8: Filtros da biblioteca Lena.JS

<b>Filtro</b>	<b>Descrição</b>
<b><i>Escala de Cinza</i></b>	A partir das primitivas vermelho, verde e azul (da escala RGB), realiza-se o cálculo $0.2126 * (\text{escala do vermelho}) + 0.7152 * (\text{escala do verde}) + 0.0722 * (\text{escala do azul})$ . Esses valores estão relacionados a natureza da sensibilidade do olho humano.
<b><i>Sépia</i></b>	O efeito sépia é obtido a partir de uma imagem em tons de cinza ( <i>grayscale</i> ), depois se acrescenta uma intensidade de 40 para o tom vermelho, 20 para o tom verde e uma redução de 20 para o tom azul. Deixando a imagem como se tivesse envelhecida.
<b><i>Vermelho</i></b>	Apenas tem o objetivo de separar a componente responsável pelo vermelho na imagem.
<b><i>Verde</i></b>	Apenas tem o objetivo de separar a componente responsável pelo verde na imagem.
<b><i>Azul</i></b>	Apenas tem o objetivo de separar a componente responsável pelo azul na imagem.
<b><i>Inverter</i></b>	Filtro responsável por realizar uma operação que altera o valor de um pixel para seu inverso, realizando uma diferença em cada componente em relação ao branco (255).
<b><i>Thresholding</i></b>	A partir das primitivas vermelho, verde e azul (da escala RGB), realiza-se o cálculo $0.2126 * (\text{escala do vermelho}) + 0.7152 * (\text{escala do verde}) + 0.0722 * (\text{escala do azul})$ . A partir dessa conversão para escala de cinza ( <i>grayscale</i> ), foi prefixado um limite ( <i>Thresholding</i> ), onde se o pixel originado na escala de cinza fosse acima de 128, seria convertido para branco, caso o contrário, para preto. Gerando no final uma imagem composta apenas pelas cores preto e branco.

Fonte: (FELLIPE, 2013)

Fellipe (2013), explica que a aplicação dos filtros citados no Quadro 9 é feita em cada pixel via o atributo *data* do *ImageData* retornado do elemento *canvas*. Vale

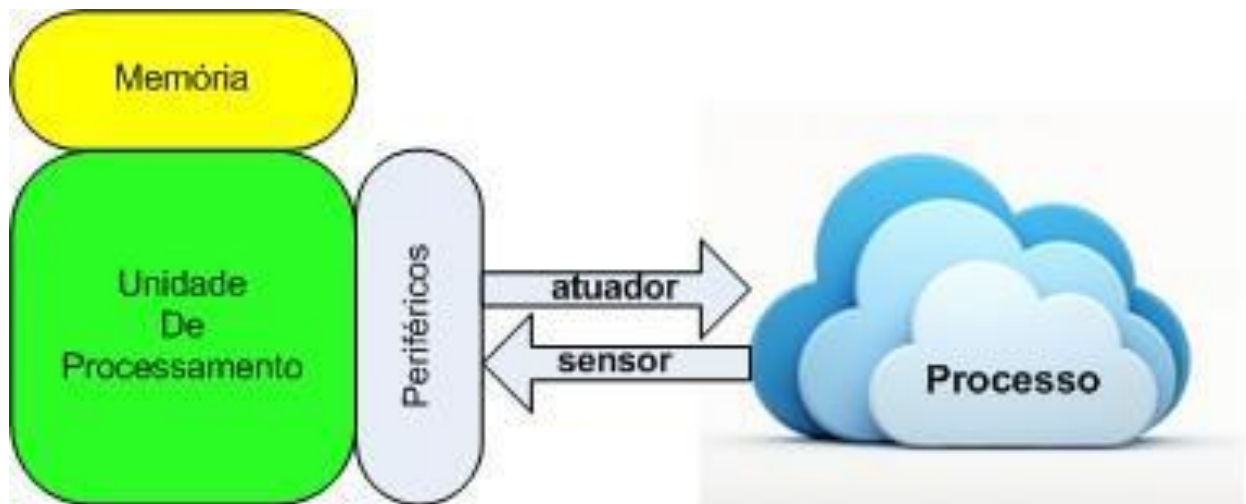
ressaltar que o pixel é um vetor constituído de 4 componentes, vermelho, verde, azul e alfa.

### 2.3.6 - Componentes de Hardware

Os sistemas embarcados são sistemas que possuem uma capacidade de processamento de informações vinda de um *software* que está sendo processado internamente nessa unidade. Ou seja, o *software* está embarcado na unidade de processamento (*firmware*). (OLIVEIRA; ANDRADE, 2010)

De acordo com Delai (2013), sistemas embarcados estão relacionados ao uso de *hardware* (eletrônica) e *software* (instruções) incorporados em um dispositivo para resolução de um problema pré-definido. E o seu objetivo principal é o de controlar processos, sendo que estes podem ser um simples acender e apagar de lâmpadas automatizadas, até gerenciamento autônomo de um avião (piloto automático). Isso é feito por intermédio dos periféricos, que são escalados e dimensionados com base no problema alvo. A Figura 8, demonstra como funciona todo o processo que envolve o sistema embarcado.

Figura 8: Forma de funcionamento do sistema embarcado



Fonte: (DELAI, 2013)

### 2.3.6.1 - Arquitetura geral de um sistema embarcado

Internamente a um sistema embarcado, o item mais evidente é a memória, pois ele necessita de um local para armazenar o *software* embarcado. Não apenas para isso, pois existem memórias que são utilizadas para armazenar dados temporários vindos de processamentos. Também existe uma memória que armazena as instruções básicas do processador. No Quadro 9, explica sobre as características das memórias eletrônicas. (OLIVEIRA; ANDRADE, 2010)

Quadro 9: Características das memórias eletrônicas

Recurso	Descrição
<b>Tempo de Acesso</b>	é o tempo necessário para acessar a memória e realizar uma operação de leitura ou gravação.
<b>Capacidade</b>	quantidade efetiva de dados que podem ser armazenados no interior da memória.
<b>Não Volatilidade</b>	capacidade de a memória manter seus dados mesmo quando não houver energia elétrica.
<b>Tempo de Latência</b>	é o intervalo mínimo entre cada operação de leitura ou escrita na memória. Ele deve ser essencialmente respeitado, pois pode acarretar na utilização de espaços indevidos da memória, podendo causar a perda ou manipulação incorreta de dados.

Autor: (OLIVEIRA; ANDRADE, 2010)

### 2.3.6.2 - Núcleo de Sistemas Embarcados

De acordo com Oliveira e Andrade (2010), explicam que primeiramente vieram os microprocessadores, substituindo milhões de transistores nos computadores, que chegavam a ocupar um andar de um prédio e após essa invenção diminuíram consideravelmente de tamanho. São usados principalmente para processamentos complexos e possuíam alto custo. Dentre algumas aplicações estão presentes nos microcomputadores pessoais, em aparelhos eletrônicos de uso doméstico e nos equipamentos médicos. E que diferentemente dos microprocessadores, os microcontroladores, em geral, possuem todos os periféricos necessários num único *chip*. Seu tamanho também é muito pequeno, mesmo



contendo vários periféricos, como memórias, barramentos, *timers*, portas de comunicação, conversores de sinal analógico em digital etc. Eles possuem desempenho menor que os microprocessadores, mas são ideais em aplicações que necessitam de menores dimensões, tempo e custos.

Oliveira (2017) completa dizendo que as funcionalidades agregadas aos microcontroladores inicialmente se resumiam a interfaces de entrada e saída (I/O) e foram agregando, a cada nova versão ou produto, memória RAM, memória EPROM para programas e dados e circuitos de oscilador (*clock*), interfaces de comunicação (serial, USB) e, mais recentemente, interfaces de rede, Ethernet, *WiFi* e Bluetooth. Sua grande vantagem é integrar, no mesmo componente, as diversas interfaces de entrada e saída (I/O), além de outras funcionalidades, como o circuito de oscilador e memória. As interfaces de entrada e saída são cruciais para definir se um determinado microcontrolador atende a um projeto de integração de sistemas embarcados. Um microcontrolador com todas essas funcionalidades integradas é capaz de atender a um número de aplicações sem precedentes. A única coisa de que ele precisa é energia, e, ainda assim, muito pouca energia, visto que vários microcontroladores atuais são classificados como de baixíssimo consumo de energia. Ou seja, uma bateria poderia mantê-los em funcionamento por um bom tempo, sem necessidade de carga ou substituição.

#### 2.3.6.3 - Recursos de hardware

### **Raspberry PI**

Inicialmente, o Raspberry PI foi proposto como uma solução de computador popular, executando uma versão do sistema operacional Linux, com interface gráfica e pacotes de aplicativos de código aberto. Seu sucesso, no entanto, abriu espaço para várias outras aplicações, especialmente em sistemas embarcados. Suas interfaces de entrada e saída, bem como seu tamanho reduzido e as demais interfaces, facilitam vários tipos de aplicações. (OLIVEIRA, 2017)

Upton e Halfacree (2013) explicam que o projeto Raspberry PI possui processador da família ARM, sendo que o coração do sistema é o processador multimídia Broadcom BCM2835, do tipo SoC (*system on chip* – em português,

sistema em um *chip*). E por isso é capaz de operar apenas com alimentação de energia de 1A e 5Vcc, fornecida pela sua porta micro-USB.

O sistema operacional Linux RaspbianOS, fusão entre Raspberry PI e sistema operacional Debian, possibilita a integração de vários tipos de periféricos cujos controladores estão disponíveis para esse sistema, incluindo câmeras, teclados, mouses, impressoras, monitores e telas, scanners, leitores biométricos, entre outros. Logo, as aplicações podem ser incrementadas com diversos tipos de dispositivos. (OLIVEIRA, 2017)

## Arduino

Para começar a falar do Arduino é preciso esclarecer que não se trata de um microcontrolador, mas, sim, de uma plataforma de desenvolvimento de sistemas embarcados de baixo custo aberta ou livre. Assim, não está vinculado a nenhum fabricante específico, embora a maioria dos módulos disponíveis utilize microcontroladores da Atmel. (OLIVEIRA, 2017)

O Arduino é uma plataforma micro controladora de desenvolvimento associada com uma linguagem de programação intuitiva que desenvolve utilizando o ambiente de desenvolvimento integrado (IDE). Ao equipar o Arduino com sensores, atuadores, luzes, alto falantes, módulos adicionais e outros circuitos integrados, pode-se transformar o Arduino em um cérebro programável para praticamente qualquer sistema de controle. (BLUM, 2016)

A popularização do Arduino trouxe algumas vantagens importantes para o seu desenvolvimento. Vários módulos, conhecidos como *shields*, começaram a ser produzidos e consumidos em larga escala, tornando-se também acessíveis e muito úteis na integração de sistemas embarcados. Há vários módulos disponíveis incluindo sensores, atuadores e módulos de comunicação. (OLIVEIRA, 2017)

Os módulos de comunicação incluem módulos *WiFi*, *Ethernet* e *Bluetooth*. Esses módulos servem para interligar o Arduino a outros dispositivos em aplicações de Internet das Coisas. Os módulos *WiFi* e *Ethernet* são usados para ligar o Arduino à *internet*. Para isso estão disponíveis diversas bibliotecas com as pilhas de protocolo TCP/IP e *WiFi*, que possibilitam a interligação nos diversos modos de WiFi, bem como algumas aplicações TCP/IP, como a *web*. Há bibliotecas para que o Arduino funcione como cliente *web*, enviando solicitações e acessando dado na

*web*, ou como servidor, disponibilizando acesso às suas informações por meio de protocolo HTTP. (OLIVEIRA, 2017)

### **Software para programação e gravação no Arduino**

O *software* IDE Arduino é, em primeira instância, uma ferramenta de edição de linguagem de alto nível C; mas, em segunda instância, é quem compila o código nessa linguagem e o converte para Assembly, para que ele possa converter em código binário e embarcar o código no Arduino. (JUNIOR; SILVA, 2015)

O Arduino UNO utiliza um microcontrolador específico (ATMEL ATMEGA 16U2) para controle da comunicação USB com o exterior. Esse microcontrolador é o responsável pela forma transparente como funciona a placa Arduino UNO, possibilitando o *upload* do código binário gerado após a compilação do programa feito pelo usuário. (JUNIOR; SILVA, 2015)

### **2.4 - Computação em nuvem**

De acordo com NIST (2011), a computação em nuvem é um modelo para habilitar o acesso por rede ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos de computação (como redes, servidores, armazenamento, aplicações e serviços) que possam ser rapidamente provisionados e liberados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços.

Um termo que tem feito sucesso nos últimos tempos na computação é *cloud computing*, ou computação na nuvem. É difícil dizer onde começa a nuvem e até onde vai, visto que boa parte dos seus conceitos já é usada desde os primórdios da internet. Basicamente, pode se dizer que todos estão usando a "nuvem". Se os serviços essenciais da rede, sejam armazenamento, processamento ou acesso, são hospedados em servidores externos. (OLIVEIRA, 2017)

O conceito de servidor externo deve considerar o cenário inicial da *internet*, no qual todas as empresas que queriam estar presentes na internet precisavam instalar seus próprios servidores, fossem de armazenamento, bancos de dados, *web* ou *e-mail*. Isso exigia um bom parque computacional das empresas, procedimentos de *backup* constantes, além de equipamentos para suprir falhas de energia. Com o

tempo, esses serviços começaram a ser transferidos para *datacenters* externos. Em síntese, podemos dizer, então, que computação na nuvem é, basicamente, terceirizar os servidores. (OLIVEIRA, 2017)

### Vantagens e funções da Nuvem

Essencialmente, a nuvem oferece cinco grandes vantagens como pode ser visto no Quadro 10.

Quadro 10: Vantagens da nuvem

VANTAGEM	EXPLICAÇÃO
<b>Mensurar o serviço ofertado</b>	Os sistemas na nuvem automaticamente controlam e otimizam o uso dos recursos através de medições em um nível de abstração apropriado para o tipo de serviço. <sup>1</sup>
<b>Ofertar um serviço “em demanda”</b>	Atualmente, uma empresa pode precisar de pouco espaço para armazenar seus dados, mas amanhã poderá precisar de mais. <sup>2</sup>
<b>Rápida elasticidade para crescimento</b>	A nuvem naturalmente já é preparada para ser elástica (para crescer), em virtude da demanda dos serviços ofertados. <sup>2</sup>
<b>Acesso por uma rede organizada</b>	Os recursos estão disponíveis através da rede e são acessados através de mecanismos padronizados. <sup>1</sup>
<b>Oferta de grupo</b>	Pool de serviços de apoio ao usuário. <sup>2</sup>

Fonte: <sup>1</sup> (NIST, 2011) e <sup>2</sup> (MOLINARI, 2018)

De acordo com o Quadro 11 descrito por Molinari (2018) pode-se afirmar que a nuvem tem sete funções essenciais.

Quadro 11: Funções da Nuvem

FUNÇÃO
Aprimorar processos das empresas tornando-os mais simples
Reduzir custos
Apoiar o aprendizado e a competência dos funcionários, compartilhando o conhecimento.
Incentivar a integração
Melhorar a comunicação interna e externa na empresa
Aumentar a produtividade
Permitir, a médio e longo prazos, que a empresa foque mais nas pessoas (e seus funcionários) e em seu negócio, e menos na tecnologia que apoia seu negócio.

Fonte: (MOLINARI, 2018)

#### 2.4.1 - Acesso a aplicações

Aplicações *web* foram as primeiras a ir para a nuvem. Apresentam um modelo distribuído, podem ser programadas em diversas linguagens populares e se hospedar em uma infinidade de servidores. Dispositivos IoT podem se conectar à *web* para enviar e receber informações e comandos ou podem, ainda, disponibilizar suas informações em servidores *web*, dando origem ao termo *Web das Coisas*, ou *Web of Things (WoT)*. (OLIVEIRA, 2017)

O assunto *Web das Coisas* tem uma comunidade de desenvolvedores, pesquisadores e projetistas que trabalham no futuro da *web*. Novos padrões estão sendo desenvolvidos, de forma aberta, flexível e escalável, que vão possibilitar a interligação de todos os dispositivos IoT à *web* para construir aplicações *web* mais fáceis de usar e integrar. Diversos métodos podem ser usados para acessar os dados dos dispositivos IoT ou enviar comandos, desde os métodos tradicionais até os serviços *web* conhecidos como *Web Services*. (OLIVEIRA, 2017)

A *internet* usa HTTP (*Hyper Text Transfer Protocol*, ou Protocolo de Transferência de Hipertexto) como seu protocolo básico. O HTTP suporta vários métodos de transmissão de dados, mas os dois métodos mais populares: *GET* e *POST*. Esses métodos executam a mesma tarefa e seu código é muito semelhante, mas há uma pequena variação no formato de seus pedidos. (JAVED, 2017)

O método *GET* tem um limite em relação à quantidade de dados que ele pode transmitir em comparação com o método *POST*, que não tem tal limitação. (JAVED, 2017)

O *POST* também é considerado mais seguro em comparação com o *GET*. Com base em seus requisitos, você pode decidir qual método é mais adequado para você. (JAVED, 2017)

Os principais métodos de acesso web são *POST* e *GET*. Esses métodos são acionados no cliente *web* para acessar dados no servidor. Ambos possibilitam o envio de parâmetros e recebem, como resposta, uma página em HTML, que pode conter os dados de resposta à solicitação. Uma página *web* é normalmente criada com um formulário para chamar os métodos *POST* ou *GET*, mas a chamada pode ser automatizada em programação para IoT. (OLIVEIRA, 2017)

#### 2.4.2 - O papel dos aplicativos na nuvem

Existem algumas aplicações que até 2010 eram as mais usadas: aplicações de armazenamento e de banco de dados eram as mais utilizadas. Com a expansão da computação em nuvem para outros segmentos, houve a ampliação de aplicações de uso geral, como aplicativos de compartilhamentos, produtividade e de sincronização, entre outras. O que aconteceu é que a nuvem, inteligente, se expandiu para fora da empresa. No Quadro 12 descrito por Molinari (2018), estão listadas as aplicações mais utilizadas atualmente na nuvem pelas empresas.

Quadro 12: Aplicações mais utilizadas da Nuvem - Específicas

<b>Aplicação</b>	<b>EXPLICAÇÃO</b>
<b>Armazenamento</b>	A nuvem possui aplicativos tanto para empresas quanto para usuários externos ou na <i>web</i> (como o Dropbox) que armazenam arquivos de vídeos, texto, apresentações etc.
<b>Bancos de dados</b>	A nuvem se tornou um grande repositório que disponibiliza o software de bancos de dados e de armazenamento dos dados na nuvem, basta acessar e configurá-lo.
<b>Produtividade e colaboração</b>	Ajudam na produtividade e na colaboração de um grupo de pessoas qualquer, ou dentro da empresa.
<b>Escritório</b>	Englobam o espectro de aplicações, como e-mail, editores de texto, editores de imagens e fotos, editores de apresentação, softwares de comunicação e reunião (como o WebEx Cloud).
<b>Desenvolvimento e testes</b>	Apoiam (ou ajudam diretamente) no desenvolvimento e testes de aplicações na nuvem ou fora dela. Tudo depende do fornecedor e dos recursos por ele apresentados.
<b>Customizados e criados para nuvem</b>	A empresa cria aplicações de tipos quaisquer para ficarem na nuvem, seja para a própria empresa ou para vender o software criado para outros.

Fonte: (MOLINARI, 2018)

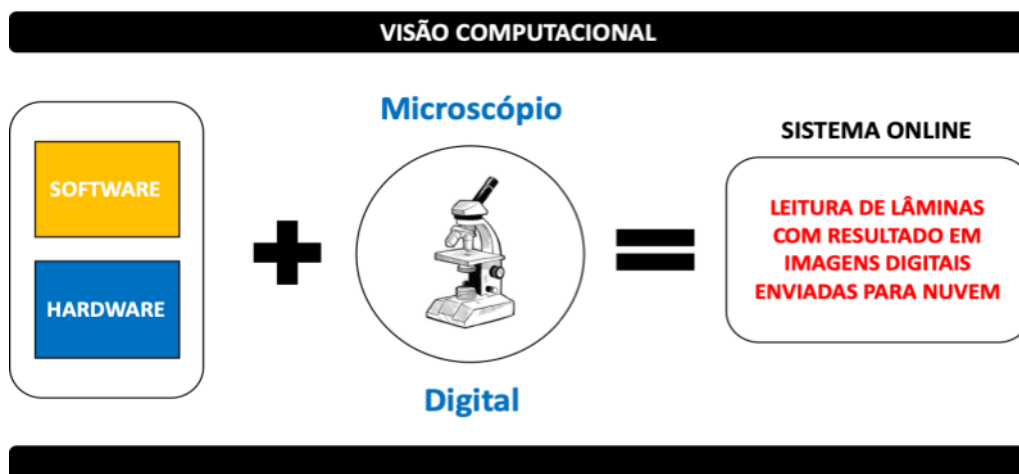
### CAPÍTULO 3 DESENVOLVIMENTO

O projeto do leitor óptico de lâminas cito patológicas com armazenamento no banco de imagens *online* para agilizar o processo de análise nos laboratórios clínicos, visa o desenvolvimento do sistema de apoio à análise clínica. Ele poderá ser utilizado por qualquer laboratório a nível nacional para processar determinados exames e/ou de enviar/transportar as coletas até outros laboratórios. Pode ser utilizado principalmente em análises que precisam ser processadas em intervalos de tempo reduzidos e as condições de transporte não são favoráveis, e/ou com exames específicos em que o tempo de resposta precisa ser rápido.

Nos métodos atuais, não interfere qual modelo é utilizado para analisar as lâminas nos laboratórios de análises, porém para o funcionamento do projeto, será adotado o conceito do microscópio digital, observando a imagem natural será criada uma cópia digital, buscando reter o máximo de características da imagem natural.

Os recursos computacionais têm o objetivo de auxiliar no processamento de imagens da área da citológica. Inicialmente as imagens digitais geradas a partir das coletas das lâminas serão disponibilizadas na nuvem para o setor responsável de análises de exames. Parte do processo de preparação das imagens será realizado por algoritmos de visão computacional que ajudarão no tratamento das imagens que serão enviadas aos laboratórios. Mas para isso é importante entender conceitos computacionais de *software* e *hardware* que estarão envolvidos no processo. A Figura 9 representa como acontece a integração computacional com a citologia.

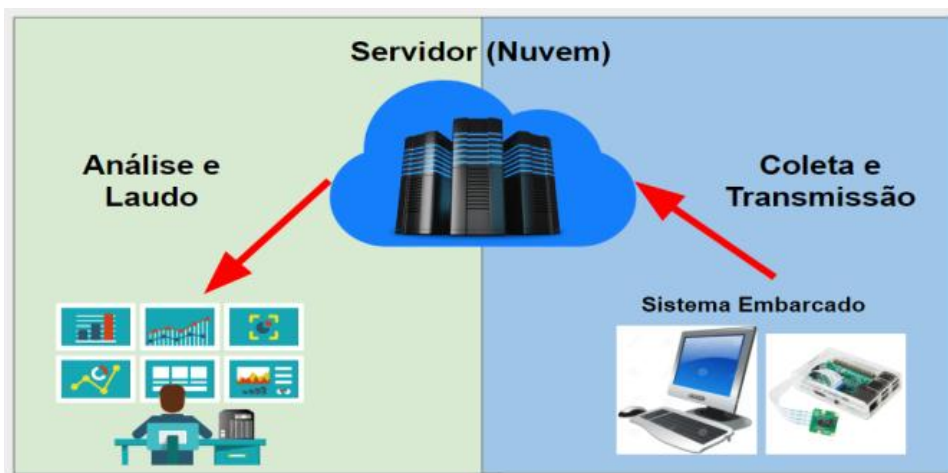
Figura 9: Integração de recursos digitais, lógicos e físicos





O sistema terá um módulo coletor de lâminas que ficará em cada laboratório de coleta. Ele fica conectado a plataforma na nuvem através da rede *internet*. A plataforma, por sua vez mantém o acesso *online* 24 horas por dia, e é responsável pela entrega das imagens das lâminas ao laboratório de análise e laudo. Além da entrega das imagens, a plataforma também disponibiliza ao laboratório de análise e laudo a função de aplicar filtros pré-definidos nas imagens. A Figura 10 mostra a visão geral do projeto.

Figura 10: Visão geral do projeto



O Quadro 13, apresenta as funções do módulo coletor e da plataforma online desenvolvida.

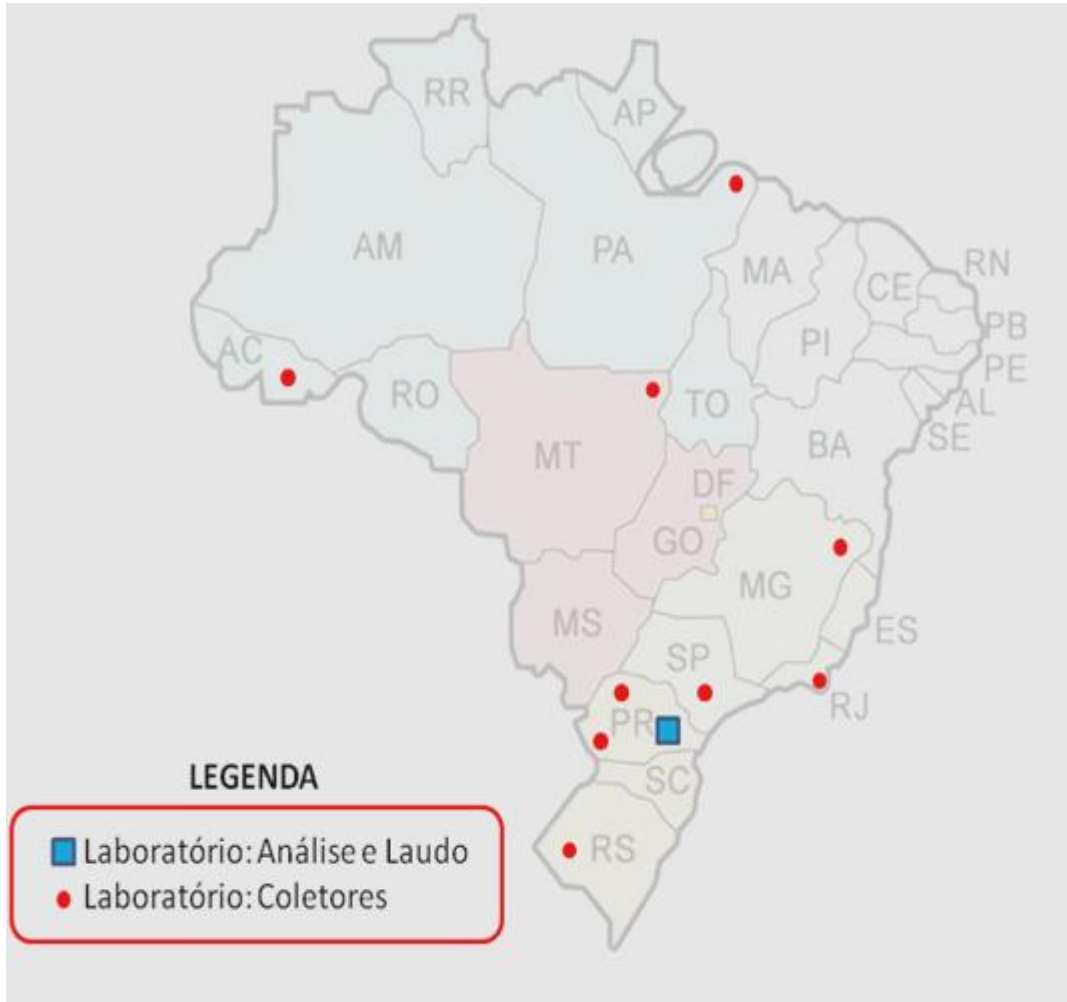
Quadro 13: Funções do Projeto

FUNÇÕES	MÓDULO COLETOR	PLATAFORMA NUVEM
<b>Coleta de Lâminas</b>	<b>X</b>	<b>X</b>
<b>Cadastro de Lâminas</b>	<b>X</b>	<b>X</b>
<b>Filtro das Lâminas</b>		<b>X</b>
<b>Visualização das Lâminas</b>		<b>X</b>

A Figura 11 mostra os laboratórios coletores nos pontos vermelhos e o laboratório de análises e laudos no quadrado azul. Utilizando o sistema coletor, independente das distâncias físicas, pode-se coletar amostras em todo o país, e enviar os dados para um único laboratório de análise. A coleta e envio de forma

automatizada o meio de transporte físico, eliminando o tempo de traslado até os laboratórios.

Figura 11: Distribuição de laboratórios no território brasileiro



## Módulo Coletor

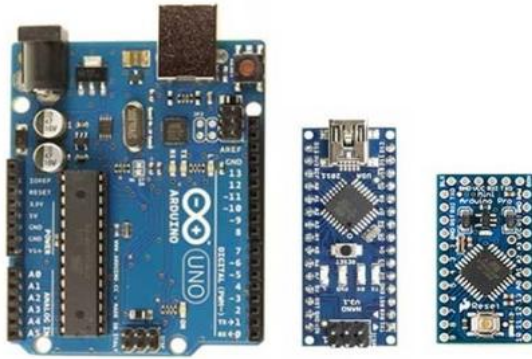
O módulo coletor está dividido em 3 partes:

### I. Hardware

A plataforma microcontrolada Arduino foi utilizado para fazer todo o controle dos motores de corrente contínua do projeto (movimento X e Y da câmera). E para definir o modelo utilizado foram selecionados três modelos como possíveis

candidatos, como mostra no Quadro 14. Na Figura 12 mostra 3 modelos analisados para o projeto.

Figura 12: Modelos de Arduino



Quadro 14: Validação de modelo

ITEM	UNO	NANO	PRO MINI
Tamanho		X	X
Suporte a <i>Shield</i> W5100	X		
Suporte ao módulo ENC28J60	X	X	X
Alimentação USB	X	X	

Por fim, foi escolhido o Arduino UNO por ter o suporte a *Shield* W5100, sem precisar utilizar módulo e chicotes de fios para conexão. Isto foi para evitar a geração de ruídos e até mesmo mal contato entre os fios e o microcontrolador com o decorrer do tempo de uso.

## II. Middleware

A controladora Raspberry PI atua como *middleware*, servindo de ponte entre o *hardware* e o *software*. Na Figura 13, é demonstrado o modelo de Raspberry PI 3 que será utilizado no projeto.

Figura 13: Modelo Raspberry PI 3



Fonte: (RASPBERYPi, 2017)

A conexão *hardware* e a placa microcontrolada Arduino acontece através dos dados enviados pela rede ponto a ponto criada. A Raspberry PI envia os comandos para o Arduino *Web Server* através de requisições POST;

A conexão entre a controladora e *software* ocorre como um microcomputador. Este conjunto utiliza o navegador do sistema operacional para acessar a aplicação local e fazer a coleta de imagens. O sistema de coleta roda em um servidor Node.JS, instalado na Raspberry, que por sua vez se conecta ao sistema de entrega das amostras hospedado e rodando na nuvem.

### III. Software

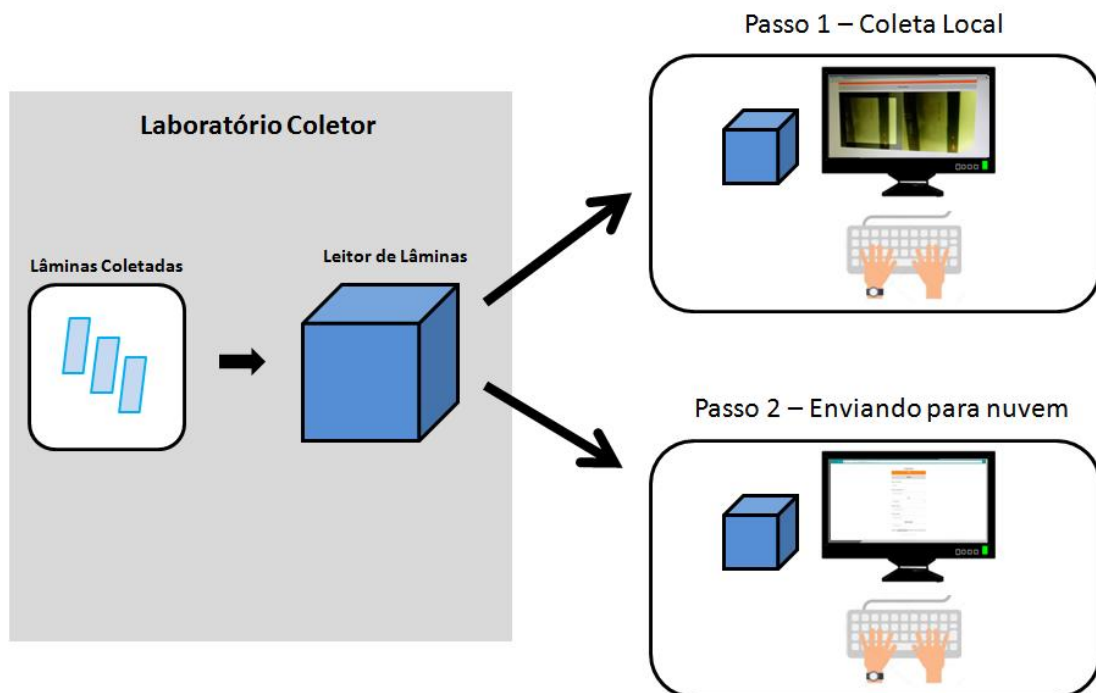
A solução prevê uma aplicação Javascript local com a utilização do protocolo HTTPS através do navegador local. Ela utilizará a câmera de 5 Megapixels da Raspberry PI como recurso nativo do usuário para coletar as imagens das lâminas citológicas, substituindo a função do microscópio eletrônico. Uma vez que as imagens estejam coletadas localmente, elas são enviadas pelo coletor ao sistema online do banco de imagens com os dados pertinentes do paciente, tais como nome, idade, sexo, tipo de exame, entre outros. A Figura 14 mostra o módulo RaspCAM PI de 5 Megapixels utilizado no projeto.

Figura 14: Câmera de 5 Megapixels Raspberry PI



A Figura 15, é um diagrama esquemático de como o módulo coletor é utilizado para coletar as imagens das lâminas físicas. No passo 1 demonstra as lâminas sendo digitalizadas e no passo 2 demonstra as lâminas sendo cadastradas com os dados do exame pelo sistema local, no qual ao finalizar o cadastro enviará para a plataforma na nuvem.

Figura 15: Módulo Coletor



## SISTEMA *ONLINE* DO BANCO DE IMAGENS

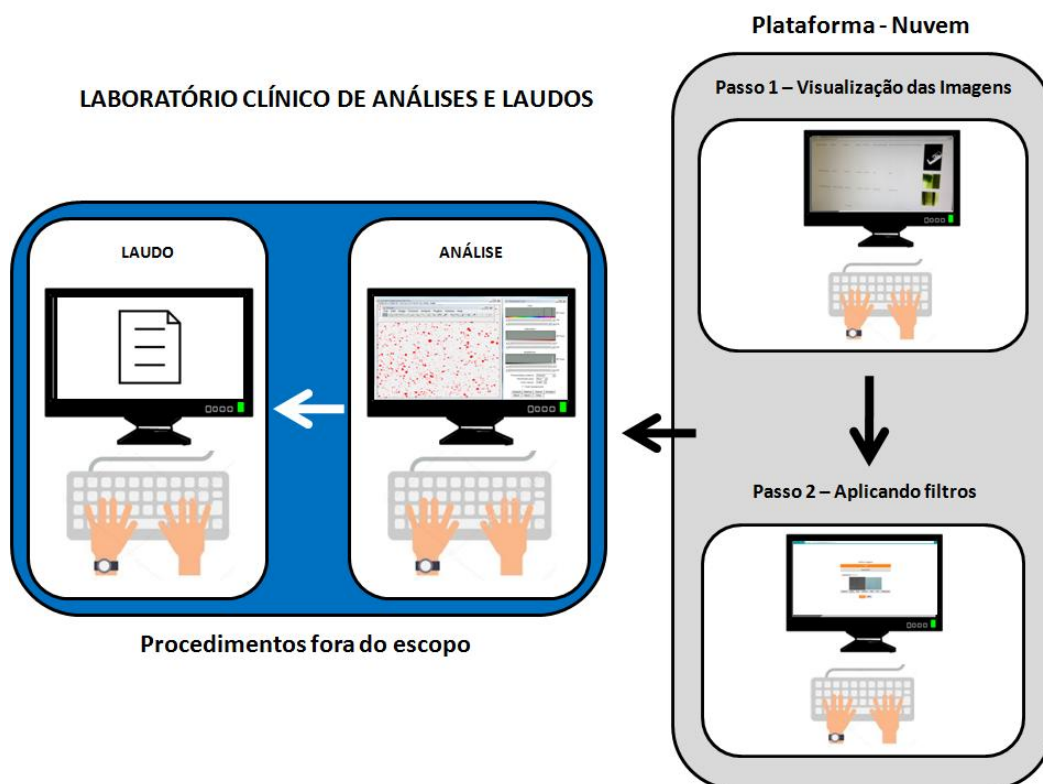
O sistema *online* do banco de imagens foi desenvolvido para armazenar as imagens e os dados dos pacientes de laboratórios que utilizam o módulo coletor.

Após enviar as imagens com os dados para a plataforma *online*, o usuário responsável em manipular as imagens poderá visualizar as imagens em tempo real.

A Figura 16, mostra o banco de imagens na plataforma *online* e os demais procedimentos de rotina dos laboratórios de análises e laudos. No passo 1, demonstra a busca do exame e o passo 2, após escolher o exame, o usuário pode fazer o *download* da imagem. Uma vez que foi feito o *download* da imagem, o especialista do laboratório de análises, prossegue com os procedimentos internos do laboratório em questão e ao final faz o laudo do exame, como já é feito normalmente realizado na rotina.

Observa-se assim, que o objetivo da proposta está na coleta digital da lâmina e a maneira de como disponibilizar essa lâmina digital no laboratório de análises.

Figura 16: Laboratório Clínico de Análises e Laudos



Nos próximos tópicos serão apresentados os recursos e os métodos aplicados no desenvolvimento do projeto, e uma análise detalhada dos ganhos ao processo de coleta e disponibilidade online das imagens para análises.

### 3.1 - Material e métodos

O modelo utilizado no projeto foi a versão mais atual do mercado, Raspberry PI 3. O motivo para usar esse modelo é que já vem com placa de rede *Wi-Fi* integrado, além de possuir um *hardware* robusto com o custo baixo.

A Raspberry PI 3 está designada no módulo coletor como computador embarcado independente possuindo Sistema Operacional baseado no Linux Debian dedicado a Raspberry PI. Seus aplicativos funcionam como em qualquer outro modelo de computador e a conexão com o Arduino atua como uma rede ponto a ponto, onde o Arduino é o servidor web e o cliente o sistema de Coleta Local de Imagens.

O leitor de lâmina foi desenvolvido baseado em linguagens e *frameworks* que existem de mais produtivo no mercado de *softwares* hoje. Utilizando linguagens atuais de alto desempenho, valor agregado, responsivo e *open source*.

É considerado como *software* apenas para lidar na questão abstrata, pois o sistema Leitor de Lâmina que está *online* e está criado com base sólida de uma plataforma *web*.

O Servidor foi a única parte do sistema que foi dividida. Essa divisão aconteceu em duas partes, a primeira no servidor local e a segunda no servidor *web*.

Na parte local quem gerencia a aquisição das imagens/dados e alimenta o banco de imagens da plataforma é o servidor Node.JS + CLI do Firebase (faz o vínculo com o acesso de dados do Firebase). Já no servidor *web*, foi contratado um servidor *online* de hospedagem da empresa Ravehost para hospedar e gerenciar as páginas de forma *online*, além de vincular ao acesso de dados do Firebase. Ainda foi contratado um domínio para vincular o nome do domínio com o servidor *web*. O domínio criado foi [www.leandruino.com](http://www.leandruino.com).

A linguagem adotada no projeto foi a Javascript por ela atualmente na *web* conseguir trabalhar dos dois lados da comunicação, ou seja, tanto do lado do cliente quanto do lado do servidor. Além de ser uma linguagem bem rápida, atual e comum aos desenvolvedores.

A aplicação do Node.JS será no módulo coletor, com o objetivo de ser o servidor local para o *web service* Firebase.

O Google Firebase será responsável por armazenar de forma atualizada em tempo real todas as lâminas coletadas. E na plataforma irá apresentar as imagens de forma organizada junto dos dados para o laboratório de análises e laudos.

Para autenticação foi utilizado o Firebase Authentication. Foram criadas senhas com no mínimo de 6 dígitos na própria plataforma do Google Firebase utilizando o perfil de Administrador de TI. Foi necessário permitir que o domínio faça as requisições ao Firebase na própria plataforma do Firebase. Essas permissões são definidas por regras de acesso a escrita e a leitura dos dados de forma independente.

O Firebase Authentication é utilizado para gerenciar a parte de login e sessões dos usuários, tanto no módulo coletor quanto na plataforma alocada na nuvem.

Para a execução do sistema local, foi necessário preparar a controladora Raspberry PI, para funcionar como um microcomputador.

Foi baixado a imagem do Sistema Operacional RaspBian Jessie com área de trabalho (Desktop) baseado no Debian Stretch, na versão atual do Kernel 4.14, pelo site [https://downloads.raspberrypi.org/raspbian\\_latest](https://downloads.raspberrypi.org/raspbian_latest).

Em seguida foi preparado um cartão SD de 16GB para receber o RaspBian e realizada a instalação através do programa Etcher que foi baixado no site <https://etcher.io/>. Com o cartão instalado foi iniciado o sistema.

Após a inicialização do sistema foi necessário a instalação e habilitação da câmera RaspCam PI. Para isso foi necessário realizar a atualização do software do Sistema Operacional com os comandos dentro do terminal:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Também foi necessário habilitar a câmera dentro da configuração da Raspberry PI, como aparece na Figura 17.



Figura 17: Habilitando a câmera Rasp

```
Setup Options
1 Expand Filesystem           Ensures that all of the SD card s
2 Change User Password        Change password for the default u
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des
4 Internationalisation Options Set up language and regional sett
5 Enable Camera                Enable this Pi to work with the R
6 Add to Rastrack              Add this Pi to the online Raspber
```

Em seguida foram rodados os comandos para testar se estava funcionando a parte da imagem e a parte do vídeo na câmera.

- Imagem:  
**\$ raspistill -o primeira\_foto.png**
- Vídeo:  
**\$ raspivid -o primeiro\_video.h264**

Além do sistema operacional, foi instalado o Node.JS, responsável para fazer o vínculo dos dados coletados com a plataforma *web*. Sem o Node.Js, ficaria inviável criar esse vínculo. A instalação foi realizada a partir do terminal, através dos comandos:

```
$ sudo apt update
$ sudo apt full-upgrade -y
$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
$ sudo apt install -y nodejs
$ node -v
```

Após ter finalizado a instalação do Node.JS, também foi necessário instalar o CLI do Firebase. Dentro do terminal, foi instalado e configurado usando os comandos:

Para instalar o CLI do Firebase:

```
$ npm install -g firebase-tools
```

Depois de instalar, foi necessário utilizar o login da conta do projeto através do comando:

```
$ firebase login
```

Para testar a comunicação, foi utilizado o código:

```
$ firebase list
```

Após instalar o CLI do Firebase, foi necessário criar e definir um diretório do projeto, para que toda vez que rodar o projeto de forma local, já execute direto deste diretório. Para isso foi utilizado o comando:

```
$ firebase init
```

Ao executá-lo, algumas orientações são exibidas durante a configuração do diretório de projetos, inclusive perguntando quais recursos do Firebase você quer usar. No final da etapa, foi necessário escolher um projeto padrão do Firebase.

Uma vez criado o diretório e o mesmo configurado, foi criada uma página teste para verificar se o servidor estava realmente funcionando. Feito isso para colocar o servidor local funcionando, foi necessário entrar no terminal, ir até o diretório do projeto e depois digitar o comando:

```
$ firebase serve
```

Após digitar o comando o servidor passou a executar e para acessá-lo foi necessário abrir o navegador com o endereço localhost:5000. Quando foi requisitado o localhost:5000, o servidor buscou no diretório do projeto a página *index* e executou no navegador rodando com todas as propriedades de vínculo aos projetos online.

Para configurar a rede *wired* no Raspberry PI foi através de um arquivo chamado *interfaces*, localizado na pasta */etc./network*. Esse é um arquivo que somente o usuário *root* pode editar, pois remover uma interface de rede dessa lista provocará a parada de seu funcionamento.

A partir do terminal, foi digitado o seguinte comando para abrir o arquivo:

```
$ sudo nano /etc./network/interfaces
```

Após digitar o comando, foi demonstrado a seguinte informação:

```
iface eth0 inet static --- pressione enter  
(tab) address xxx.xxx.xxx.xxx  
(tab) netmask xxx.xxx.xxx.xxx
```

Para o projeto, a Raspberry PI ficou configurada da seguinte maneira:

```
iface eth0 inet static  
address 200.200.0.1  
netmask 255.255.255.0
```

Como já mencionado, o *hardware* utilizado foi o Arduino UNO R3 com a *shield* W5100 para comunicar com a Raspberry PI com a função de acionar os motores no modo X e Y da câmera PI.

Primeiro foi necessário desenvolver a programação do *firmware* no IDE do Arduino para que ele funcionasse como *web server* e que pudesse receber as requisições do sistema local do módulo coletor (Raspberry PI).

Para colocar o Arduino na rede ponto a ponto foi através das bibliotecas utilizadas pela *shield* W5100, que é responsável pela comunicação via rede *wired* (cabeadada).

Para o projeto, o Arduino ficou configurado da seguinte maneira:

```
Endereço IP:      200.200.0.2  
Máscara de rede: 255.255.255.0  
Gateway:         200.200.0.1
```

No Código 1 parte I e parte II mostram como ficou a configuração da rede *wired* (cabeadada) para o Arduino.

## Código 1: Configuração do Arduino - Parte I

```

1 #include <SPI.h>
2 #include <Ethernet.h>
3 String readString;
4
5 int m1b = 5;
6 int m1a = 6;
7 int m2b = 7;
8 int m2a = 8;
9
10 //Informacoes de endereco IP, gateway, mascara de rede
11 byte mac[] = { 0xB2, 0x16, 0x75, 0xD1, 0x78, 0x4F };
12 byte ip[] = { 200, 200, 0, 2 };
13 byte gateway[] = { 200, 200, 0, 1 };
14 byte subnet[] = { 255, 255, 255, 0 };
15
16 EthernetServer server(80);

```

## Código 1: Configuração do Arduino - Parte II

```

18 void setup()
19 {
20   Serial.begin(9600);
21
22   pinMode(m1b, OUTPUT);
23   pinMode(m1a, OUTPUT);
24   pinMode(m2b, OUTPUT);
25   pinMode(m2a, OUTPUT);
26   pinMode(9, OUTPUT);
27   digitalWrite(9, 1);
28
29   //Inicializa Ethernet Shield
30   Ethernet.begin(mac, ip, gateway, subnet);
31   server.begin();
32 }

```

Para abrir o sistema Local no módulo Coletor, é necessário abrir o navegador e digitar o endereço localhost:5000 para rodar com os recursos do Node.JS e do CLI do Firebase.

No sistema Local as funções disponíveis são: capturar as imagem das Lâminas e cadastrar os exames na plataforma *online*.

Na função Capturar, o processo de capturar a imagem da lâmina e criar a imagem digital só foi possível devido ao desenvolvimento de duas funções com recursos de imagem criadas no Javascript.

A primeira função é responsável por capturar a imagem física para digital. E a segunda função é responsável de visualizar a imagem com *zoom* e forma *crop* (recorte). E a forma de chamá-las no Sistema Local é através da tecla < enter > para a primeira e < shift > + C para a segunda.

Ainda existe a terceira função que serve para reiniciar o processo de captura. Para chamar essa função é necessário pressionar a tecla < espaço >.

No sistema local, possui a integração com o Arduino para poder mover a câmera com os motores nas direções horizontal (X) e vertical (Y). E a forma de comandá-los é através de um navegador com 4 posições (cima / baixo / direita / esquerda) como pode ser observado no Quadro 15.

Quadro 15: Requisições pelo navegador

COMANDO	REQUISIÇÃO
↑	<a href=200.200.0.2/?cima target="fantasma">#8679;</a>
↓	<a href=200.200.0.2/?baixo target="fantasma">#8681;</a>
←	<a href=200.200.0.2/?esq target="fantasma">#8678;</a>
⇒	<a href=200.200.0.2/?dir target="fantasma">#8680;</a>

O Arduino funciona como um Web Server. No qual fica na espera do cliente enviar uma requisição. Toda vez que o Sistema Local enviar uma requisição para o IP 200.200.0.2, o Arduino trata como um comando e aciona os motores de acordo com a requisição. O Código 2, demonstra como o Arduino recebe a requisição e interpreta o comando.

## Código 2: Tratando as requisições

```

34 void loop(){
35   EthernetClient client = server.available();
36   if (client) {
37     while (client.connected()) {
38       if (client.available()) {
39         char c = client.read();
40         if (readString.length() < 100) {
41           readString += c;
42         }
43         if (c == '\n'){
44           if (readString.indexOf("?cima") > 0) {
45             digitalWrite(m1a, 1); digitalWrite(m1b, 0);
46             digitalWrite(m2a, 0); digitalWrite(m2b, 0);
47             delay(500); digitalWrite(m1a, 0);
48           }
49           if (readString.indexOf("?baixo") > 0) {
50             digitalWrite(m1a, 0); digitalWrite(m1b, 1);
51             digitalWrite(m2a, 0); digitalWrite(m2b, 0);
52             delay(500); digitalWrite(m1b, 0);
53           }
54           if (readString.indexOf("?esq") > 0) {
55             digitalWrite(m1a, 0); digitalWrite(m1b, 0);
56             digitalWrite(m2a, 1); digitalWrite(m2b, 0);
57             delay(500); digitalWrite(m2a, 0);
58           }
59           if (readString.indexOf("?dir") > 0) {
60             digitalWrite(m1a, 0); digitalWrite(m1b, 0);
61             digitalWrite(m2a, 0); digitalWrite(m2b, 1);
62             delay(500); digitalWrite(m2b, 0);
63           }
64           readString = "";
65           delay(1);
66           client.stop();
67         }
68       }
69     }
70   }
71 }
72

```

Na função Cadastrar, serve para enviar todos os dados dos exames e as imagens das lâminas para a plataforma *online* (Nuvem). Para isso é necessário ter acesso ao Firebase Realtime Database, caso contrário não haverá conexão com a Nuvem.

Com a conexão estabelecida, o sistema disponibiliza os campos na forma de texto, um botão para selecionar a imagem e uma barra de progresso.

Quando os dados são inseridos, e a imagem é selecionada, o sistema roda a função de inserção no banco de imagens (*storage*) e logo após a imagem ser inserida, é chamada a função de inserção dos dados com o caminho do banco de imagens (*storage*) onde a imagem foi inserida no banco de dados em tempo real (Firebase Realtime Database).

Após todos os procedimentos, se a conexão foi estabelecida, a imagem inserida e os dados cadastrados, o sistema dispara uma mensagem de cadastro completo.

Para a plataforma online foi comprado um domínio [www.leandruino.com](http://www.leandruino.com), e alugado uma hospedagem para disponibilizar todo o sistema de visualização das imagens.

Para acessar a plataforma online, é necessário abrir um navegador (browser) e digitar o endereço: < <https://www.leandruino.com> >. É importante utilizar o protocolo HTTPS no lugar do HTTP, pois o protocolo HTTPS consegue habilitar o uso dos recursos locais do computador de forma direta.

A plataforma possui as funções de visualização das imagens e aplica filtros de imagem, além de ter as funções do módulo coletor, porém para conseguir capturar alguma lâmina, seria necessário o uso de um microscópio digital.

Para entrar na plataforma online é necessário fazer o *login* de acesso, com e-mail e senha.

Na plataforma *online*, as funções disponíveis são: exibir banco de imagens, aplicar filtros, e também repete as funções locais de capturar as imagem das Lâminas e cadastrar os exames na plataforma *online*.

A função Exibir, é a parte principal da plataforma, pois estão dispostos nessa função todos os dados e imagens que o(s) módulo(s) coletor(es) enviaram.

A função aplicar filtro utiliza a biblioteca *Lena.JS* para aplicar os efeitos nas imagens em cima do elemento *canvas*. Com isso o navegador utilizado exige executar com HTML 5, pois o elemento *canvas* é específico do HTML 5.

Quando a imagem é selecionada para receber a aplicação do filtro, a plataforma duplica a imagem com o elemento *canvas*, e roda a função do filtro específico. Dessa forma fica na tela a imagem original e a imagem com o filtro aplicado.

### **3.2 - Testes e validações específicos**

#### **TESTES DE PRÉ-CRIAÇÃO DO MÓDULO COLETOR**

Antes de instalar todos os componentes na estrutura do módulo coletor, primeiro foi necessário testar individualmente cada componente, como demonstra o Quadro 16.

Observação:

Antes de chegar nesse passo do projeto com todos esses testes, o modelo inicial com testes de motores levaram mais de 6 meses para atingir o ideal proposto, deixando quase inviável a aplicação da ideia no projeto.

Quadro 16: Testes primários dos componentes

Nº	TESTE	STATUS	OBSERVAÇÃO
1	Instalação Sistema Operacional Raspberry PI	OK	Funcionando SO.
2	Instalação da RaspCAM PI	OK	Funcionando com a imagem nítida sem zoom.
3	Instalação do Node.JS	OK	Instalado e executando de forma correta.
4	Instalação do CLI Firebase	OK	Instalado e rodando no endereço localhost:5000
5	Arduino + W5100 no Raspberry	OK	Conexão estabelecida entre o <i>Hardware</i> e <i>Software</i> .
6	Arduino + Motores CC com redução	OK	Motores funcionando em modo X e Y através das requisições do navegador

## TESTES PARA CAPTURAR AS IMAGENS

Na parte da captura de imagens, o maior problema foi dentro da Raspberry PI, rodar o navegador Chromium, ele precisa rodar um código antes de iniciar a captura de imagens da câmera PI.

Enquanto não foi descoberto esse problema, ocorria o erro de que não poderia exibir as imagens. Para realizar a aquisição das imagens antes de abrir o sistema no navegador, executar o comando no terminal:

```
$sudo modprobe bcm2835 -v4l2 gst_v4l2src_is_broken=1
```

Esse comando possui a função de habilitar a câmera da Raspberry PI no navegador do Chromium. Sem esse código o sistema não consegue encontrar a câmera em questão. Depois de utilizar o código, acabaram os problemas de imagem.



## TESTES DE ESTRUTURA - MOTORES

Os testes dos motores para a estrutura foram essenciais para saber como funcionaria o conjunto X e Y. Porém o projeto ficou atrasado durante 6 meses para solucionar o problema mecânico, de relação peso x torque dos motores = velocidade aceitável de controle com custo baixo. A solução encontrada para o protótipo do módulo foi utilizar 2 conjuntos de motores de DVD *Player* com redução e *kit* de engrenagens já definidas no conjunto. No Quadro 17 demonstra todos os testes feitos de estrutura e motores.

Quadro 17: Testes e validações - Estrutura / Motores

Nº	Teste	Status	Observação
1	Ligar motor de passo sozinho	OK	Nenhuma.
2	Ligar motor de passo no formato X e Y	Não OK	Não possui força para subir a estrutura
3	Ligar motor de passo no formato X e Y	Não OK	Modificado a estrutura porém continua sem força para subir a estrutura
4	Ligar motor de corrente contínua com <i>driver</i> ponte H	OK	Motor lento, porém funcionando perfeitamente.
5	Ligar motor de corrente contínua com <i>driver</i> ponte H na guia (parafuso de rosca sem fim)	OK	Funcionou, porém ficou lento demais, demorando mais de 5 minutos para subir toda a guia, fora que a base ficou muito pesada.
6	Ligar motor de corrente contínua com <i>driver</i> (posição Y) + motor de passo (posição X)	OK	Funcionou, porém o tempo de resposta do sistema X e Y está muito lento, precisa ser corrigido para otimizar.
7	Testar novo motor CC, leitor de dvd com redução	OK	Nenhuma.
8	Testar a estrutura com dois motores novos CC, posição X e Y	OK	Funcionou bem, roda leve e rápido

## TESTE DE ARMAZENAMENTO DAS IMAGENS - *ONLINE*

Os testes para armazenamento inicialmente eram simples, utilizando a linguagem PHP resolveria enviando de forma FTP, porém como o projeto tinha o propósito de ser todo numa linha mais moderna utilizando somente linguagem Javascript para se tornar mais dinâmico, leve e rápido, acabou se tornando um

desafio a ser vencido. No Quadro 18 apresenta todos os testes de armazenamento das imagens *online*.

Quadro 18: Testes e validações - Armazenamento das imagens - online

Nº	Teste	Status	Observação
1	Teste de envio com PHP para um servidor FTP	OK	Funcionou sem problemas de comunicação.
2	Teste de envio com Javascript	Não OK	Erro na hora de enviar o arquivo para o servidor.
3	Teste de envio com Javascript	Não OK	Foi alterado o código e na hora de executar está com erro, informando que enviou o arquivo, porém não enviou nada.
4	Teste de envio com Javascript/Jquery	OK	Envia no formato FTP, porém ele atualiza todas as informações do FTP, todas as imagens que foram excluídas no servidor voltam devido a permanecer no banco local.
<b>Migração para o Firebase</b>			
5	Teste de envio de texto para o Firebase	OK	-
6	Teste de envio de imagem para o Firebase	Não OK	Não estava definido o caminho do Storage no Firebase, faltava habilitar no código.
7	Teste de envio de imagem para o Firebase	OK	Faltava o caminho do Storage.

### TESTE DE VISUALIZAÇÃO DAS IMAGENS - ONLINE

Inicialmente a visualização das imagens seria através do servidor com acesso via FTP, onde o usuário enviaria as imagens e dados por pastas e atualizaria o servidor. Porém, no decorrer do projeto foi visto que não seria interessante essa forma de visualização. Foi analisado em utilizar a linguagem PHP de volta para criar um banco de dados, porém estaria fora do contexto do projeto que seria aplicar uma linguagem leve, rápida e dinâmica. Com isso chegou a conclusão de migrar todo o conceito para o Firebase, que mantém a linha do projeto que é a utilização plena do Javascript. No Quadro 19 mostra os testes sobre os processos para conseguir demonstrar as imagens na plataforma *online*.

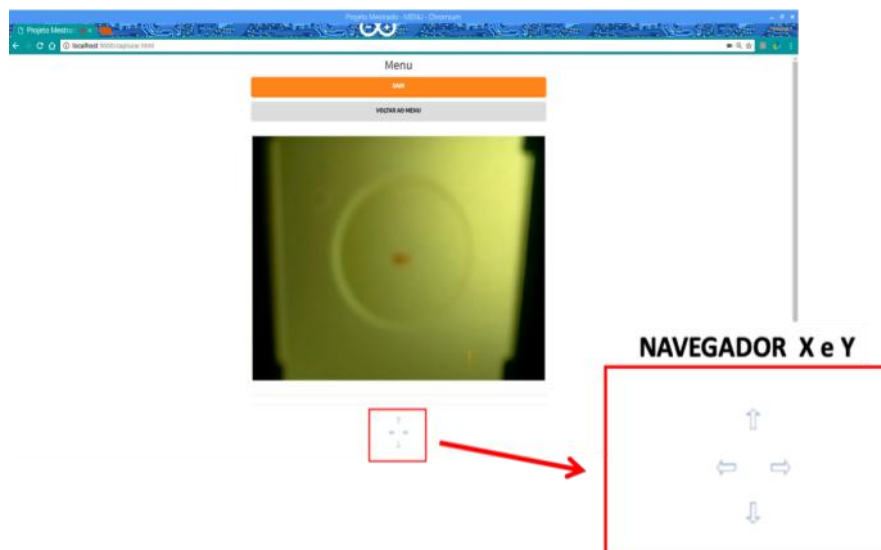
Quadro 19: Testes e validações - Visualização das imagens - online

Nº	Teste	Status	Observação
1	Visualização da imagem	Não OK	Não estava aparecendo nada na tabela de exibição.
2	Visualização da imagem	Não OK	Começou a aparecer, porém estava aparecendo texto no lugar do código estar sendo executado
3	Visualização da imagem	Não OK	Foi verificado que o código estava colocando aspas simples no código o tornando texto, porém ao corrigir voltou a não aparecer nada na tabela de exibição.
4	Visualização da imagem	Não OK	A forma de capturar a URL com o <i>token</i> de autenticado não está passando da forma correta.
5	Visualização da imagem	OK	Corrigido a forma de buscar a URL

#### TESTE DO SISTEMA LOCAL E DA PLATAFORMA ONLINE

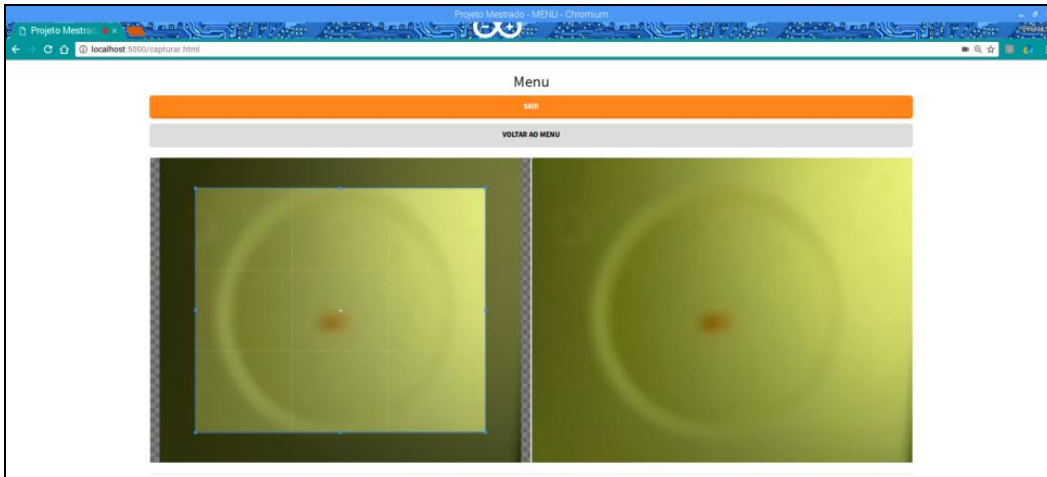
No sistema local as funções disponíveis são de capturar e cadastrar. As Figuras 18 e 19, mostram a tela de captura. Já na Figura 20 mostra a tela de cadastro da imagem da lâmina. A Figura 18 mostra a imagem sendo visualizada e também o menu direcional para mover a câmera nos sentidos X e Y do módulo coletor local.

Figura 18: Capturar Imagem



A Figura 19 mostra no processo da captura da lâmina as ações de *crop* (recorte) e *zoom* da imagem.

Figura 19: Capturar Imagem com as ações *Zoom* e *Crop*



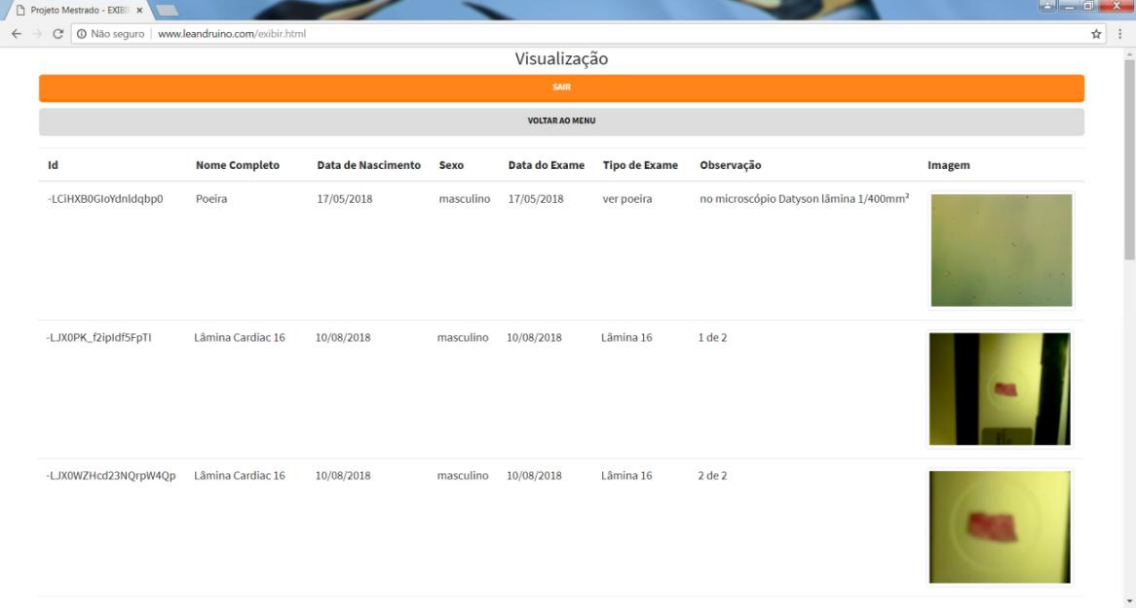
A Figura 20 mostra a tela de cadastro de novas imagens com seus dados, ao selecionar a imagem ela já anexa todo o pacote e envia para plataforma *online*.




Figura 20: Cadastrar Imagem

A plataforma *online*, possui quatro funções, sendo duas exclusivas da plataforma: exibir o banco de imagens e aplicar filtros nas imagens, e ainda as outras duas funções que o módulo coletor possui: Coletar e Cadastrar Imagens.

A Figura 21 mostra o banco de imagens com três coletas, sendo duas de um mesmo paciente e outra de um segundo paciente da plataforma online.

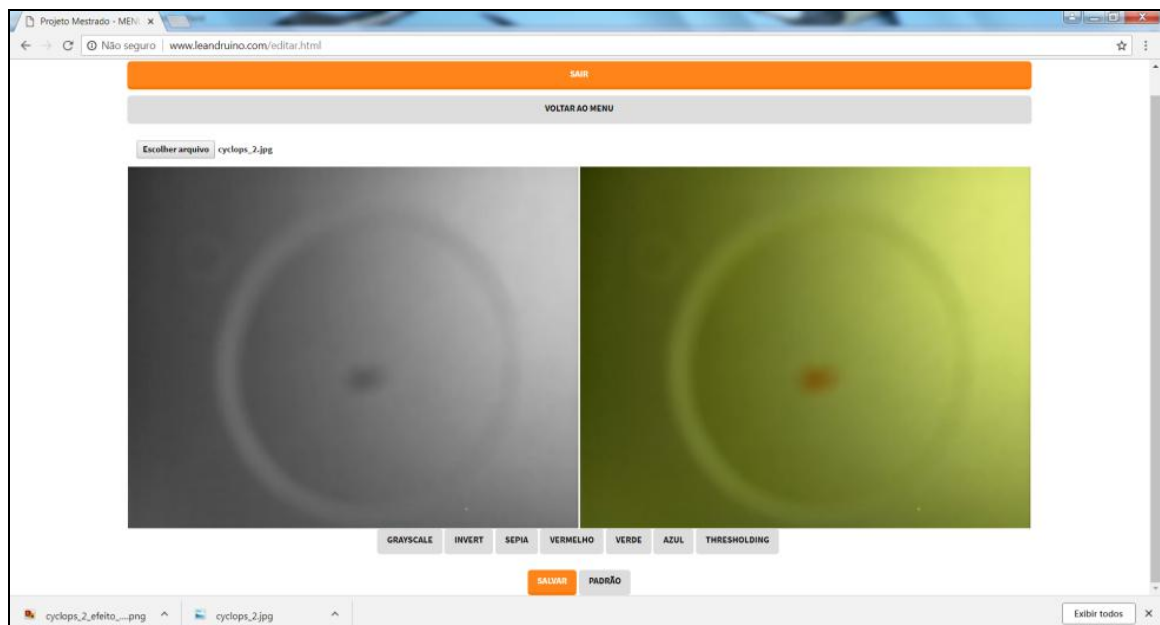
Figura 21: Banco de imagens



Id	Nome Completo	Data de Nascimento	Sexo	Data do Exame	Tipo de Exame	Observação	Imagem
-LCHXB0GtoYdnlqbp0	Poeira	17/05/2018	masculino	17/05/2018	ver poeira	no microscópio Dabysen lâmina 1/400mm <sup>2</sup>	
-LJX0PK_f2lpld5FpT1	Lâmina Cardiac 16	10/08/2018	masculino	10/08/2018	Lâmina 16	1 de 2	
-LJX0WZHcd23NQrpW4Qp	Lâmina Cardiac 16	10/08/2018	masculino	10/08/2018	Lâmina 16	2 de 2	

A Figura 22 mostra a tela com a aplicação do filtro tons de cinza na imagem coletada.

Figura 22: Aplicando o filtro na imagem



### 3.3 - Validação do protótipo

#### CADASTRO MÓDULO COLETOR + PLATAFORMA *ONLINE*

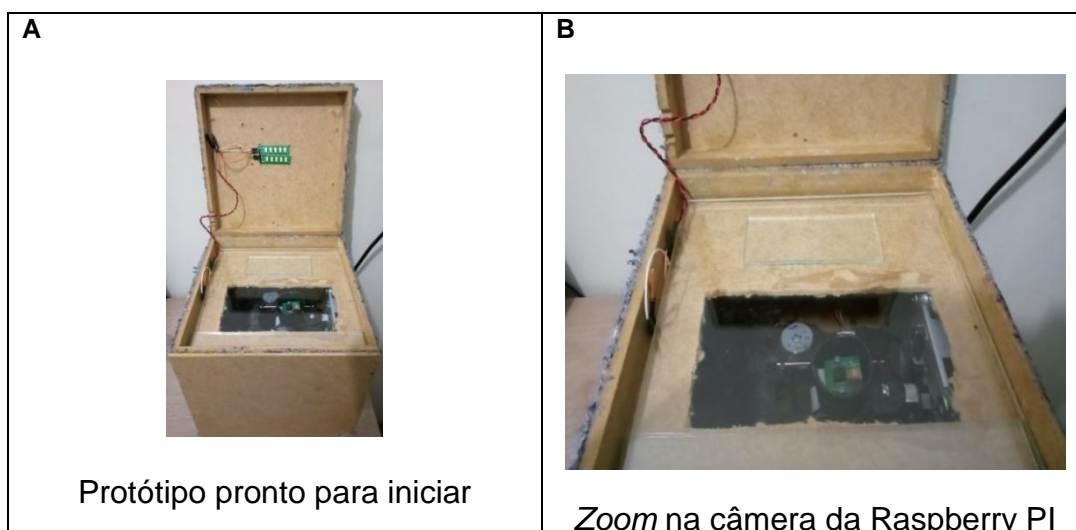
Foi utilizado para teste e validação do protótipo o envio da lâmina capturada para a plataforma *online*, e logo depois o teste na plataforma para verificar se a lâmina estava no banco de imagens.

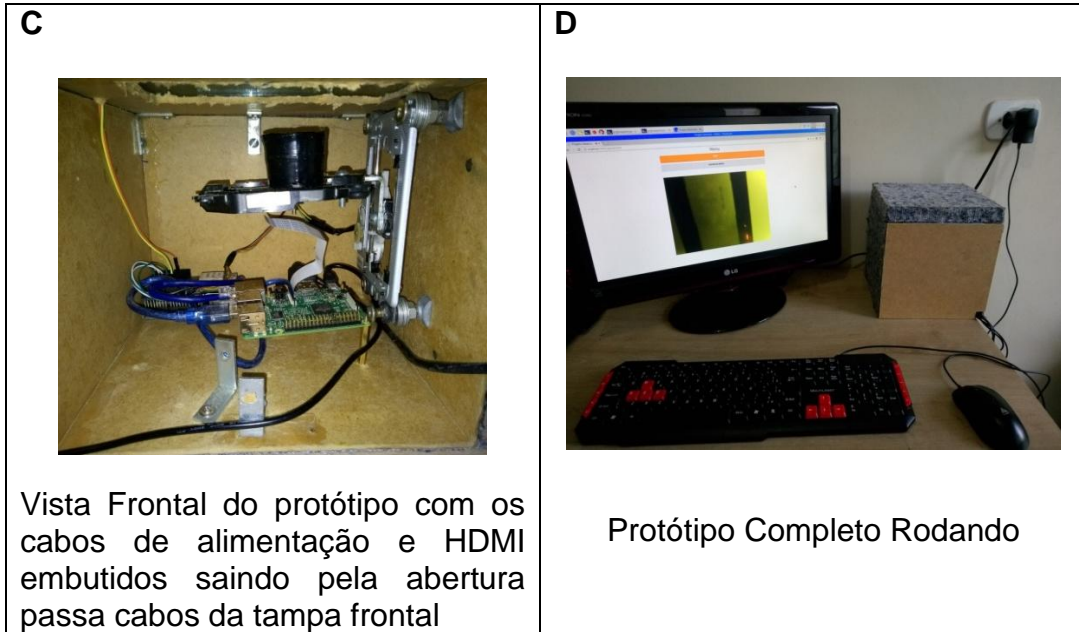
Primeiro será necessário executar o CLI do Firebase junto ao Node.JS. Para isso é necessário entrar no terminal, ir a pasta raiz do sistema e executar o comando:

**\$sudo firebase serve**

Uma vez que o servidor Node.JS já está em execução, é necessário abrir o navegador Chromium e digitar o endereço <https://localhost:5000>, onde irá executar direto o sistema para Capturar e Cadastrar na plataforma. É importante frisar que é preciso alterar o protocolo de HTTP para o de segurança HTTPS, dessa forma é possível habilitar o uso dos recursos do computador cliente. E assim, o sistema irá reconhecer o recurso de vídeo e poderá ser utilizado sem pedir nenhum certificado extra. No Quadro 20, onde demonstram 4 imagens do protótipo.

Quadro 20: Processos do módulo coletor - Protótipo



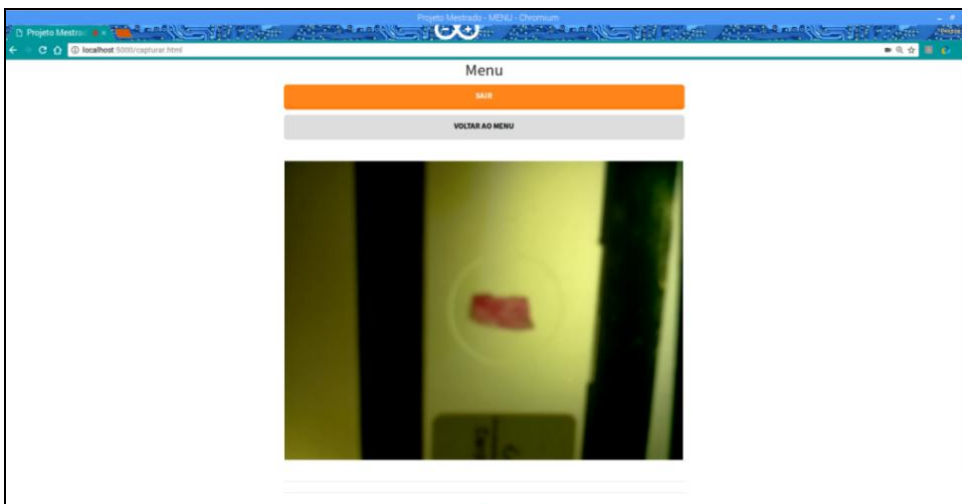


## CAPTURANDO PELO MÓDULO COLETOR

### *Etapa I - Capturando a imagem*

Nessa etapa foi realizado o teste de capturar direto pela plataforma sem utilizar o módulo. A única diferença que pela plataforma não possui o navegador dos motores e exige que no local onde está sendo feita a captura exista o recurso de coleta de imagem, como um microscópio digital. Na Figura 23 pode demonstrar a coleta de uma lâmina sem os recursos do módulo.

Figura 23: Capturando Imagem




Feito a captura, agora é necessário cadastrar essa imagem no banco de imagens para poder ser acessado através da plataforma *online*.

### *Etapa II - Cadastrando os dados*

Da mesma maneira como foi feita a captura pela plataforma *online*, também existe o recurso de cadastrar novos exames a partir da plataforma. Na Figura 24, pode ser visualizado a mesma tela de cadastro do módulo também se encontra na plataforma *online*.

Figura 24: Tela de Cadastro



A imagem mostra a interface de usuário para o cadastro de exames. No topo, há um botão laranja com o texto "SIR" e um botão cinza com o texto "VOLTAR". Abaixo, há campos de entrada para "Nome Completo" (com o placeholder "Nome"), "Data de Nascimento" (com o placeholder "Ex: dd/mm/aaaa"), "Sexo" (com o menu suspenso selecionado "Masculino"), "Data do Exame" (com o placeholder "Ex: dd/mm/aaaa"), "Tipo de Exame" (com o placeholder "Tipo de exame") e "Observações" (com o placeholder "Observações"). Na base, há uma seção para upload de imagem com o texto "Imagem" e um botão "Escolher arquivo", além do status "Nenhum arquivo selecionado".

Uma vez realizado o teste de captura e cadastro, agora o próximo passo é validar se na plataforma *online* a imagem está disponível.

## VISUALIZANDO PELA PLATAFORMA ONLINE

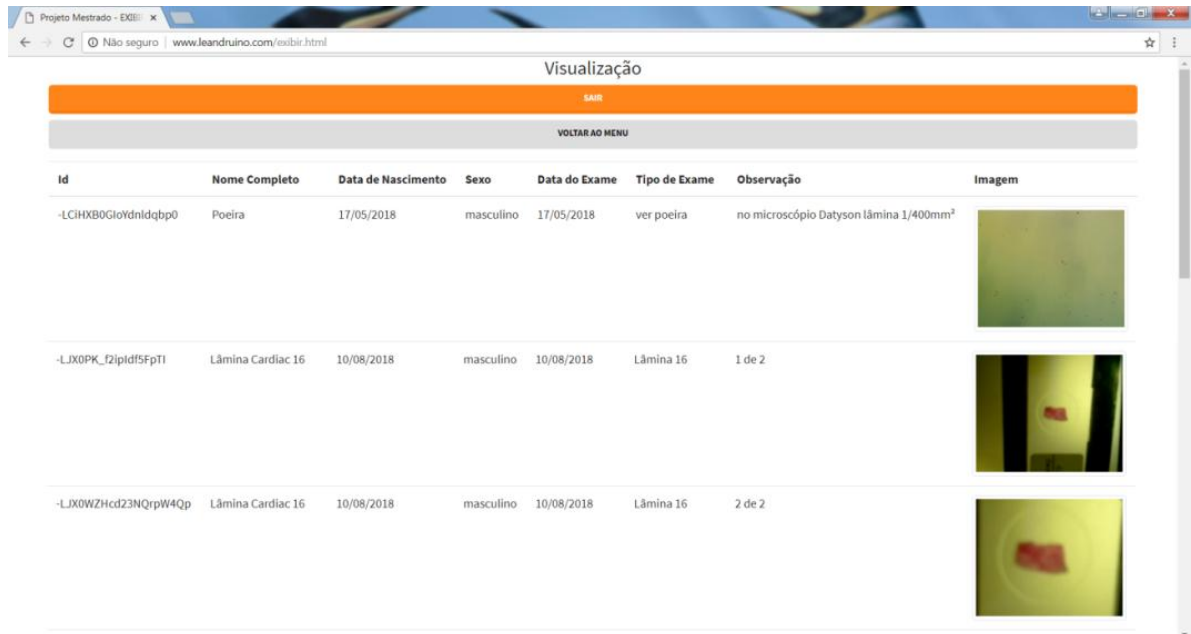
### *Etapa III - Na plataforma online*

A parte principal da plataforma é o local de visualização das imagens. Nessa parte fica o acesso de todas as imagens das lâminas com seus respectivos dados



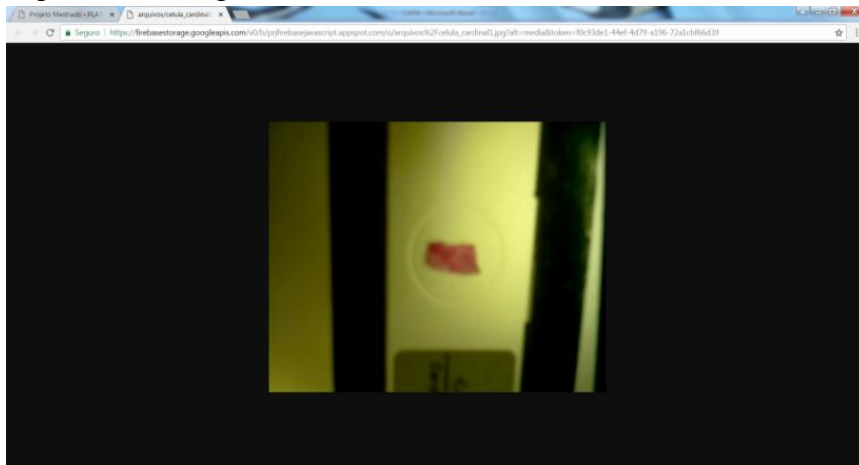
que os módulos coletores enviam. Na Figura 25 mostra a tela com as imagens das lâminas e seus dados.

Figura 25: Imagem já disponível para visualização na Plataforma *online*



Uma vez a imagem disponível, clicar sobre a imagem com o botão principal do *mouse* que ela abrirá e estará disponível para ser feito o *download* e começar a trabalhar no exame. A Figura 26 demonstra a imagem ampliada.

Figura 26: Imagem visualizada



Como resultado obtido, percebe que a plataforma está estável, sem problemas de comunicação e sincronismo com o módulo coletor. Conseguiu realizar a captura da imagem, passou pelo cadastro dos dados para a plataforma *online*, e por último demonstrou a imagem em tempo real que já estava no banco de imagens.

## CAPÍTULO 4

### DISCUSSÃO

De acordo com Ramos, Ralha e Teodoro (2016), a área de saúde e, em especial, a patologia clínica com análise automatizada de imagens microscópicas é um campo com demanda crescente por poder computacional. Uma barreira para a utilização eficaz de computação nesse cenário é a indisponibilidade de recursos computacionais suficientes, devido aos altos custos relacionados. Dessa forma, foi avaliado a utilização de uma arquitetura de baixo custo e gasto energético formada por equipamentos Raspberry PI para utilização em aplicações médicas em análise de imagens microscópicas.

E Neto (2014) ainda diz que pesquisas em computação ubíqua estão sendo realizadas em tópicos como: acesso básico a qualquer tipo de dispositivo sem fio, suporte à mobilidade na rede de forma transparente, segurança, tratamento de contexto, uso eficiente de energia, etc. O Arduino e o Raspberry PI são plataformas amplamente utilizadas por profissionais que têm interesse no campo das aplicações ubíquas. Tanto Arduino quanto o Raspberry PI permitem o desenvolvimento de diversos tipos de projetos.

No projeto a Raspberry PI funcionou como computador embarcado, funcionando com um sistema operacional, além de baratear os custos do projeto, deu independência para o sistema, devido a não precisar de um computador dedicado para rodar. Para o módulo as dimensões da Raspberry PI, facilitaram a acomodação física e fez com que módulo coletor se tornou funcional e com portatibilidade.

Ainda explicando sobre a utilização dos *hardwares* do projeto, ao utilizar a plataforma microcontrolada Arduino como *web server*, facilitou muito a interação do sistema rodando no navegador da Raspberry com os motores. Como o sistema local está executando todo em HTML com Javascript, colocar uma função com o próprio HTML, tornou muito mais amigável a forma de interação dos *Hardwares* com o Sistema.

Como não houve verba destinada para o projeto, o protótipo desenvolvido foi construído com recursos próprios. Tal situação resultou no fato de algumas peças utilizadas possuíam qualidade inferior ao idealizado inicialmente, entretanto as funções continuaram semelhantes ao das peças ideais.

De acordo com Machado e Souki (2004), um dos grandes benefícios que a evolução tecnológica proporcionou à Ortodontia foi a utilização das imagens digitais. O uso dessa nova tecnologia em Odontologia, e especificamente em Ortodontia, aliada à introdução da Fotografia Digital, permite aos profissionais desta área utilizar recursos, antes inimagináveis, facilitando a elaboração do diagnóstico ortodôntico. (MACHADO, SOUKI, 2004)

Pereira e Regis (2015) ainda dizem que foi feito o desenvolvimento de um software para determinação do tipo sanguíneo em humanos, nos sistemas ABO e Rh, usando técnicas de processamento digital de imagens. Utilizou-se como referência o teste manual em lâmina para a análise do tipo sanguíneo das amostras. Os resultados dessa análise foram fotografados por dois dispositivos diferentes, produzindo imagens com resoluções de 640 x 360 pixels e 4320 x 3240 pixels. Em seguida, as imagens obtidas foram processadas pela aplicação desenvolvida para detecção das reações de aglutinação, as quais permitem a classificação do tipo sanguíneo. A utilização do programa permite a minimização do risco de falha humana na interpretação dos resultados da aplicação do teste manual.

No projeto foi utilizada o modelo de 5 Megapixels, porém já existe um modelo mais novo e robusto com 8 Megapixels. Esse modelo mais novo amplia a angulação de aproximadamente 80° para 100° de visão.

A qualidade obtida nas imagens não ficou com uma resolução nítida onde haviam textos com fonte menor que tamanho 8 e imagens menores que o tamanho de 50mm.

Dessa forma a imagem precisa ser ajustada através dos motores X e Y, e esses muito das vezes possuem o passo maior do que o necessário, não conseguindo deixar o foco correto na imagem.

Sobre o movimento dos motores foi na verdade um problema que fez um retardo de evolução do projeto durante aproximadamente uns 6 meses. Com testes no qual conseguia movimentar o eixo X porém quando tentava movimentar o eixo Y não havia relação torque x velocidade.

O conjunto dos motores X e Y, foram utilizados motores de DVD *players* com conjuntos de engrenagens. O ideal seria criar uma relação entre peso e avanço em X, além de peso e avanço em Y.

Para Luz e Araújo (2016), entre as tecnologias aplicadas à vigilância em saúde, o mapeamento de epidemias se destaca pela importância em detectar possíveis pontos de surtos de uma determinada doença e com isso possibilitar uma alocação de recursos mais eficiente para aquela localidade. Neste sentido, a proposta foi o desenvolvimento de uma plataforma Web que possa detectar casos suspeitos de doenças da Tríplice Epidemia e possíveis focos de proliferação do mosquito *Aedes Aegypti*. Para tanto, foi desenvolvido o PortalVCS com o uso de conceitos como o design responsivo e tecnologias como Web Services, JavaScript e Firebase.

No projeto o maior problema na nuvem foi associar o Google Firebase com página desenvolvida para a visualização das imagens na plataforma *online*, o problema estava na parte em que era para relacionar as imagens com o caminho de onde elas seriam salvas e ainda entender que esse caminho não é o mesmo caminho onde serão disponibilizadas para visualização e *download* nos navegadores.

A primeira parte foi solucionada usando o próprio Javascript, porém aplicando funções diferentes. Dessa forma era possível construir o caminho externo como *hiperlink*, porém ainda não era possível utilizar os *links* das imagens salvas no *storage* do Firebase.

A segunda parte foi solucionada depois de ser analisada a maneira de como a imagem era apresentada dentro do console do Firebase. Para entregar o caminho de uma imagem salva no *storage* do Firebase é necessário fazer o *login* e passar o *token* de acesso, esse que é uma chave criada no momento em que o usuário salva a imagem no *storage*. Feito isso é possível buscar o caminho da imagem salva no *storage* de forma que o usuário consiga visualizar e fazer o *download*.

Com o sistema rodando na linguagem HTML com Javascript, sendo que o Javascript rodando tanto no lado do cliente como no lado do servidor, tornou o sistema multiplataforma, rodando de forma rápida em qualquer plataforma e ainda com qualquer Sistema Operacional, sendo ele para pc, *tablet* ou qualquer outro tipo de *mobile*.

Por ser um sistema responsivo (bootstrap) e com visual estilizado (CSS), o sistema local e a plataforma online se tornaram muito mais limpo do que um sistema convencional.

Sistema definido como solução para armazenar os dados e as imagens, esse sistema atualmente é a plataforma que o próprio Google usa para armazenar suas tecnologias. A vantagem de utilizar essa plataforma que o crescimento do banco de dados em tempo real, acontece também em tempo real, ou seja, o crescimento e população do banco de dados dentro do Firebase acontece de acordo com a demanda necessária e em tempo real.

A maneira como ficou a plataforma online, mesmo que ainda simples, podendo passar por melhorias, ela consegue rodar em qualquer ambiente, tem a possibilidade de utilizar os recursos dos clientes através do protocolo HTTPS, e pode ser adaptada de forma rápida sem muitas mudanças para suportar outros fins sobre o tema de imagem.

A plataforma *online* ainda possui a vantagem de trabalhar com recursos do usuário, ou seja, caso o usuário possua um microscópio digital ele poderá conectar na plataforma e adquirir as imagens a partir dela.

Não foram feitos testes em laboratório Clínico, com exames de pacientes. Os testes que foram feitos ocorreram no laboratório de desenvolvimento, e foram com lâminas citológicas adquiridas por *kits* prontos. Devido a não precisar passar por liberação do comitê de ética.

## CAPÍTULO 5 CONCLUSÃO

O projeto "Leitor óptico de lâminas citológicas com armazenamento no banco de imagens online para agilizar o processo de visualização nos laboratórios de análises" alcançou todos os objetivos. Nos Quadros de 21 a 25, estão demonstrados como cada objetivo alcançou a validação.

Quadro 21: Objetivos Geral e Específicos I

Objetivo	Esperado
<b>Geral</b>	Desenvolver um sistema de leitura de lâminas citológicas com armazenamento feito no banco de imagens <i>online</i> para agilizar o processo de análise nos laboratórios clínicos.
Solução Realizada	
Foi desenvolvido um sistema com uma plataforma <i>online</i> para receber as imagens de um ou mais laboratórios em tempo real com as lâminas e seus dados.	
Resultado obtido	
Não será mais necessário fazer o transporte, diminuindo o tempo gasto e eliminando todos os possíveis erros de procedimentos do transporte.	

Quadro 22: Objetivos Geral e Específicos II

Objetivo	Esperado
<b>Forma de armazenamento</b>	No servidor em banco de dados em tempo real, com possibilidade de acesso remoto simultâneo desde que o usuário tenha direito ao acesso.
Solução Realizada	
Toda a plataforma online roda em função do Google Firebase, com ele foi possível tornar o acesso multiusuário, e com atualização dos dados em tempo real.	
Resultado obtido	
O armazenamento das imagens atingiu seu objetivo, visto que toda imagem com os dados que chegam na plataforma, é atualizada para todos os usuários e com a vantagem de ser em tempo real.	

Quadro 23: Objetivos Geral e Específicos III

Objetivo	Esperado
<b>Segurança na informação</b>	Para que as imagens sejam restritas aos laboratórios clínicos.
<b>Solução Realizada</b>	
Toda nova inserção que ocorre na plataforma, sendo os dados e as imagens que chegam são armazenadas com segurança no banco de dados em tempo real (Firebase Realtime Database) e também no banco de imagens ( <i>storage</i> ). Para acessar qualquer um dos dois, é necessário <i>fazer login</i> na plataforma, caso contrário não são disponibilizados os dados para o usuário.	
<b>Resultado obtido</b>	
A parte da segurança na informação, atingiu o objetivo já que para conseguir fazer acesso a qualquer imagem é necessário ter uma conta vinculada ao sistema e para visualizar qualquer imagem ou dado, obrigatoriamente é preciso ter feito <i>login</i> no sistema para criar a sessão com o <i>token</i> de permissão gerado pelo Google Authentication.	

Quadro 24: Objetivos Geral e Específicos IV

Objetivo	Esperado
<b>Portatibilidade</b>	Desenvolver um produto no qual possa ser de simples manuseio e fácil instalação.
<b>Solução Realizada</b>	
O protótipo possui menos de 5 quilogramas, com tamanho 30 cm x 25 cm x 25 cm, tornando fácil o transporte. E a forma da instalação física é um monitor com entrada HDMI, uma tomada elétrica e internet <i>Wi-Fi</i> . Uma vez configurado a <i>internet</i> , o sistema já está operacional.	
<b>Resultado obtido</b>	
Da maneira como ficou desenvolvida a plataforma precisa apenas de um computador com acesso a <i>internet</i> para coletar as imagens e depois serem analisadas. E os módulos coletores precisam apenas serem configurados na <i>internet</i> e ligados a tomada e a um monitor. Sem precisar alocar nenhum recurso a mais. Facilitando a disposição no ambiente devido seu tamanho e facilidade de manusear.	

Quadro 25: Objetivos Geral e Específicos V

<b>Objetivo</b>	<b>Esperado</b>
<b>Tempo de acesso</b>	Velocidade na visualização das lâminas. Devido à eliminação do transporte das lâminas até o local de análise.
<b>Solução Realizada</b>	
Após ser coletada a imagem pelo módulo coletor, o tempo de acesso a lâmina fica associado a velocidade da <i>internet</i> contratada pela clínica de análise e pela clínica de coleta onde está o módulo coletor.	
<b>Resultado obtido</b>	
A visualização cumpriu seu objetivo visto que antes era necessário o transporte da lâmina do laboratório de coleta para o laboratório de análise. Com o sistema implantado, elimina todo esse período e fica apenas dependendo do <i>upload</i> dos dados para a plataforma e o <i>download</i> da página para a visualização na plataforma.	

De acordo com os Quadros 21 ao 25, pode-se dizer que todos os objetivos foram alcançados. Mas deve ser levado em consideração o nível que os objetivos foram alcançados. Ao tratar o projeto como único sem ter nenhum outro projeto similar como base, existem dois níveis a serem analisados, nível de prototipagem e nível de produto comercial.

Como prototipagem novamente pode-se dizer que tudo foi alcançado, porém como nível de produto comercial ainda falta buscar e refinar mais, além de estudar mais cada parte envolvida no projeto. E quem sabe criar novas linhas de pesquisa a partir da prototipagem, uma específica para o módulo coletor e outra específica para plataforma online.

### **5.1 - Considerações finais**

Como considerações finais, pode se dizer que o projeto alcançou o objetivo, conseguindo solucionar e entregar solução para todos objetivos específicos. O projeto é pioneiro no segmento de laboratórios com essa visão de diminuir o tempo de entrega das lâminas e possui potencial para mais estudos, podendo futuramente ser base para novos projetos e linhas de pesquisas.

Tanto o módulo coletor quanto a plataforma *online*, são fáceis de replicar e possuem componentes de simples aquisição.



Quanto aos laboratórios caso queiram alcançar novos horizontes, aplicando a forma proposta no projeto, poderão criar parcerias com novos laboratórios, colocando o módulo coletor em cada laboratório parceiro, e centralizando a unidade de análise para os procedimentos e laudos num único ponto. Dessa forma poderão se tornar especializados em mais exames, concentrando os profissionais especializados nesse ponto, diminuindo nos demais laboratórios, não terão custos de transportes dos exames e o tempo que era gasto no transporte passa não existir.

Como foi demonstrado o protótipo ainda está em fase de melhorias para se tornar um produto, porém os benefícios já colhidos, tornam o projeto proposto válido e com potencial no mercado uma vez que se tornar produto.

## 5.2 - Melhorias futuras

Apresentamos a seguir melhorias no produto desenvolvido para torná-lo mais eficiente:

- Confeccionar por um designer de produtos para chegar no tamanho otimizado, com relação entre resistência e um bom *design*.
- Utilizar uma câmera de maior qualidade no módulo coletor e que esse possa utilizar ferramentas como autofoco, *zoom in/out* e filtros naturais.
- Definir quantas imagens de lâminas são de um mesmo paciente na forma automática para o módulo coletor.
- Acrescentar uma forma de armazenar as imagens de um paciente com histórico na plataforma. Para análises futuras ter toda a evolução ou mesmo ter todos os exames para comparar quando necessário.
- Melhorar as telas dos sistemas, deixando os sistemas mais apresentados aos usuários finais.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABRALAPAC - Associação Brasileira de laboratórios de Anatomia Patológica e Citopatologia. **O que é Citopatologia?** Disponível em: <[http://www.abralapac.org.br/v3/utilidade\\_publica/texto.php?id=10](http://www.abralapac.org.br/v3/utilidade_publica/texto.php?id=10)>. Acesso em 20 maio de 2018.

AGUILAR, Luis Joyanes. **Fundamentos de programação - Algoritmos, estruturas de dados e objetos**. Tradução da 3ª Ed. Editora McGraw-Hill, 2011.

BRITO, Felipe. **Qual a diferença entre zoom óptico e zoom digital?** Disponível em <<https://www.zoom.com.br/camera-digital/deumzoom/qual-a-diferenca-entre-zoom-optico-e-zoom-digital>>, 2014. Acesso em 10 de março de 2017.

BLUM, Jeremy. **Explorando o Arduino - Técnicas e ferramentas para mágicas de engenharia**. 1ª Edição, Rio de Janeiro. Editora Alta Books, 2016.

CARDOSO, André. **Firestore - banco de dados em tempo real**. Disponível em <<https://www.andrebian.com/firebase-banco-de-dados-em-tempo-real/>>, 2018. Acesso em 17 de fevereiro de 2019.

CARDOSO, Marcelo. **Qual a diferença entre zoom óptico e zoom digital?** Disponível em <<http://www.techtudo.com.br/artigos/noticia/2012/07/qual-diferenca-entre-zoom-optico-e-zoom-digital.html>>, 2012. Acesso em 10 de março de 2017.

CARNEIRO, José; JUNQUEIRA, Luiz C. **Biologia Celular e Molecular**. 9ª ed. Editora Guanabara Koogan, 2012.

CCB – UFPE. **Citologia**. Disponível em: <<https://www.ufpe.br/citologia/hp04.htm>> Acesso em 04 março de 2017.

COCIAN, Luis Fernando Espinosa. **Manual da linguagem C**. 1ª Edição, Canoas. Editora ULBRA, 2004.

CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. **Computação Gráfica - Processamento e Análise de Imagens Digitais - Volume 2.** 1ª Edição, 2009. Editora Campus

CORELDRAW. **Entender a profundidade da cor.** Disponível em: <[http://product.corel.com/help/CorelDRAW/540240626/Main/BR/Doc/wwhelp/wwhimpl/common/html/wwhelp.htm?context=CorelDRAW\\_Help&file=CorelDRAW-Understanding-color-depth.html](http://product.corel.com/help/CorelDRAW/540240626/Main/BR/Doc/wwhelp/wwhimpl/common/html/wwhelp.htm?context=CorelDRAW_Help&file=CorelDRAW-Understanding-color-depth.html)>, 2012. Acesso em 10 de março de 2017.

CURVELLO, André. et al. **Linguagens de programação para sistemas embarcados.** Disponível em: <<https://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/>>, 2015. Acesso em 17 de fevereiro de 2019.

DELAI, André Luiz. **Sistemas Embarcados: Computação Invisível.** Disponível em: <<http://www.hardware.com.br/artigos/sistemas-embarcados-computacao-invisivel/conceito.html>>, 2013. Acesso em 04 março de 2017.

DIAGNÓSTICOS DO BRASIL. **Anatomopatologia e citologia Procedimento de coleta.** 2016. Disponível em <<http://www.diagnosticosdobrasil.com.br/wp-content/uploads/2016/03/Anatomopatologia-e-Citologia1.pdf>> com acesso em 20 de julho de 2018.

EDELWEISS, Nina; LIVI, Maria Aparecida Castro. **Algoritmos e programação com exemplos em Pascal e C.** 1ª Edição, São Paulo. Editora Bookman, 2014.

FARIA, Ruben. **O que é JavaScript?** Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\\_steps/O\\_que\\_e\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript)>. Acesso em 11 janeiro de 2018

FELLIPE, Deivison. **Lena.js.** Disponível em: <<https://webserver2.tecgraf.puc-rio.br/~mgattass/fcg/trb13/DavidsonFelipe/t1.html>>, 2013. Acesso em 04 março de 2017

FIREBASE, GOOGLE. **Cloud Storage.** Disponível em: <<https://firebase.google.com/docs/storage/?hl=pt-br>>, 2018. Acesso em 20 de maio de 2018.

FIREBASE, GOOGLE. **Firestore Realtime Database**. Disponível em: <<https://firebase.google.com/docs/database/?hl=pt-br>>, 2018. Acesso em 20 de maio de 2018.

FOCUSFOTO, Escola de Fotografia. **Conheça a profundidade de cor**. Disponível em: <<https://focusfoto.com.br/conheca-profundidade-de-cor/>>, 2014. Acesso em 21 de Fevereiro de 2019.

GALVANI, Ricardo. **Qual a diferença entre Definição, Resolução, Dimensão e Densidade de imagem?** <<https://www.rgbcomunicacao.com.br/blog/qual-a-diferenca-entre-definicao-resolucao-dimensao-e-densidade-de-imagem.html>>. Acesso em 10 de março de 2017.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento digital de imagens**. 3ª Edição, São Paulo. Editora Pearson, 2010.

JAVED, Adeel. **Criando projetos com Arduino para Internet das Coisas**. 1ª Edição, São Paulo. Editora Novatec, 2017.

JUNIOR, Sergio Luiz Stevan; SILVA, Rodrigo Adamshuk. **Automação e Instrumentação Industrial com Arduino: Teoria e Projetos**. 1ª edição, São Paulo, 2015. Editora Érica.

LACEN. **Manual de Coleta e Envio de Amostras Biológicas ao Lacen/PR**. Disponível em: <[http://www.lacen.saude.pr.gov.br/arquivos/File/Manuais/Manual\\_de\\_Coleta\\_e\\_Envio\\_de\\_Amostras\\_Biologicas.pdf](http://www.lacen.saude.pr.gov.br/arquivos/File/Manuais/Manual_de_Coleta_e_Envio_de_Amostras_Biologicas.pdf)>, 2016. MANUAL 1.30.001 / REVISÃO 02. Acesso em 14 abril de 2017.

LEITE, Enio. **Fotografia digital – aprendendo a fotografar com qualidade**. São Paulo, 3ª edição, 2015. Editora Viena.

LUZ, A.; ARAÚJO, D. Plataforma Colaborativa para Auxílio na Identificação de Casos Suspeitos de Doenças da Tríplice Epidemia e Focos do Mosquito Aedes Aegypti. **Revista de Engenharia e Pesquisa Aplicada**, v. 3, n. 1, 26 dez. 2016.

MACHADO, André Wilson; SOUKI, Bernardo Quiroga. **Simplificando a obtenção e a utilização de imagens digitais - scanners e câmeras digitais**. Disponível em: <<http://www.scielo.br/pdf/dpress/v9n4/a12v9n4>>, 2004. Acesso em 21 de fevereiro de 2019.

MAES, Jefferson. **Firestore o que é? E para que serve?** Disponível em: <<https://digitalprimews.com/google-firebase/>>. Acesso em 20 de maio de 2018.

MDN, WEB DOCS. **HTML5**. Disponível em < <https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>>, 2017. Acesso em 20 de maio de 2018.

MICROSOFT; **HTML5 - Guia do desenvolvedor para o HTML5 Canvas**. Retirado do site <<https://msdn.microsoft.com/pt-br/library/dn151487.aspx>>, 2013. Acesso em 06 março de 2018.

MIYAGUSKU, Renata Hiromi Minami. **Desvendando os recursos do CSS: Aprenda passo a passo a criar folhas de estilos**. 1ª Edição, São Paulo. Editora Digerati Books, 2007.

MOLINARI, Leonardo. **Cloud Computing: A inteligência da nuvem e seu novo valor em TI**. 1ª Edição, São Paulo. Editora Érica, 2018.

MOLINARO, Etélcia; CAPUTO, Luzia Fátima Gonçalves; AMENDOEIRA, Maria Regina Reis. **Conceitos e Métodos para formação de profissionais em laboratórios de saúde - volume 2**. 1ª Edição, Rio de Janeiro. Editora EPSJV/IOC, 2010.

MOREIRA, Adriana; SILVESTRE, Pedro. **Compressão de Imagem**. Disponível em: <<https://greeceintech.wordpress.com/2014/11/14/compressao-de-imagens/>>, 2015. Acesso em 10 de março de 2017.

MORENO, Luciano. **Digitalização de imagens - Resolução de um scanner**. Disponível em: <<http://www.criarweb.com/artigos/digitalizacao-imagens-resolucao-de-scanner.html>>, 2008. Acesso em 10 de março de 2017.

NETO, Manoel. **Desenvolvimento de Aplicações Ubíquas com Arduino e Raspberry PI**. Disponível em: <[https://www.researchgate.net/publication/268685128\\_Desenvolvimento\\_de\\_Aplicacoes\\_Ubiquas\\_com\\_Arduino\\_e\\_Raspberry\\_Pi](https://www.researchgate.net/publication/268685128_Desenvolvimento_de_Aplicacoes_Ubiquas_com_Arduino_e_Raspberry_Pi)>, 2014. Acesso em 21 de fevereiro de 2019.

NIST. **The NIST definition of cloud computing**. Disponível em: <<https://csrc.nist.gov/publications/detail/sp/800-145/final>>, 2011. Acesso em 17 de fevereiro de 2019.

NODEBR. **O que é Node.JS?** Disponível em: <<http://nodebr.com/o-que-e-node-js/>>, 2016. Acesso em 21 de maio de 2018.

OLIANI, Breno Scalla. **Microscópio Eletrônico de Varredura**. Disponível em: <https://www.ebah.com.br/content/ABAAAAC8cAD/microscopio-eletronico-varredura>. Acesso em 04 março de 2018.

OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry PI**. 1ª Edição. Editora Novatec, São Paulo, 2017.

OLIVEIRA, André Schneider; ANDRADE, Fernando Souza de. **Sistemas Embarcados - Hardware e Firmware na prática**. 2ª Edição, São Paulo. Editora Érica, 2010.

ONCOGUIA, Instituto. **Tipos de Citologia**. Disponível em <<http://www.oncoguia.org.br/conteudo/tipos-de-citologia/7381/886/>>. Acesso em 20 de maio de 2018.

PATOLOGIA MÉDICA - Diagnóstico, pesquisa e consultoria. **Exames Cito patológicos**. Disponível em: <<http://novo.patologiamedica.com.br/fazemos/citopatologia>>. Acesso em 20 de maio de 2018.

PEREIRA, Erlyck Lucena Duarte; REGIS, Carlos Danilo Miranda. Determinação de Tipo Sanguíneo nos Sistemas ABO e Rh Utilizando Processamento Digital de Imagens. **Revista de Tecnologia da Informação e Comunicação**, [S.l.], v. 5, n. 1, p. 29-38, maio 2015. Disponível em: <<http://rtic.com.br/index.php/rtic/article/view/61>>. Acesso em: 21 fev. 2019.

RAMOS, Rafael; RALHA, Célia; TEODORO, George. **Avaliação de Cluster Raspberry PI para Execução de Aplicações de Análise de Imagens Microscópicas Médicas**. Disponível em: <[https://www.researchgate.net/publication/308636572\\_Avaliacao\\_de\\_Cluster\\_Raspberry\\_Pi\\_para\\_Execucao\\_de\\_Aplicacoes\\_de\\_Analise\\_de\\_Imagens\\_Microscopicas\\_Medicas](https://www.researchgate.net/publication/308636572_Avaliacao_de_Cluster_Raspberry_Pi_para_Execucao_de_Aplicacoes_de_Analise_de_Imagens_Microscopicas_Medicas)>, 2016. Acesso em 21 de fevereiro de 2019.

RASPBERRYPI. Blog: **RaspBerry PI 2 on Sale**. Disponível em: <<https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>>. Acesso em 04 março de 2017.

SANTOS, Rafael. **Introdução à programação orientada a objetos usando JAVA**. 2ª Edição, Rio de Janeiro. Editora Elsevier, 2013.

SCHILDT, Herbert. **C Completo e Total**. 3ª ed. Revista e Atualizada. Makron Books, 1997.

SCHLAFFER , Georg. **O que você sempre quis saber sobre microscopia digital, mas nunca perguntou**. Disponível em <<https://www.leica-microsystems.com/pt/empresa/o-que-voce-sempre-quis-saber-sobre-microscopia-digital/>>. Acesso em 28 de julho de 2018.

SCURI, Antônio Escaño. **Fundamentos da Imagem Digital**. Disponível em <<https://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>. 2002. Acesso em 28 de julho de 2018.

SILVA, Maurício Samy. **Fundamentos de HTML 5 e CSS3**. 1ª ed. Editora Novatec, São Paulo, 2015.

SOARES, Ana. **Introdução à imagem digital, definição de pixel e cores digitais**. Disponível em <<https://anasoares1.wordpress.com/2010/11/22/introducao-a-imagem-digital-definicao-de-pixel-e-cores-digitais/>>, 2010. Acesso em 20 de agosto de 2018.

UNIPAN, União Pan-americana de Ensino (Faculdade de Ciências Aplicadas de Cascavel – FACIAP). **Linguagem C/C++**. Núcleo de Ciência da Computação FACIAP. 2004

UPTON, Eben; HALFACREE, Gareth. **Raspberry PI - Manual do Usuário**. 1ª Edição. Editora Novatec. 2013