



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM COMPUTAÇÃO**

JOSÉ RANIERY LUCAS DOS SANTOS

**PROPOSIÇÃO ARQUITETURAL DE UM SISTEMA *MOBILE* PARA
AUTOMATIZAÇÃO DA COLETA DE DADOS DOS AGENTES DE COMBATE A
ENDEMIAS DA CIDADE DE PATOS**

**PATOS
2020**

JOSÉ RANIERY LUCAS DOS SANTOS

**PROPOSIÇÃO ARQUITETURAL DE UM SISTEMA *MOBILE* PARA
AUTOMATIZAÇÃO DA COLETA DE DADOS DOS AGENTES DE COMBATE A
ENDEMIAS DA CIDADE DE PATOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Computação da Universidade Estadual da Paraíba, em cumprimento a exigências para obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Prof. Ms. Pablo Ribeiro Suárez

**PATOS
2020**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S237p Santos, José Raniery Lucas dos.
Proposição arquitetural de um sistema mobile para automatização da coleta de dados dos agentes de combate a endemias da cidade de Patos [manuscrito] / Jose Raniery Lucas dos Santos. - 2020.
74 p. : il. colorido.
Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2020.
"Orientação : Prof. Me. Pablo Ribeiro Suárez ,
Coordenação do Curso de Computação - CCEA."
1. Dispositivos Móveis. 2. Aplicativo mobile. 3. Automatização. 4. Agente de Endemias. I. Título
21. ed. CDD 005.3

José Raniery Lucas dos Santos

**PROPOSIÇÃO ARQUITETURAL DE UM SISTEMA MOBILE PARA
AUTOMATIZAÇÃO DA COLETA DE DADOS DOS AGENTES DE COMBATE A
ENDEMIAS DA CIDADE DE PATOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciências da Computação.

Aprovado em 11/12/2020

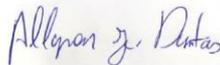
BANCA EXAMINADORA



Prof. Pablo Ribeiro Suárez
(Orientador)



Prof. Janine Vicente Dias
(Examinadora)



Prof. Allyson Jeronimo Dantas
(Examinador)

Dedico este trabalho a Deus que sempre esteve ao meu lado em todos os momentos e me ajudou a superar todos os desafios que surgem ao longo da minha caminhada.

AGRADECIMENTOS

Em primeiro lugar, quero agradecer a Deus por ter me dado força de vontade para concluir este trabalho, este último ano que está passando foi bem complicado para mim e se não fosse a sua enorme graça eu não teria concluído este trabalho.

Aos meus familiares, que sempre estiveram ao meu lado e me ajudaram nos momentos de maior desafio da minha vida, que sempre me deram forças para continuar e superar todas as adversidades que surgiam ao longo do caminho.

Aos meus professores que passaram por minha vida, carrego cada um de vocês em meus pensamentos e boa parte do que sou como cidadão se deve aos seus ensinamentos. Em especial ao meu orientador professor Pablo Ribeiro Suárez que deu uma força enorme, não só ao longo de todo curso, como também, neste trabalho de conclusão de curso.

Aos meus amigos e colegas que sempre me ajudaram e estiveram ao meu lado nesta longa caminhada e sempre estiveram à disposição para me ajudar sempre que precisava.

RESUMO

Endemias são doenças que ocorrem de forma sazonal em algumas regiões do mundo. A Dengue, a Febre Amarela, a Zika e a Chikungunya são as doenças endêmicas mais comuns que ocorrem no Brasil. Estas doenças são transmitidas por um vetor conhecido como o *Aedes Aegypti*, o popular mosquito da dengue. Para controlar o surto destas doenças, o governo federal criou uma categoria profissional chamada de Agente de Combate a Endemias. Dentre as atribuições destes profissionais estão as de inspecionar e tratar os depósitos com água parada, eliminar os focos e orientar a comunidade acerca dos devidos cuidados, outra atribuição dos agentes de endemias é a de coletar as informações referentes ao seu dia a dia de trabalho. Todas estas informações são escritas em boletins de papel para serem enviados ao Ministério da Saúde. Como forma de melhorar esta tarefa, este trabalho propõe a arquitetura para o desenvolvimento de um sistema mobile chamado de Itinerário Eletrônico capaz de automatizar e agilizar o armazenamento e envio destes dados. A proposição arquitetural deste aplicativo obedecerá a metodologia de desenvolvimento easYProcess (YP) seguindo o fluxo de trabalho como a Definição de Papeis, Inicialização e Planejamento para a posterior Implementação do sistema.

Palavras-Chaves: Endemias, *Aedes Aegypti*, Agente de Endemias, Sistema *Mobile*, *Android*.

ABSTRACT

Endemics are diseases which occur in a seasonal way in some regions of the world. *Dengue* fever, yellow fever, zika and chikungunya fever are the most common endemics in Brazil. These diseases are transmitted by a vector called *Aedes Aegypti*, commonly known as *dengue* mosquito. In order to control the outbreak of these diseases, the Federal Government created a category of professionals called Endemic Combat Agents. Among the functions of these professionals are the inspection and treatment of still water tanks, to eliminate the outbreaks and raise the community awareness for all the care needed, another function of these agents is to collect information regarding their work routine. All information is written on paper reports in order to be sent to the Ministry of Health. As a way of improving this function, this work proposes the architecture in order to develop a mobile system called Itinerário Eletrônico (Electronic Itinerary) which enables them to automatize and streamline the storage and transmission of data. The architectural proposition of this application will obey the easyProcess (YP) development methodology, following workflow such as definition of roles, Initialization and Planning for the further implementation of the system.

Key Words: Endemics; *Aedes Aegypti*; Endemic Combat Agents; Mobile System; Android.

LISTA DE FIGURAS

Figura 1 - Cálculo da Taxa de Incidência de Endemias	9
Figura 2 - Celulares no Mundo (Bilhões).....	13
Figura 3 - Diagrama que representa a Arquitetura do <i>Android</i>	16
Figura 4 - Exemplo de troca de informações entre um cliente e servidor usando a tecnologia <i>REST</i>	18
Figura 5 - Exemplo de um Código JSON.....	19
Figura 6 - Síntese de Fluxo do Processo YP	21
Figura 7 - Tarefa do <i>Login</i>	29
Figura 8 - Tarefa Tela de Apresentação	30
Figura 9 - Tarefa Itinerário	30
Figura 10 - Tarefa resumo das visitas.....	31
Figura 11 - Resumo de Visitas	31
Figura 12 - Resumo de Visitas	32
Figura 13 - Tarefa Pendências	32
Figura 14 - Tarefa Arquivo.....	33
Figura 15 - Tarefa Informações Sobre o Aplicativo.....	33
Figura 16 - Telas de Login e Apresentação	39
Figura 17 - Telas do CRUD de Bairros	40
Figura 18 - Telas do CRUD de Quarteirões	40
Figura 19 - Telas do CRUD de Imóveis	41
Figura 20 - Telas dos Resumos de Visitas Domiciliares	42
Figura 21 - PDF do Resumo de Visitas	42
Figura 22 - Telas da Relação de Imóveis Pendente.....	43
Figura 23 - Telas da Relação de Arquivos do Aplicativo.....	43
Figura 24 - Tela de Informações do Aplicativo.....	44
Figura 25 - Arquitetura do Sistema	45
Figura 26 - Modelo Lógico de Dados do Aplicativo.....	46
Figura 27 - Big Chart.....	50

LISTA DE QUADROS

Quadro 1 - Equipe envolvida no desenvolvimento do aplicativo.....	22
Quadro 2 - Requisitos não-funcionais do Aplicativo Itinerário Eletrônico.....	27
Quadro 3 - Objetivos de usabilidade	28
Quadro 4 - User Stories e Teste de Aceitação	34
Quadro 5 - Release 01 - (06/08/2019) Gerente: Raniery.....	47
Quadro 6 - Release 02 - (05/09/2019) Gerente: Raniery.....	47
Quadro 7 - Release 02 – (05/09/2019 – 12/09/2019) Gerente: Raniery	47
Quadro 8 - Big Chart	50
Quadro 9 - Tabela de Riscos.....	54

SUMÁRIO

1 INTRODUÇÃO.....	5
1.1 Problemática	6
1.2 Justificativa.....	6
1.3 Objetivos.....	7
1.3.1 <i>Objetivo geral</i>	7
1.3.2 <i>Objetivos específicos</i>	7
1.4 Metodologia.....	7
1.5 Estrutura do Trabalho	8
2 REVISÃO BIBLIOGRÁFICA	9
2.1 Doenças Endêmicas.....	9
2.2 Agentes de Combate a Endemias	11
2.3 Dispositivos Móveis	12
2.4 Android.....	13
2.5 Serviços <i>Web</i>	17
2.5.1 <i>Web service rest</i>	18
2.5.2 <i>Json</i>	18
2.6 Considerações Finais	20
3 PROPOSIÇÃO ARQUITETURAL DE UM APLICATIVO <i>MOBILE</i> PARA SISTEMATIZAÇÃO DO TRABALHO DOS ACEs DA CIDADE DE PATOS-PB	21
3.1 Definição de Papeis	22
3.2 Conversa com o Cliente.....	22
3.2.1 <i>Documento de visão</i>	22
3.2.2 <i>Descrição do sistema</i>	23
3.2.3 <i>Requisitos funcionais</i>	24
3.2.4 <i>Requisitos não-funcionais</i>	27
3.2.5 <i>Perfil do usuário</i>	27
3.2.6 <i>Objetivos de usabilidade</i>	28
3.3 Inicialização.....	29
3.3.1 <i>Modelo de tarefas</i>	29
3.3.2 <i>User stories e teste de aceitação</i>	34
3.3.3 <i>Protótipo da interface</i>	39
3.3.4 <i>Projeto arquitetural</i>	44
3.3.5 <i>Modelo lógico de dados</i>	45
3.4 Planejamento (Release e Iteração).....	47
3.4.1 <i>Plano de Release</i>	47
3.4.2 <i>Plano de iteração</i>	48
3.4.3 <i>Big chart</i>	50
3.4.4 <i>Análise de riscos</i>	51
3.5 Considerações Finais	54
4 CONCLUSÃO.....	55
4.1 Considerações Finais	55
4.2 Contribuições da Pesquisa	55
4.3 Limitações da Pesquisa.....	55
4.4 Trabalhos Futuros	56
REFERÊNCIAS	57
APÊNDICE A – Iterações do Projeto Que Não Foram Concluídas.....	60

1 INTRODUÇÃO

As endemias são doenças de origem parasitárias ou transmitidas por vetor. Essas doenças predominantemente ocorrem em espaços rurais, o ambiente dessa localidade como clima e tempo são condições perfeitas para a existência de algumas determinadas endemias, os surtos ocorridos dessas endemias são chamados de epidemias. No país, por séculos essas doenças se constituem como preocupação central da saúde pública brasileira (SILVA, 2003).

Doenças transmitidas por contato direto, as quais incluem as transmitidas por vetor, requerem a existência de um ambiente propício para a replicação de mosquito ou carrapatos. Majoritariamente, os vetores se reproduzem em regiões com clima quente e úmido, mas enquanto alguns vetores se reproduzem em águas limpas e paradas, como o *Aedes Aegypti*, vetor da Dengue e Febre Amarela, outros necessitam de matéria orgânica, como o *Lutzomyia*, vetor da *leishmaniose* visceral. Existem também as doenças que são transmitidas por formas infectantes presentes no ambiente, como a esquistossomose, relacionada à falta de saneamento básico (MOURA, 2012).

Uma das endemias com mais incidências no Brasil é a dengue, uma doença viral transmitida pelo *Aedes Aegypti* e que foi identificada no país em 1986. Estima-se que ocorram anualmente 50 milhões de infecções no mundo. Recentemente, em 2015, outras duas endemias foram identificadas em território nacional e também são transmitidas pelo mesmo mosquito transmissor da dengue, se trata da febre *chikungunya* em 2014 e a febre zika (BRASIL, 2001).

O número de casos de dengue segue crescendo no Brasil. Nas primeiras 14 semanas de 2020, o país já registrou 525.381 casos prováveis de dengue e 181 mortes provocadas pela doença. Também foram registrados este ano 15.051 casos e três mortes por *chikungunya*, e 2.054 casos de zika (BRASIL, 2001).

O Centro-Oeste é a região com maior incidência de casos (606,7 casos por 100 mil habitantes), seguida das regiões Sul (589,9 casos por 100 mil habitantes), Sudeste (226,9 casos por 100 mil habitantes), Norte (76,6 casos por 100 habitantes) e Nordeste (61,4 casos por 100 mil habitantes). Neste cenário, destacam-se os estados do Acre, São Paulo, Paraná, Mato Grosso do Sul, Mato Grosso, Distrito Federal e Goiás com incidências acima de 300 casos por 100 mil habitantes (BRASIL, 2001).

O profissional responsável pelo controle do mosquito *Aedes Aegypti* e demais vetores de endemias é o Agente de Combate a Endemias (ACE), esta categoria profissional surgiu no

final dos anos 90 de forma emergencial com o objetivo de erradicar o mosquito da dengue e teve sua atividade regulamentada através da Lei n. 11.350 de 05 de outubro de 2006 e trabalha em campo realizando visitas domiciliares com uma zona fixa de 800 a 1000 imóveis, visitados em ciclos bimensais em todos os municípios infestados por este vetor. Ele tem como obrigação básica: descobrir focos, destruir e evitar a formação de criadouros, impedir a reprodução de focos e orientar a comunidade com ações educativas (BRASIL, 2001 p. 27).

1.1 Problemática

Durantes as visitas, os agentes de endemias registram todas as informações coletadas em boletins de papel chamado de resumo de visitas domiciliares. Nestes boletins estão todas as relações de imóveis a serem visitados com os seus respectivos endereços como: i) Rua, Quarteirão e Bairro; ii) Horário, data e situação das visitas; e iii) Os tipos de depósitos, quantidades de focos dos mosquitos encontrados e larvicida utilizado. Além disso, a cada fim de expediente, os agentes realizam o fechamento do boletim realizando os cálculos de todos os imóveis visitados, fechados, o número de depósitos inspecionados, eliminados e tratados. Depois de conclusos, os resumos são entregues aos supervisores de campo que os levam para sede da coordenação da vigilância ambiental.

Todas essas tarefas são feitas manualmente, com o uso dos boletins e lápis, gerando uma certa monotonia e fazendo com que alguns agentes venham a cometer alguns erros durante o fechamento dos boletins, além de ser uma prática bem ultrapassada que demanda uma quantidade de oitenta boletins por agentes durante todo o ciclo de trabalhos. E por se tratarem de registros escritos em papeis, podem ser facilmente rasurados e perdidos.

Diante do exposto, o estudo pretende desvendar: **como o uso de aplicativos para plataforma *Android* pode auxiliar na eficiência dos registros de informações realizados pelos Agentes de Combate a Endemias na cidade de Patos-PB?**

1.2 Justificativa

Com o avanço da engenharia da computação no desenvolvimento de equipamentos computacionais portáteis, se torna possível a criação de um sistema que automatize todos os trabalhos dos Agentes de Combate a Endemias na coleta desses dados, dando mais praticidade e dinâmica a essa atividade, fazendo com que aquele trabalho manual realizado em papeis seja

substituído por uma solução computacional através de um aplicativo móvel, permitindo maior segurança e agilidade no processo de preservação desses dados.

1.3 Objetivos

1.3.1 Objetivo geral

O trabalho objetiva propor o desenvolvimento de um aplicativo para plataforma *Android* capaz de automatizar o trabalho de coleta e registro de dados/informações escrito pelos ACE no município de Patos-PB.

1.3.2 Objetivos específicos

- Realizar um estudo sobre as questões relacionadas às doenças endêmicas, dentre elas, as que são transmitidas pelo mosquito *Aedes Aegypti* e sobre trabalho dos Agentes de Combate a Endemias no controle desse vetor;
- Identificar as tecnologias associadas ao desenvolvimento *Mobile* com ênfase na plataforma *Android*;
- Realizar um estudo sobre o sistema operacional móvel *Android* e sobre o desenvolvimento de aplicativos;
- Elaborar e projetar diagramas descrevendo a arquitetura e modelos de dados do aplicativo para posterior implementação e avaliação do aplicativo.

1.4 Metodologia

A metodologia deste trabalho consiste em três etapas que pretende levantar o máximo de requisitos possíveis que irão ajudar no objetivo final. Primeiramente será realizada uma revisão bibliográfica, que obedece a uma lógica qualitativa. Através dos estudos bibliográficos, informações importantes referentes a doenças endêmicas, o trabalho dos agentes de endemias e sobre a plataforma na qual será feita o sistema, foram conhecidas, assim podendo ajudar ainda mais na realização do trabalho.

Com essas informações, o próximo passo é a proposição arquitetural de um sistema *Mobile*, tendo como método escolhido para a realização da aplicação a metodologia

easYProcess, uma metodologia de ambiente acadêmico que consegue obedecer aos requisitos fundamentais para o desenvolvimento de um software. A seguir, está demonstrada a sequência de tarefas que será construída ao longo do processo arquitetural:

- Levantamento dos requisitos;
- Análise dos requisitos;
- Elaboração do projeto;

A próxima etapa é a coleta dos dados que obedece a uma lógica quali-quantitativa, após a implantação do sistema, será realizada uma avaliação com um grupo de usuários escolhidos quanto a sua satisfação com o aplicativo desenvolvido. Todas as informações obtidas por meio dessa pesquisa serão importantes para futuras melhorias que o sistema venha a necessitar.

1.5 Estrutura do Trabalho

Este trabalho apresenta quatro capítulos e está organizado da seguinte maneira: no Capítulo 1 é apresentada uma visão geral desta investigação com relação a contextualização do problema, objetivos, justificativa do trabalho e metodologia aplicada; no Capítulo 2, são apresentados os temas relacionados a pesquisa; no Capítulo 3 é apresentado o estado da arte onde serão descritos todas as etapas referentes a proposição do projeto arquitetural de sistema mobile Itinerário Eletrônico, onde será elaborado o Documento de Visão, Modelo da Tarefa, User Stories e Teste de Aceitação, Protótipo da Interface, Projeto Arquitetural, Modelo Lógico de Dados, Plano de Release, Plano de Iteração, Big Chart e Análise de Riscos. E por fim o Capítulo 4 onde são apresentadas as considerações finais e as recomendações para trabalhos futuros e encontra-se, ao final, as referências e anexos.

2 REVISÃO BIBLIOGRÁFICA

2.1 Doenças Endêmicas

As doenças endêmicas são um dos principais problemas de saúde no mundo, principalmente nas regiões onde há falta de saneamento básico adequado, moradia precária e a falta de um sistema de saúde eficaz. Esses fatores associados contribuem bastante para que certas endemias venham a surgir com força total e acometer a população dessas localidades, o pico em grande escala de muitos casos de uma determinada endemia é chamado de epidemias (SILVA, 2003).

No Brasil, diversas comunidades ainda sofrem com a falta de moradia digna e a falta de esgotamento sanitário, além da precária distribuição de água potável. Por essa razão, algumas endemias se tornaram recorrentes, como a malária, Febre Amarela, doenças de chagas, algumas verminoses, *leishmanioses* e mais recentemente a dengue, *zika* e a *chikungunya* (CASTRO, 2016).

Ao longo da história, a humanidade vem sofrendo com várias doenças endêmicas, e diversos episódios de epidemias acima mencionadas, esses episódios não se sabe o tempo exato, mas alguns dão conta de ocorridos há centenas de anos antes de Cristo (TOLEDO JÚNIOR, 2006).

Para saber se um determinado quadro é endêmico ou epidêmico é necessário fazer um cálculo da taxa de incidências da determinada doença. Em seguida fazer uma comparação dos resultados com a média histórica de ocorrência do agravo tanto na localidade como na época do ano (MOURA; ROCHA, 2012).

Figura 1 - Cálculo da Taxa de Incidência de Endemias

$$\text{Taxa de incidência} = \frac{\text{número de casos novos de determinado agravo em determinada população em determinado período de tempo}}{\text{Total da população no mesmo local e período}} \times 100.000 \text{ hab.}$$

Fonte: Moura e Rocha (2012).

O Brasil sofre com as recorrentes epidemias de dengue que já matou milhares de pessoas. O Principal responsável pela transmissão desta doença é o mosquito *Aedes Aegypti*. Esse mosquito é um vetor que se ambientou muito bem no Brasil, o país por estar em uma

parte tropical do continente, favoreceu demais a proliferação desse inseto, um dos principais transmissores da dengue, não só dessas doenças, mas também de outras três como a Febre Amarela, a *Zika* e a *Chikungunya* (FIOCRUZ, 2020).

Para melhor compreender os transtornos causados por essas doenças, faremos aqui uma breve descrição de cada uma delas. A Dengue se trata de uma das doenças endêmicas mais conhecidas pela população brasileira, e com a maior incidência de picos epidêmicos por todo o país. Em épocas chuvosas ou de racionamento no abastecimento de água potável, ocorreram enormes infestações do mosquito e conseqüentemente as epidemias de dengue que infectaram muita gente inclusive levando algumas delas ao óbito. Em 2018 já são 1659 casos confirmados e 62 mortes, esses dados foram divulgados pelo Ministério da Saúde no dia 6 de julho (G1 NOTÍCIAS, 2018).

A Febre Amarela também se trata de endemia transmitida pelo mosquito *Aedes Aegypti*, ela é comum nas florestas tropicais da América e África com surtos periódicos (VASCONCELOS, 2003). Essa doença também é mortal quando atinge um estágio mais avançado.

A *zika* foi descoberta em Uganda no ano de 1947, nas florestas de *Zika*, por isso tem esse nome, o vírus da doença estava presente em macacos *Rhesus*, e fazia muito tempo que eles estavam isolados, mas recentemente um surto da febre *Zika* ocorreu nas ilhas Yap, na Micronésia em 2007 (BRASIL, 2001).

A *chikungunya* se trata de mais uma endemia transmitida pelo agente transmissor *Aedes Aegypti*, ela é provocada por um vírus chamado *Alphavirus CHIK*. Esse nome estranho que a doença tem é de origem maconde (uma língua falada no norte de Moçambique e sul da Tanzânia, no continente africano) que quer dizer “aqueles que se dobram” (BRASIL, 2001).

A *zika* e *chikungunya* são doenças de ocorrência recente no Brasil, mais precisamente no ano de 2014, por serem transmitidas pelo mosquito da dengue, rapidamente se alastraram em algumas regiões do Brasil, provocando verdadeiras epidemias, a *chikungunya* foi extremamente letal para pessoas idosas, pois os seus sintomas agravavam alguns problemas de saúde. Já a *zika* deixou sequelas graves em crianças recém-nascidas cuja as mães haviam contraído a *zika* durante a gestação, vários bebês nasceram com microcefalia, um distúrbio neurológico grave (BRASIL, 2001).

2.2 Agentes de Combate a Endemias

Os profissionais responsáveis pelo controle de alguns vetores, dentre eles, o próprio mosquito *Aedes Aegypti*, é o Agente de Combate a Endemias, também conhecido como ACEs. Esse profissional é encarregado de visitar os imóveis para fiscalizar, orientar a população em alguns cuidados básicos com os seus depósitos com água, eliminação de potenciais criadouros, tratamento de alguns depósitos que tenham focos ou com a possibilidade de surgir (BRASIL, 2001).

Além destas atribuições, o agente tem que realizar determinadas atividades em seu trabalho. Algumas das atribuições de combates aos vetores, como o próprio *Aedes Aegypti* estão descritas no Manual de Normas Técnicas da Vigilância Epidemiológica (BRASIL, 2001, p.27):

- Pesquisa larvária para levantamento de índice e descobrimento de focos;
- Eliminação de criadouros, por meio de controle mecânico (remoção, destruição, vedação, etc.);
- Orientação da população com relação aos cuidados básicos para evitar a infestação de mosquitos;
- Manter sempre atualizados a relação de imóveis que devem ser visitados durante os ciclos de trabalhos;
- Sempre registrar todas as informações referentes aos trabalhos realizados nos boletins diários.

Na cidade de Patos, são 62 Agentes de Combate de Endemias, cada agente é encarregado de uma determinada área fixa com algo em torno de 800 a 1000 imóveis a serem visitados dentro de um ciclo de trabalho que compreende um período de dois meses.

Os ACEs trabalham em equipes coordenadas por um supervisor de área, eles comandam os trabalhos na região de atuação da equipe, geralmente é um conjunto de bairros localizado em uma determinada parte da cidade.

O supervisor é responsável pela divisão de algumas tarefas de campo, pela distribuição do material de trabalho necessário no dia a dia, pela orientação e correção de algumas atividades dos agentes de campo, além de auxiliar em pendências e problemas que venham a surgir em cada área.

Acima do supervisor de área, tem o supervisor-geral, esse profissional por sua vez coordena os supervisores de área na hierarquia do trabalho, distribuindo as tarefas de cada

grupo de agentes e elaborando melhores estratégias que ajudarão no trabalho (BRASIL, 2001). Além do supervisor-geral, o responsável por todos os grupos de trabalhos tem também a figura do Coordenador de Vigilância Ambiental, um profissional comissionado ligado diretamente à Secretaria de Saúde de cada município.

2.3 Dispositivos Móveis

A tecnologia *Mobile* teve início, na teoria, em 1947. Porém, somente em 1973, foi realizada a primeira chamada feita através de um telefone móvel sem precisar de uma conexão com um fio. Foi um grande passo da tecnologia rumo ao surgimento dos dispositivos móveis (LG, 2019).

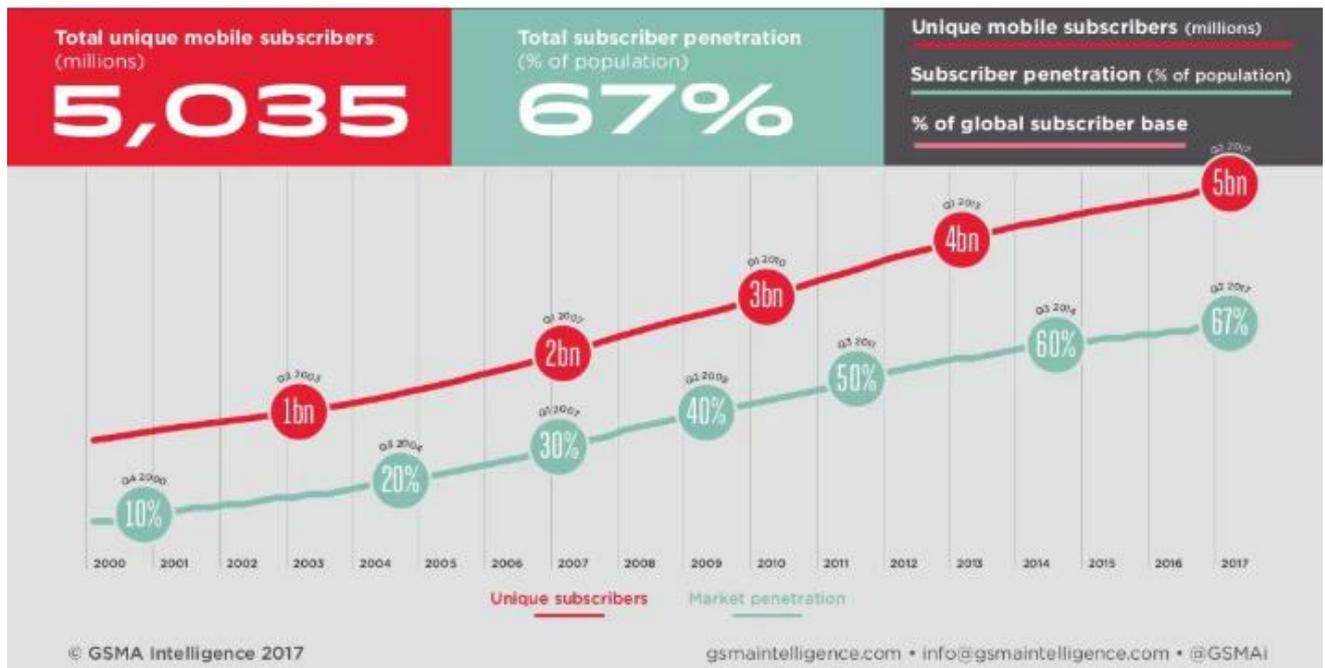
Em todo o planeta, já são mais de 5 bilhões de usuários de aparelhos celulares, esses são os dados do relatório a Economia Móvel 2019, da *Global System for Mobile Communications* (GSMA), esse número equivale a cerca de 67% da população do planeta (AGÊNCIA BRASIL, 2019).

No recorte por região, com maior percentual de celulares está a Europa, com 85%. Em seguida vêm a Comunidade dos Estados Independentes 80%, América do Norte 83%, América Latina 67%, Ásia e Pacífico 66%, Oriente Médio e Norte da África 64% e África Subsaariana 45%. A variação de penetração dos celulares evidencia a persistência de desigualdade regional no acesso a essa tecnologia. Em 2018, o Brasil chegou à marca de 204 milhões de *smartphones* (AGÊNCIA BRASIL, 2019).

Já o total de pessoas acessando a *internet* pelo celular ficou em 3,6 bilhões em 2018. O número de usuários corresponde a 47% dos habitantes do planeta. Os *smartphones* devem puxar esse crescimento. Em 2018, eles eram 60% dos dispositivos móveis em funcionamento (AGÊNCIA BRASIL, 2019).

O ano de 2018 marcou a hegemonia do 4G, que se tornou o principal padrão de conexão de dispositivos móveis no mundo, chegando a 3,4 bilhões, o equivalente a 43% do total. Do conjunto da base, 29% eram de conexões 2G e 28%, de 3G. Em 2025, a projeção da GSMA aponta que o 4G deve estar em 60% dos serviços (AGÊNCIA BRASIL, 2019).

Figura 2 - Celulares no Mundo (Bilhões)



Fonte: Tele Síntese (2018).

Como mostra a figura 2, dentre os aparelhos computacionais, os dispositivos móveis são os que ganham cada vez mais espaço no gosto das pessoas, sendo os mais procurados em relação a *notebooks* e *tablets*, por se tratarem de aparelhos com boa parte das funcionalidades de um computador de mesa. Por serem mais baratos dos que os computadores e *notebooks*, a sua procura é bem maior nas lojas.

2.4 Android

O *Android* é um sistema operacional móvel *Open Source*, desenvolvido pela empresa *Google* e baseado no núcleo da versão 2.6 do *Kernel Linux*, responsável por funcionalidades como segurança, gerência de memória, processos *driver* e rede (LECHETA, 2013).

Um consórcio de grandes empresas do mercado de dispositivos móveis contribui no desenvolvimento com o objetivo de propagar cada vez mais o acesso a essa tecnologia, esse consórcio se chama *Open Handset Alliance*, esse grupo conta com a participação de empresas conhecidas como a *Intel*, *Samsung*, *LG*, *Motorola*, *Sony Ericsson*, *Asus*, *Acer*, *Dell*, entre outras (LECHETA, 2013).

A principal vantagem do *Android* é a sua capacidade de desenvolvimento de aplicativos, sendo uma abordagem unificada dando a capacidade de fazer com que esses

aplicativos desenvolvidos possam ser executados em diferentes aparelhos desde que tenham como o seu sistema operacional o *Android* (LECHETA, 2013).

O *Android* dispõe das seguintes funcionalidades que dão a ele a capacidade de operar de forma simples e dinâmica todas as suas tarefas, são elas (LECHETA, 2013):

- Armazenamento – *SQLite*, uma base de dados relacional;
- Conectividade – suportando GSM/EDGE, IDEN, CDMA e outros;
- Troca de mensagens – capaz de suportar tanto o SMS como o MMS;
- Multitarefas – podendo gerenciar diversas tarefas e a execução de muitos aplicativos.

O *Android* em sua arquitetura dispõe de cinco camadas (LECHETA, 2013). São essas camadas que garante as funcionalidades característica do sistema:

- *Applications*: nesta camada são representadas as aplicações que executam sobre a plataforma. Essa camada pode ter tanto as aplicações nativas do próprio sistema como telefonia, navegador, calendário etc., como outras aplicações feitas por desenvolvedores. Essa característica faz do *Android* um sistema bem flexível dando um alto grau de extensibilidade para a plataforma;
- *Application Framework*: nessa camada ficam as APIs utilizadas pelas aplicações que executam sobre a plataforma. Esse *framework* disponibiliza alguns gerenciadores de serviços de telefonia, localização e notificação dentre outros;
- *Libraries*: esta camada possui as bibliotecas nativas escritas em C/C++ que fazem parte da plataforma, estão presentes nesta camada as API *OpenGL ES* para renderização em 3D, *SQLite* que gerencia o banco de dados do sistema, e suporte a muitos outros formatos de áudio e vídeos;
- *Runtime*: esta camada é responsável pela execução do sistema, nela constam componentes como o *core libraries*, que disponibiliza uma API *Java* utilizada para a programação, pois boa parte das funcionalidades encontradas em *Java SE* estão disponibilizadas para *Android* e o outro componente é o ART que é projetado para executar várias máquinas virtuais em dispositivos de baixa memória executando arquivos DEX, um formato de bytecodes projetado especialmente para Android, otimizando para oferecer consumo mínimo de memória (DEVELOPER ANDROID, 2020);
- *Kernel Linux*: esta camada é responsável por gerenciar todos os *drivers* do *smartphone*, as principais funcionalidades presentes em um aparelho são administradas

por esta camada, como, por exemplo, o *driver* da câmera, o *driver* do *wifi*, o *driver* do áudio, o *driver* de vídeo, *driver* do *touchscreen*, o *driver* do USB, o *driver* de energia e dentre outros.

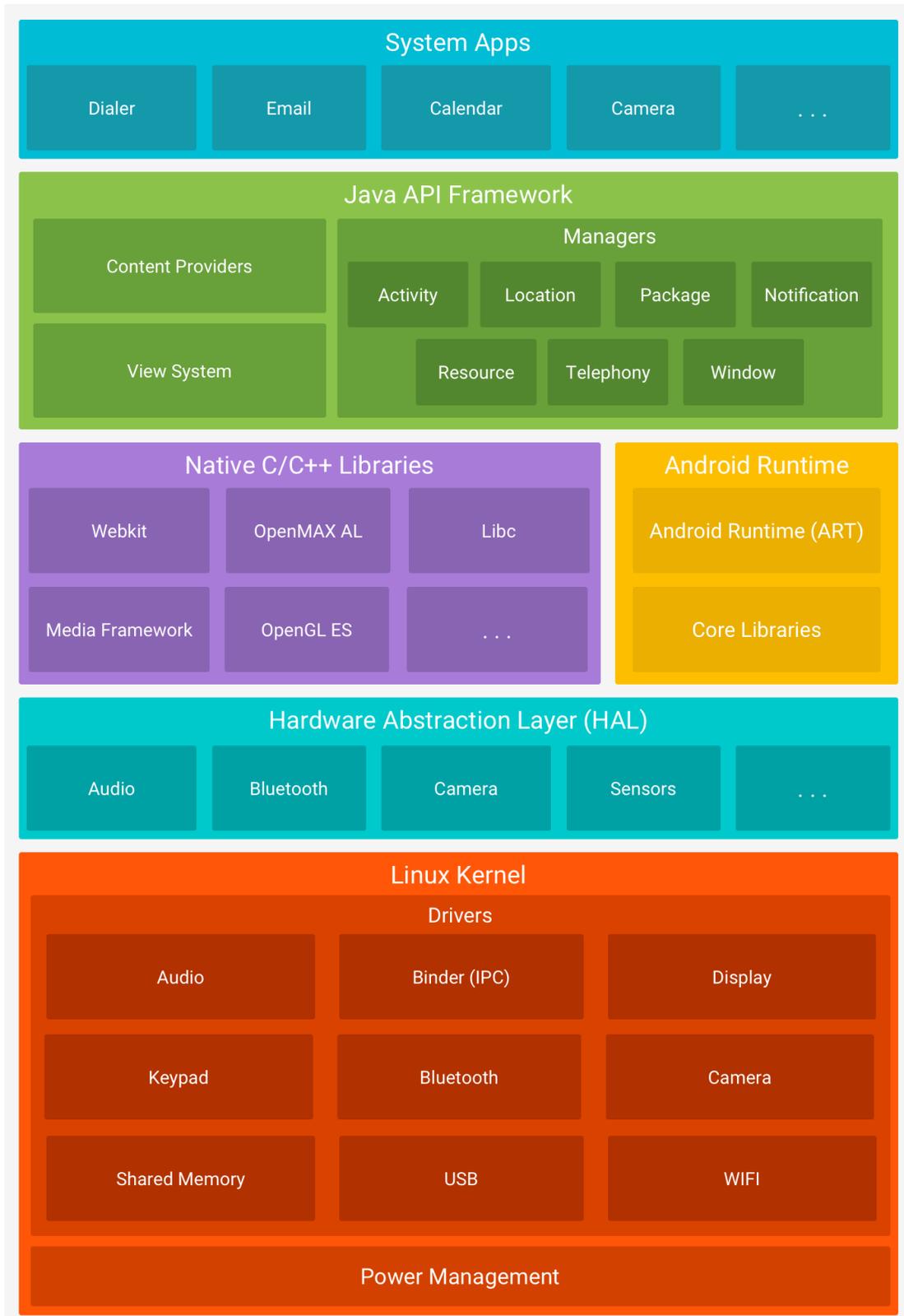
Duas linguagens de programação são suportadas nativamente e oficialmente, são elas:

- **Java:** é a linguagem padrão para escrever aplicativos do Android desde que a plataforma Android foi introduzida, em 2008. É uma linguagem de programação orientada a objetos originalmente desenvolvida pela Sun Microsystem em 1995 (agora é propriedade da Oracle) (TECMUNDO, 2020).
- **Kotlin:** em 2017, o Google anunciou o suporte ao Kotlin como uma linguagem alternativa de alto desempenho para a programação em Android. Kotlin é interoperável com Java, e todas as bibliotecas Java podem ser acionadas por Kotlin (TECMUNDO, 2020).

Duas abordagens multiplataforma contribui para o desenvolvimento de aplicativo Android, são elas:

- **React Native:** é um framework JavaScript criado para construir a interface do usuário em aplicativo móveis – smartphone e tablets. Criado pelo Facebook baseado no React – que cria interfaces voltadas para aplicações web, o React Native é a solução para criar aplicativos nativos tanto para Android quanto para iOS (DEVMEDIA, 2020).
- **Flutter:** é um framework construído pela Google para facilitar o desenvolvimento mobile multiplataforma (Android/iOS) que tem o Dart como principal linguagem de desenvolvimento. Ele utiliza uma abordagem até então única para lidar com os componentes nativos de cada plataforma, em que cada um deles é implementado pelo próprio framework e apresentado ao usuário por um motor de renderização próprio (DEVMEDIA, 2020).

Figura 3 - Diagrama que representa a Arquitetura do *Android*



Fonte: Developer (2020).

O *Android* é o sistema operacional mais utilizado na América Latina. A maioria dos usuários de *smartphones* possuem aparelhos com o *Android* como sistema operacional presente em seus dispositivos móveis. No Brasil esse número já ultrapassa os 90%, entre os anos 2012 e 2017, o crescimento de usuários do *Android* aumentou 90,4%. Isso mostra claramente o gosto que os brasileiros têm por aparelhos com esse sistema operacional (CARVALHO, 2017).

2.5 Serviços Web

Serviços *Web* ou *Web Services* é o nome da tecnologia que permite a comunicação entre aplicações independentemente de sistemas operacionais ou linguagens de programação. Essa arquitetura está sempre em comunicação com a rede e permite que diversos sistemas possam estar conectados até mesmo globalmente, permitindo ao máximo a integração (LECHETA, 2013).

Com a expansão da *internet*, muitos sistemas tiveram que evoluir com o intuito de permitir que certas aplicações pudessem funcionar mesmo em diversos lugares do mundo, sendo assim muitas delas mudaram a sua arquitetura permitindo que os seus serviços funcionassem de forma remota, facilitando os trabalhos dos seus usuários (LECHETA, 2013).

Os *Web Services* permitem que as aplicações enviem e recebam dados no formato XML (*Extensible Markup Language*), protocolo que permite a geração automática tanto do código cliente quanto do código servidor, cada aplicação pode ter a sua própria linguagem, mas essa por sua vez é traduzida para a forma de XML que é uma linguagem universal (LECHETA, 2013).

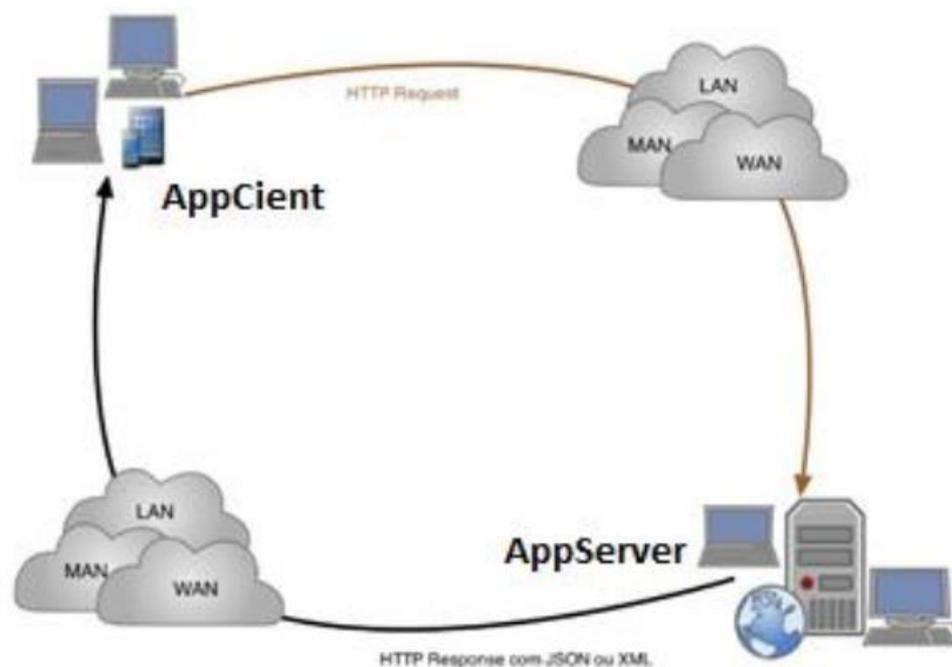
O protocolo SOAP, abreviação para *Simple Object Access Protocol*, é uma especificação para a troca de informações entre sistemas, ou seja, uma especificação de formato de dados para envio de estruturas de dados entre serviços, com um padrão para permitir a interoperabilidade entre eles. Seu design parte do princípio da utilização de XMLs para a transferência de objetos entre aplicações, e a utilização, como transporte, do protocolo de rede HTTP (DEVMEDIA, 2020).

2.5.1 Web service rest

O *REST* é um padrão de arquitetura de serviços *Web* que utiliza apenas o protocolo HTTP para as suas comunicações. O *REST* possui conjunto de operações responsáveis pela manipulação das informações, essas operações são chamadas de CRUD para a persistência de dados, às operações são: *GET*, *POST*, *PUT* e *DELETE* (LECHETA, 2013).

- GET: recupera os dados;
- POST: acrescenta dados no servidor;
- PUT: insere e atualiza dados;
- DELETE: apaga os dados.

Figura 4 - Exemplo de troca de informações entre um cliente e servidor usando a tecnologia *REST*



Fonte: Ferreira e Mota (2014).

2.5.2 Json

JSON (*JavaScript Object Notation* – Notação de Objetos *JavaScript*) é uma formatação leve de troca de dados. O JSON é um formato de texto completamente

independente de linguagem que utiliza texto legível a humanos para transmitir objetos de dados (JSON, 2018).

O JSON ganhou muita força nos últimos anos como o formato mais utilizado no retorno de *webservices*, devido a sua simplicidade em comparação com XML (LECHETA, 2013).

JSON ganhou muita força nos últimos anos como o formato mais utilizado no retorno de *webservices*, devido a sua simplicidade em comparação com XML

O JSON está constituído em duas estruturas:

- Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, *record*, *struct*, dicionário, *hash table*, *keyed list* ou *arrays* associativas (JSON, 2018).
- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência (JSON, 2018).

Figura 5 - Exemplo de um Código JSON

```
1 JSONObject json = new JSONObject ( jsonString );
2 try {
3 String campo1 = json . getString ( " campoObrigatorio " );
4 String campo2 = json . optString ( " campoOpcional ", null );
5 JSONObject objeto = json . getJSONObject ( " objetoAninhado " );
6 } catch ( JSONException e) {
7 e. printStackTrace ();
8 }
```

Fonte: Elaborado pelo próprio autor com o software Notepad++ (2020).

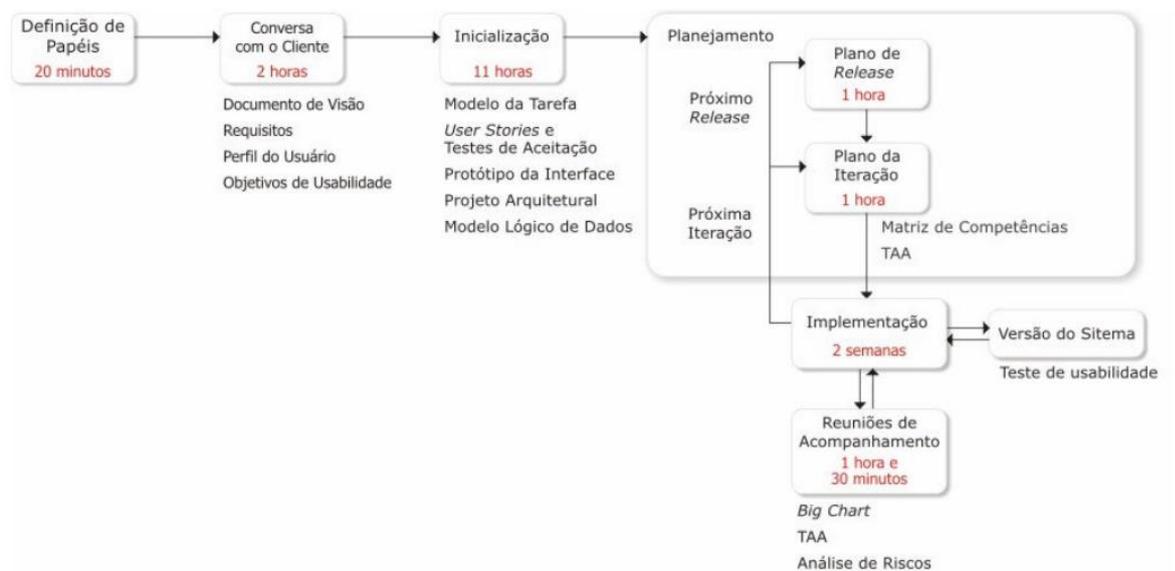
2.6 Considerações Finais

Neste capítulo, foi apresentado o referencial teórico utilizado como fonte para pesquisa que auxiliou no desenvolvimento do trabalho. Nele foi explicado sobre algumas doenças endêmicas conhecidas como a Dengue, Zika e Chikungunya e o seu principal vetor de transmissão, o mosquito *Aedes Aegypti*, falou sobre profissional responsável que dentre uma de suas atribuições, é a de controlar o índice de infestação deste mosquito. Também foi abordado sobre os dispositivos móveis, o sistema operacional Android e o desenvolvimento de aplicativos para este mesmo sistema com o intuito de desenvolver um aplicativo que dinamize e melhore o trabalho do profissional Agente de Endemias na coleta dos dados referente ao seu trabalho. A seguir, o capítulo 3 que mostrará o estudo de caso da pesquisa.

3 PROPOSIÇÃO ARQUITETURAL DE UM APLICATIVO *MOBILE* PARA SISTEMATIZAÇÃO DO TRABALHO DOS ACEs DA CIDADE DE PATOS-PB

Este estudo baseou-se na metodologia *easYProcess* YP (GARCIA; LIMA; FERREIRA et al., 2007) para a realização das etapas no desenvolvimento do aplicativo Itinerário Eletrônico. Onde cada etapa da metodologia será descrita ao longo do processo de construção do sistema. O primeiro artefato da metodologia YP versa sobre a definição dos papéis e em seguida sobre o documento de visão, onde é composto por cinco artefatos como: Descrição do Sistema, Requisitos Funcionais, Requisitos Não Funcionais, Perfil do Usuário e Objetivos de Usabilidade. Além disso, a inicialização também se faz presente com 6 artefatos como: Modelo de Tarefa, *UserStories* e Teste de Aceitação, Protótipo da Interface, Projeto Arquitetural e Modelo Lógico de Dados. O planejamento conta com quatro artefatos e eles são os seguintes: Plano de Release, Plano de Iteração, Big Chart e Análise de Risco. A seguir, serão demonstradas cada etapa do sistema Itinerário Eletrônico.

Figura 6 - Síntese de Fluxo do Processo YP



Fonte: YP (2007)

3.1 Definição de Papéis

Em uma equipe de desenvolvimento de *software*, os membros assumem os seguintes papéis: cliente, usuário, gerente, desenvolvedor e testador. Cada papel tem uma responsabilidade essencial no desenvolvimento do projeto. Alguns membros podem assumir mais de um papel em um projeto (GARCIA; LIMA; FERREIRA et al., 2007).

Quadro 1 - Equipe envolvida no desenvolvimento do aplicativo

Membros da Equipe	Papéis
Vigilância Ambiental de Patos	Cliente/Usuário
Allyson Jerônimo Dantas	Gerente/Desenvolvedor/Testador
José Raniery Lucas dos Santos	Gerente/Desenvolvedor/Testador

Fonte: Elaborado pelo autor (2020).

3.2 Conversa com o Cliente

Nas primeiras conversas com o cliente, foram obtidas todas as informações necessárias para orientar o desenvolvedor e dá a ele, uma ideia precisa de como deve ser feito o programa. O desenvolvedor procurou extrair o máximo de informações possíveis do cliente para transformá-las no produto desejado pelo mesmo (GARCIA; LIMA; FERREIRA et al., 2007).

3.2.1 Documento de visão

Após as primeiras conversas com o cliente, foi elaborado um documento de visão que explica com todos os detalhes que é o aplicativo Itinerário Eletrônico e todas as suas funções, como também explica a maneira como o aplicativo irá realizar as suas operações e tarefas, além de todas as suas características e o perfil de quem irá utilizá-lo. A seguir será mostrado todos os artefatos. Este artefato é composto por 5 subpartes:

- Descrição do Sistema;
- Requisitos Funcionais;
- Requisitos Não Funcionais;
- Perfil do Usuário;
- Objetivos de Usabilidade.

3.2.2 Descrição do sistema

O sistema terá como o objetivo automatizar a coleta dos dados feita pelos Agentes de Combate a Endemias durante a realização das suas visitas domiciliares. Em seu trabalho diário, os ACEs visitam os imóveis para realizar a inspeção do mesmo com o intuito de encontrar algum depósito com água que venha a ser um provável criadouro das larvas do mosquito da dengue, ou seja, algum foco do *Aedes Aegypti*.

No trabalho em campo os ACEs anotam todas as suas atividades em boletins de papel chamado de resumo de visita, neste boletim, basicamente tudo que é feito pelos agentes é escrito, desde os endereços dos imóveis, se o imóvel está aberto ou fechado ou se o morador está recusando a visita; a data e a hora da visita; informa a quantidade de depósitos por tipo, a quantidade de depósitos que foram descartados pelo agente na hora da visita, se alguns destes depósitos foram tratados; a quantidade de larvicida utilizado no tratamento e a quantidade de larvas coletadas por depósitos durante a visita.

O objetivo é automatizar toda coleta dos dados, através de um aplicativo instalado em um dispositivo móvel, onde o agente irá digitar as informações referentes a cada ação que ele irá realizar durante a visita, tornando mais prática e segura a tarefa de coletar os dados que os ACEs fazem durante o seu trabalho. Todas as informações serão armazenadas em um banco de dados do próprio aparelho para que sejam compartilhadas com um servidor, informando a coordenação da Vigilância Ambiental da cidade de Patos sobre toda as atividades feitas pelos ACEs para o programa de controle da dengue.

O sistema não só armazenará os dados, como também fará sempre o resumo dos trabalhos, tanto no final de cada expediente como também ao final de cada ciclo de trabalho, gerando relatórios parciais e finais e enviando todas essas informações para o sistema da vigilância ambiental de Patos. Desta forma o programa de controle da dengue pode realizar *feedback* acerca de como anda os trabalhos, formular novas estratégias de trabalho e analisar os trabalhos de forma individual e coletiva dos agentes, dando mais dinamismo aos trabalhos e facilitando o dia a dia dos profissionais que realizam as visitas domiciliares.

3.2.3 *Requisitos funcionais*

Os requisitos funcionais de um sistema descrevem todas as funções que serão executadas pelo mesmo durante a sua operação (GARCIA; LIMA; FERREIRA et al., 2007).

A seguir será exibida todas as funcionalidades do aplicativo Itinerário Eletrônico:

➤ Manutenção do usuário:

- Cadastro de usuário;
- Autenticação do usuário;

➤ Manutenção da interface inicial:

- Visualizar acesso ao itinerário;
- Visualizar acesso a relação de pendências;
- Visualizar acesso ao relatório de visita e trabalho;
- Visualizar acesso a informações do aplicativo;

➤ Manutenção de Bairros:

- Adicionar;
- Editar;
- Visualizar;
- Excluir;

➤ Manutenção de Quarteirões:

- Adicionar;
- Editar;
- Visualizar;
- Excluir;

➤ Manutenção de Imóveis:

- Adicionar;
- Editar;
- Visualizar;
- Excluir;

➤ Manutenção do Estado da Visita:

- Aberto;
- Fechado;
- Recusa;

➤ Manutenção do Dia da Visita:

- Atualizar Data da Visita Automaticamente;
- Atualizar Hora da Visita Automaticamente;

➤ Manutenção dos Tipos de Depósitos:

- Informar Depósito do Tipo A1;
- Informar Depósito do Tipo A2;
- Informar Depósito do Tipo B1;
- Informar Depósito do Tipo C1;
- Informar Depósito do Tipo D1;
- Informar Depósito do Tipo D2;
- Informar Depósito do Tipo E1;

➤ Manutenção de Depósitos Eliminados:

- Informar;
- Editar;

➤ Manutenção de Depósitos Tratados:

- Informar;
- Editar;

➤ Manutenção da Quantidade de Larvicida:

- Informar;
- Editar;

➤ Manutenção da Quantidade de Focos:

- Informar;
- Editar;

➤ Relatórios:

- Resumo de Visitas Diárias;
- Resumo de Repasses;
- Resumo do Ciclo de Trabalho;

➤ Regras de Negócios:

- Armazenar os dados coletados a cada visita feita no banco de dados do aparelho;
- Realizar o cálculo dos dados de todas as visitas feitas no dia de trabalho;
- Exibir todas as informações referentes a todos os dados coletados no dia de trabalho;
- Armazenar o resumo de trabalho do dia no banco de dados do aparelho;
- Enviar o resumo de todos os dados coletados no dia de trabalho para um banco de dados nas nuvens;
- Realizar o cálculo de todos os dados coletados durante o ciclo de trabalho;
- Exibir todas as informações referentes a todos os dados coletados durante o ciclo de trabalho;
- Armazenar o resumo de trabalho do ciclo no banco de dados do aparelho;

- Enviar o resumo de visitas e do ciclo de trabalho para um banco de dados nas nuvens.

3.2.4 Requisitos não-funcionais

São os requisitos relacionados a desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos definem as propriedade e restrições de um programa (GARCIA; LIMA; FERREIRA et al., 2007). No quadro abaixo serão exibidos os requisitos não-funcionais do Aplicativo Itinerário Eletrônico.

Quadro 2 - Requisitos não-funcionais do Aplicativo Itinerário Eletrônico

Requisitos não funcionais	Descrição
Utilizar Implementação Java SE	Esta é a linguagem de programação utilizada para escrever os códigos responsáveis pelas funções e comandos do aplicativo durante sua execução.
Interface Desenvolvida em XML	Esta é a linguagem de marcação responsável pelo <i>design</i> das páginas do aplicativo.
Bando de Dados <i>SQLite</i>	Banco de dados relacional para aplicativos <i>Android</i> .
Segurança	Para acessar o aplicativo, haverá a necessidade de um <i>login</i> com senha.
Conexão <i>Web</i>	A conexão <i>web</i> se dará pelo <i>JSON</i> que é um formato leve de intercâmbio de dados, permitindo o compartilhamento das informações com um servidor. A biblioteca utilizada é a <i>retrofit</i> , que oferece um padrão simples de implementação para transmissão de dados entre aplicação e servidor.

Fonte: Elaborado pelo autor (2020).

3.2.5 Perfil do usuário

Os usuários que irão utilizar este aplicativo são os profissionais Agentes de Combate à Endemias, com idades entre 18 e 60 anos, nível médio de escolaridade e conhecimento básico em computação. Como se trata de um aplicativo novo, boa parte desses profissionais precisarão de um breve treinamento, nada tão complexo, pois boa parte desses trabalhadores já estão situados de como funciona os aplicativos móveis mais populares e boa parte deles são usuários de algum tipo de aplicativo.

Este aplicativo tem como propósito ser intuitivo para facilitar o aprendizado dos profissionais, além disso, irá dispor de um pequeno tutorial que ensinará algumas das funções

básicas de funcionamento facilitando a completa aceitação do aplicativo por partes desses profissionais.

3.2.6 *Objetivos de usabilidade*

É o conjunto de metas de usabilidade que podem ser medidas e que devem ser alcançadas pelo Sistema Itinerário Eletrônico (GARCIA; LIMA; FERREIRA et al., 2007). O Quadro 3, a seguir, ilustra os objetivos de usabilidade elencados para o sistema anteriormente citado.

Quadro 3 - Objetivos de usabilidade

Objetivos de Usabilidade	
Objetivo	Mensuração
Ser visualmente atrativo	Verificar o nível de satisfação dos usuários com relação ao <i>design</i> da interface do aplicativo.
Baixa taxa de erros	Frequência com a qual as tarefas são concluídas sem que haja erros.
Ser o mais intuitivo possível	Observar a facilidade com a qual os usuários do aplicativo utilizam o programa sem nenhum problema.
Disponibilizar uma boa documentação	O grau de satisfação e a facilidade com a qual usuários entendem as informações referente ao sistema em uma documentação clara e objetiva.
Facilidade para memorizar	Verificar se o usuário é capaz de utilizar o programa sem a menor dificuldade tempos depois de ter utilizado pela última vez.

Fonte: Elaborado pelo autor (2020).

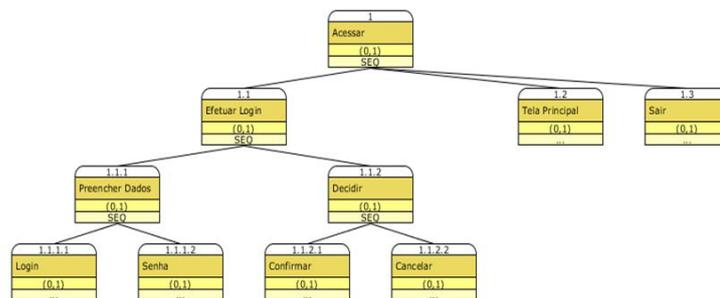
3.3 Inicialização

Foi iniciado as atividades de análise de sistema com o objetivo de construir um **Modelo da Tarefa**, *User Stories* e seus respectivos **Teste de Aceitação**, o **Protótipo de Interface**, a **Arquitetura do Sistema** e o **Modelo Lógico de Dados**.

3.3.1 Modelo de tarefas

No modelo da Tarefa será demonstrado, todas as sequências de tarefas realizadas pelo o usuário do Aplicativo Itinerário Eletrônico. **Tarefa Login:** Esta tarefa diz respeito ao acesso e segurança do sistema, o agente irá acessar o sistema através de um *login* e senha válido.

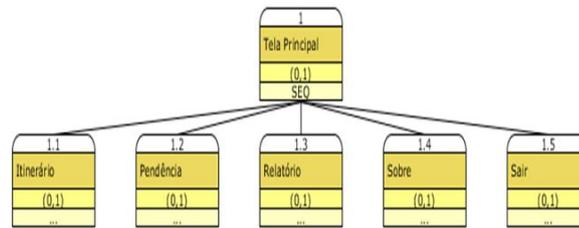
Figura 7 - Tarefa do Login



Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Tarefa Tela de Apresentação: esta é a tarefa em que o agente acessa a tarefa de apresentação do aplicativo e acessos a outras tarefas: Itinerário, Pendência, Arquivos e Informações do aplicativo.

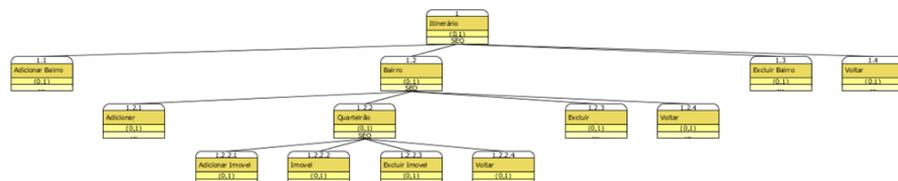
Figura 8 - Tarefa Tela de Apresentação



Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Tarefa Itinerário: esta tarefa mostra todas as seqüências de passos que o agente irá realizar para ter acesso a todas as funcionalidades como o Cadastrar, Pesquisar, Editar, Excluir (CRUD) das listas de Bairros, Quarteirões e Imóveis e o acesso ao resumo eletrônico de visitas domiciliares.

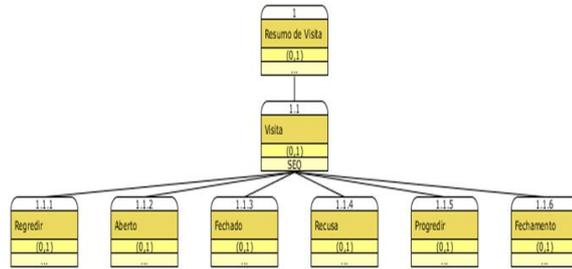
Figura 9 - Tarefa Itinerário



Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

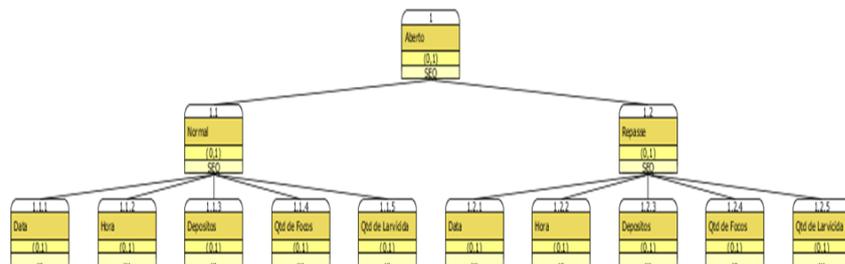
Tarefa Resumo de Visitas: esta é a tarefa que exibe o resumo eletrônico de visitas para que o agente colete as informações referentes as visitas realizadas durante o dia de trabalho.

Figura 10 - Tarefa resumo das visitas

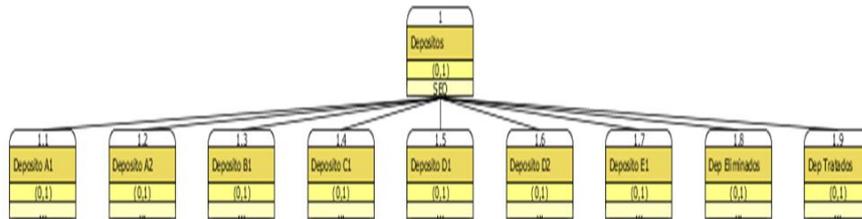


Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Figura 11 - Resumo de Visitas

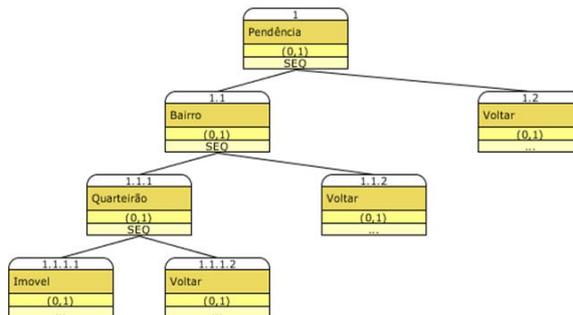


Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Figura 12 - Resumo de Visitas

Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

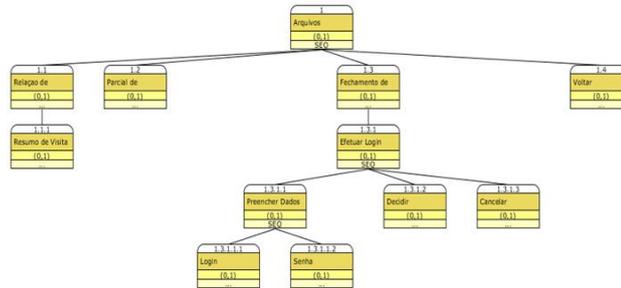
Tarefa Pendências: esta é a tarefa responsável pela exibição da relação de imóveis com as suas visitas pendentes, ou seja, imóveis que não foram visitados pelo agente, ou por estarem fechados ou porque o morador recusou a visita.

Figura 13 - Tarefa Pendências

Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Tarefa Arquivos: esta tarefa exibe o armazenamento de todas as informações coletadas pelo agente durante as suas visitas. Desde os resumos de acordo com cada dia, como um resumo de trabalho referente ao ciclo de trabalho.

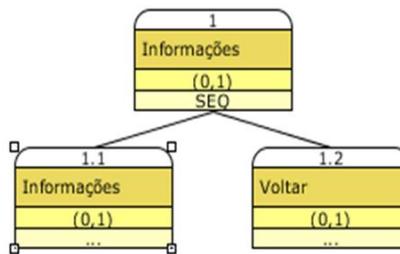
Figura 14 - Tarefa Arquivo



Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

Tarefa Informações sobre o Aplicativo: esta tarefa exibe as informações básicas sobre o aplicativo para o agente que estiver utilizando.

Figura 15 - Tarefa Informações Sobre o Aplicativo



Fonte: Elaborada pelo autor a partir do *software* ITAOS (2020).

3.3.2 User stories e teste de aceitação

Com base nas especificações feitas pelo cliente, foram definidas as *User Stories* do aplicativo. Para cada *User Stories* foram realizados testes de aceitação para que o sistema saísse da maneira exigida. No Quadro 4, a seguir, todos os User Stories e Teste de Aceitação.

Quadro 4 - User Stories e Teste de Aceitação

US.01	Implementação das funcionalidades da <i>activity</i> de acesso e segurança Tempo Estimado: 10h
TA1.1	Acessar o aplicativo através de um login válido (autenticação realizada com sucesso).
TA1.2	Acessar o aplicativo através de um <i>login</i> inválido (mensagem de erro de autenticação deve ser exibida).
US.02	Implementação das funcionalidades da <i>activity</i> de visualização da interface principal do aplicativo Tempo Estimado: 5h
TA2.1	Visualizar o acesso ao itinerário.
TA2.2	Visualizar o acesso as visitas pendentes.
TA2.3	Visualizar o acesso aos relatórios de visitas e trabalhos.
TA2.4	Visualizar o acesso as informações sobre o aplicativo.
US.03	Implementação das funcionalidades da <i>activity</i> da lista de Bairros Tempo Estimado: 5h
TA3.1	Cadastrar um bairro com todos os seus dados corretos (Bairro cadastrado com sucesso).
TA3.2	Cadastrar um bairro sem informar todos os seus dados obrigatórios (cadastro não deve ser efetuado).
TA3.3	Cadastrar um bairro informando dados inválidos (cadastro não deve ser efetuado).
TA3.4	Editar as informações de um bairro informando novos dados corretos (edição realizada com sucesso).
TA3.5	Editar as informações de um bairro informando dados inválidos (edição não deve ser efetuada).
TA3.6	Editar das informações de um bairro sem informar os dados obrigatórios (edição não deve ser efetuada).
TA3.7	Excluir o cadastro de um bairro clicando em cima e selecionando a opção confirmar (esse bairro não deve mais ser exibido).
TA3.8	Acessar a lista de bairros clicando em um bairro (uma relação de bairros deve ser exibida).
US.04	Implementação das funcionalidades da <i>activity</i> da lista de bairro Tempo Estimado: 5h
TA4.1	Cadastrar um bairro com todos os dados corretos (bairro cadastrado com sucesso).

TA4.2	Cadastrar um quarteirão sem informar os dados obrigatórios (cadastro não deve ser efetuado).
TA4.3	Cadastrar um quarteirão informando dados inválidos (cadastro não deve ser efetuado).
TA4.4	Edição das informações de um quarteirão informando novos dados corretos (edição realizada com sucesso).
TA4.5	Edição das informações de um quarteirão informando dados inválidos (edição não deve ser realizada).
TA4.6	Editar das informações de um quarteirão sem informar os dados obrigatórios (edição não deve ser efetuada).
TA4.7	Excluir cadastro de um quarteirão selecionado (esse quarteirão não deve mais ser exibido).
TA4.8	Acessar a lista de imóveis clicando em um quarteirão (uma relação de imóveis deve ser exibida).
US05	Implementação das funcionalidades da <i>activity</i> da lista de Imóvel Tempo Estimado: 5h
TA5.1	Cadastrar um imóvel com todos os dados corretos (cadastro realizado com sucesso).
TA5.2	Cadastrar um imóvel sem informar os dados obrigatórios (cadastro não deve ser efetuado).
TA5.3	Cadastrar um imóvel informando dados inválidos (cadastro não deve ser efetuado).
TA5.4	Editar as informações de um imóvel informando novos dados corretos (edição realizada com sucesso).
TA5.5	Editar as informações de um quarteirão sem informar os dados obrigatórios (edição não deve ser efetuada).
TA5.5	Editar as informações de um imóvel informando dados inválidos (edição não deve ser efetuada).
TA5.6	Excluir cadastro de um imóvel selecionado (esse imóvel não deve mais ser exibido).
TA5.7	Acessar o resumo diário de visitas clicando em um dos imóveis não visitados.
US06	Implementação das funcionalidades da <i>activity</i> do resumo diário de visitas Tempo Estimado: 10h
TA6.1	Marcar a opção “Fechado” (a visita não deve ser feita e as suas informações não poderão ser atualizadas).
TA6.2	Marcar a opção “Recusado” (a visita não deve ser feita e as informações não poderão ser atualizada).
TA6.3	Marcar a opção “Aberto” (realização da visita e coleta dos dados referentes ao imóvel visitado).
TA6.4	Atualizar o tipo da visita para “Normal” automaticamente.
TA6.5	Atualizar a data e a hora de visita automaticamente.
TA6.6	Informar a quantidade de depósitos do tipo “A1” inspecionados.

TA6.7	Informar a quantidade de depósitos do tipo “A2” inspecionados.
TA6.8	Informar a quantidade de depósitos do tipo “B1” inspecionados.
TA6.9	Informar a quantidade de depósitos do tipo “C1” inspecionados.
TA6.10	Informar a quantidade de depósitos do tipo “D1” inspecionados.
TA6.11	Informar a quantidade de depósitos do tipo “D2” inspecionados.
TA6.12	Informar a quantidade de depósitos do tipo “E1” inspecionados.
TA6.13	Informar a quantidade de depósitos eliminados.
TA6.14	Informar a quantidade de depósitos tratados.
TA6.15	Informar a quantidade de focos capturados.
TA6.16	Informar a quantidade de larvicida utilizado.
TA6.17	Seguir adiante com as visitas.
TA6.18	Fechar todas as visitas e encerrar os trabalhos (boletim fechado com sucesso).
US07	Implementação das funcionalidades da <i>activity</i> de pendência Tempo Estimado: 10h
TA7.1	Visualizar o acesso a lista de bairros.
TA7.2	Acessar a lista de quarteirões clicando em cima de um bairro.
TA7.3	Visualizar o acesso a lista de quarteirões.
TA7.4	Acessar a lista de imóveis clicando em cima de um quarteirão.
TA7.5	Visualizar o acesso a lista de imóveis.
TA7.6	Acessar o resumo diário de visitas clicando em cima de um imóvel não visitado.
US08	Implementação das funcionalidades da <i>activity</i> do resumo diário de visitas pendentes Tempo Estimado: 10h
TA8.1	Marcar a opção “Aberto” (a visita poderá ser feita e as informações referentes ao imóvel serão atualizadas).
TA8.2	Atualizar o tipo da visita para “Repasse” automaticamente.
TA8.3	Atualizar a data e a hora de visita automaticamente.
TA8.04	Informar a quantidade de depósitos do tipo “A1” inspecionados.

TA8.5	Informar a quantidade de depósitos do tipo “A2” inspecionados.
TA8.6	Informar a quantidade de depósitos do tipo “B1” inspecionados.
TA8.7	Informar a quantidade de depósitos do tipo “C1” inspecionados.
TA8.8	Informar a quantidade de depósitos do tipo “D1” inspecionados.
TA8.9	Informar a quantidade de depósitos do tipo “D2” inspecionados.
TA8.10	Informar a quantidade de depósitos do tipo “E1” inspecionados.
TA8.11	Informar a quantidade de depósitos eliminados.
TA8.12	Informar a quantidade de depósitos tratados.
TA8.13	Informar a quantidade de focos capturados.
TA8.14	Informar a quantidade de larvicida utilizado.
TA8.15	Seguir adiante com as visitas.
TA8.16	Fechar todas as visitas e encerrar os trabalhos (boletim fechado com sucesso).
US09	Implementação das funcionalidades da <i>activity</i> Fechamento do Boletim Tempo Estimado: 10h
TA9.1	Informar um login válido (o fechamento realizado com feito com sucesso).
TA9.2	Informar um login inválidos (fechamento não deve ser feito e uma mensagem de erro será exibida).
TA9.3	Exibir o boletim com todas as visitas normais realizadas no dia.
TA9.4	Exibir o boletim com todas as visitas pendentes realizadas no dia.
TA9.5	Armazenar os boletins no bando de dados do dispositivo.
TA9.6	Enviar os dados e as cópias dos boletins para um servidor.
US010	Implementação das funcionalidades da <i>activity</i> de Arquivos Tempo Estimado: 10h
TA10.1	Exibir uma relação de boletins de visitas normais.
TA10.2	Exibir uma relação de boletins de visitas pendentes.
TA10.3	Acessar relatório de trabalho clicando em “Relatório de Trabalho”.
TA10.3	Exibir relatório parcial de trabalho.

TA10.4	Realizar o fechamento do ciclo de trabalho informando um <i>login</i> válido (fechamento realizado com sucesso).
TA10.5	Realizar o fechamento do ciclo de trabalho informando um <i>login</i> inválidos (fechamento não deve ser feito e uma mensagem de erro será exibida).
TA10.6	Armazenar resumo de trabalho no banco de dados do dispositivo.
TA10.7	Enviar as informações e uma cópia do resumo de trabalho para um servidor.
US11	Implementação das funcionalidades de persistência para armazenamento das informações coletadas Tempo Estimado: 10h
TA11.1	Armazenar os dados coletados referentes a cada visita realizada.
TA11.2	Armazenar os resumos de visitas a cada dia de trabalho concluído.
US12	Implementação da funcionalidade para exibição das informações sobre o aplicativo Tempo Estimado: 5h
TA12.1	Exibir as informações referente ao aplicativo para o usuário.

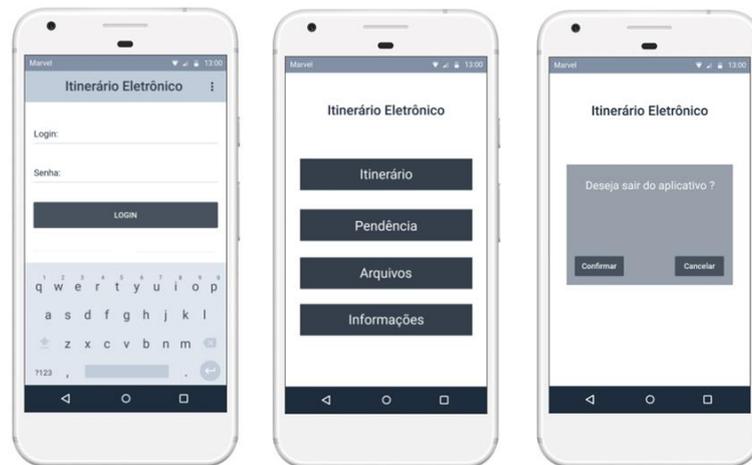
Fonte: Elaborado pelo autor (2020).

3.3.3 Protótipo da interface

Foi construído um protótipo da interface do aplicativo Itinerário Eletrônico que demonstra como ficará cada *activity* do sistema no momento de sua exibição para o usuário. Este protótipo explica de forma clara como o desenvolvedor deve implementar todas as telas do aplicativo.

Login e Apresentação: esta são as telas de acesso e segurança e a tela de apresentação do aplicativo. O usuário ao acessar o aplicativo terá de inserir um *login* e senha válido para ter acesso a tela principal e poder acessar as demais funcionalidades do aplicativo.

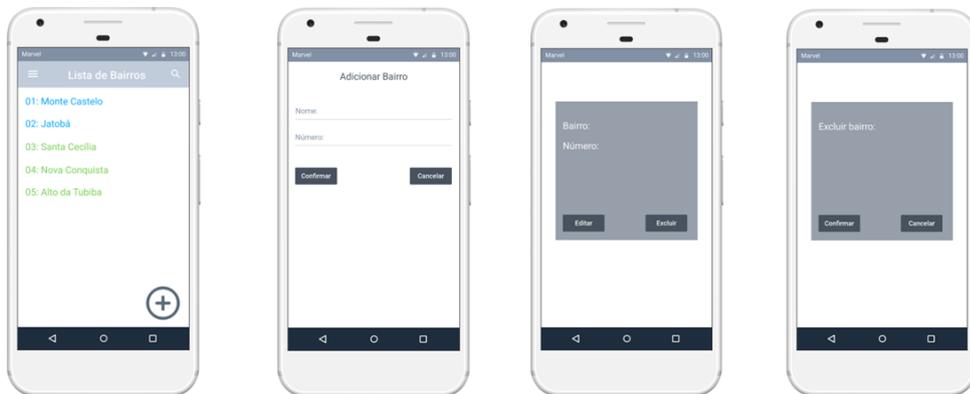
Figura 16 - Telas de Login e Apresentação



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

CRUD de Bairros: estas são as telas exibidas durante a manipulação das funcionalidades da lista de bairros, onde o usuário poderá adicionar um bairro, editar as suas informações e excluí-lo caso precise, além de poder acessar o conteúdo de cada bairro escolhido.

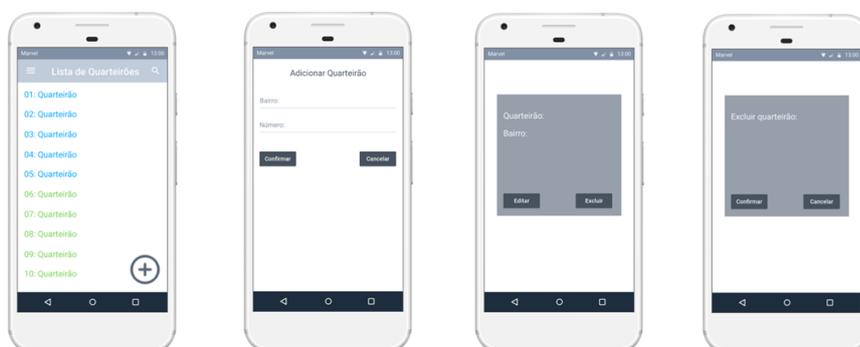
Figura 17 - Telas do CRUD de Bairros



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

CRUD de Quarteirões: estas são as telas exibidas durante a manipulação das funcionalidades da lista de quarteirões, onde o usuário poderá adicionar um quarteirão, editar as suas informações e excluí-lo caso precise, além de poder acessar o conteúdo de cada quarteirão escolhido.

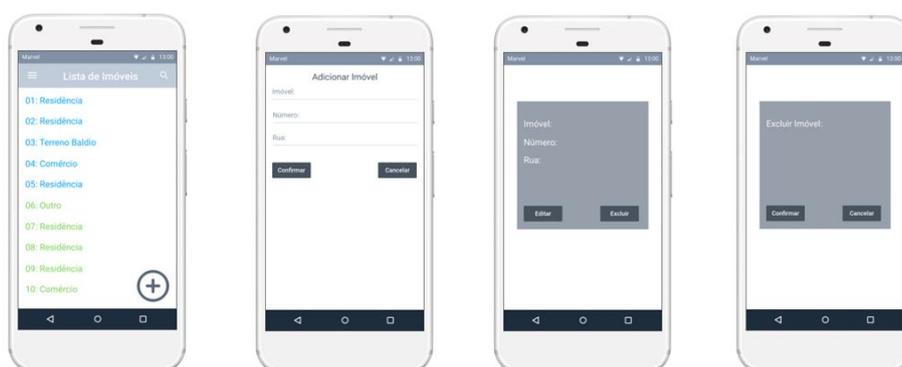
Figura 18 - Telas do CRUD de Quarteirões



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

CRUD de Imóveis: estas são as telas exibidas durante a manipulação das funcionalidades da lista de imóveis, onde o usuário poderá adicionar um imóvel, editar as suas informações e excluí-lo caso precise, além de poder acessar o conteúdo de cada imóvel escolhido.

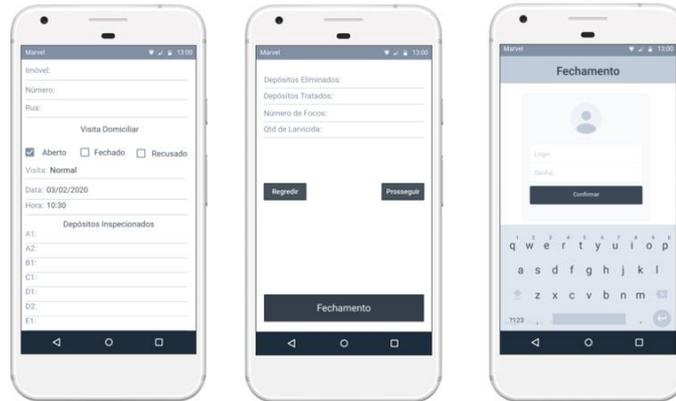
Figura 19 - Telas do CRUD de Imóveis



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

Resumo das Visitas Domiciliares: estas são as telas exibidas durante o acesso às funcionalidades do resumo de visitas domiciliares. Essas telas são exibidas quando o usuário acessa o conteúdo de cada imóvel das listas de imóveis e o mesmo poderá manipular todas as funções responsáveis pelas coletas de informações durante as visitas e também realizar o fechamento dos trabalhos a cada fim de expediente.

Figura 20 - Telas dos Resumos de Visitas Domiciliares



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

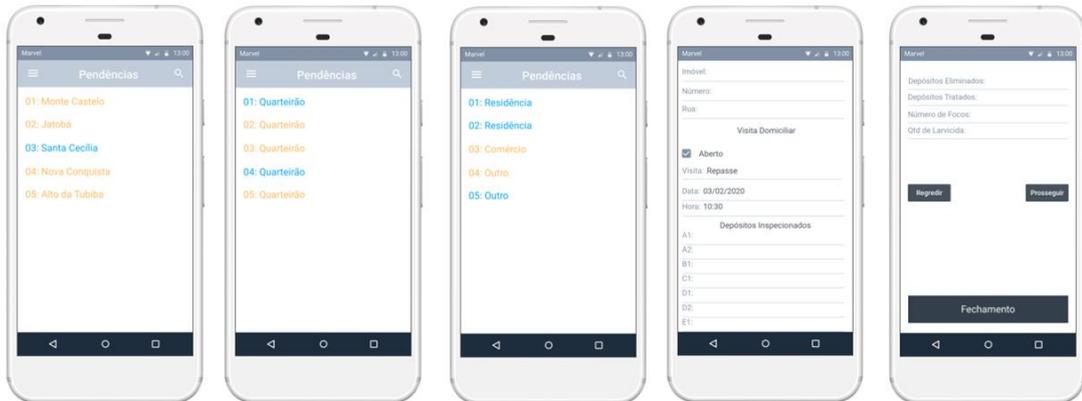
PDF de Visitas: após a conclusão das visitas domiciliares, o aplicativo realiza o fechamento dos trabalhos do dia e gera um *PDF* contendo todo o resumo de visitas feitas pelo agente durante o seu trabalho.

Figura 21 - PDF do Resumo de Visitas

Fonte: Brasil (2001).

Pendências: estas são as telas que serão exibidas quando o usuário acessar a opção Pendências. Nesta opção, o usuário poderá ter acesso aos imóveis pendentes de cada quarteirão ao qual eles pertencem e realizar a visita do mesmo sempre que houver a possibilidade. Toda vez que alguns desses imóveis forem visitados, ele automaticamente sairá da relação.

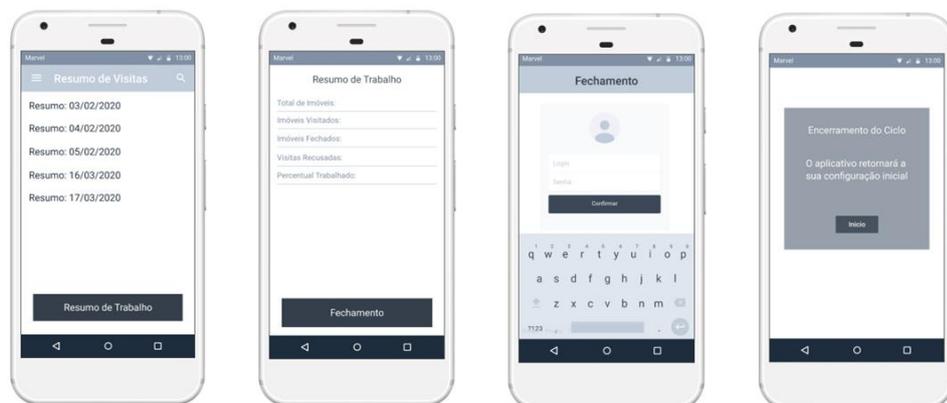
Figura 22 - Telas da Relação de Imóveis Pendente



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

Arquivos: estas são as telas exibidas quando for acessado a opção Arquivos. Aqui o usuário poderá ver a lista de resumos de visitas por datas e poderá acessar o resumo de trabalho contendo a porcentagem de imóveis trabalhados, pendentes e a opção que realiza o fechamento de todo o ciclo de trabalho.

Figura 23 - Telas da Relação de Arquivos do Aplicativo



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

Informações sobre o Aplicativo: esta é a tela que exibe todas as informações referente ao aplicativo Itinerário Eletrônico.

Figura 24 - Tela de Informações do Aplicativo



Fonte: Elaborada pelo autor a partir do *software online Marvelapp* (2020).

3.3.4 Projeto arquitetural

A ferramenta utilizada para o desenvolvimento do aplicativo será o *Android Studio*, um Ambiente de Desenvolvimento Integrado (IDE) oficial para o desenvolvimento de aplicativos para *Android*, que dispõe de editor de código, ferramentas de desenvolvedor avançadas do *IntelliJ* e outros recursos para aumentar a produtividade na compilação de aplicativos *Android* (DEVELOPER ANDROID, 2020).

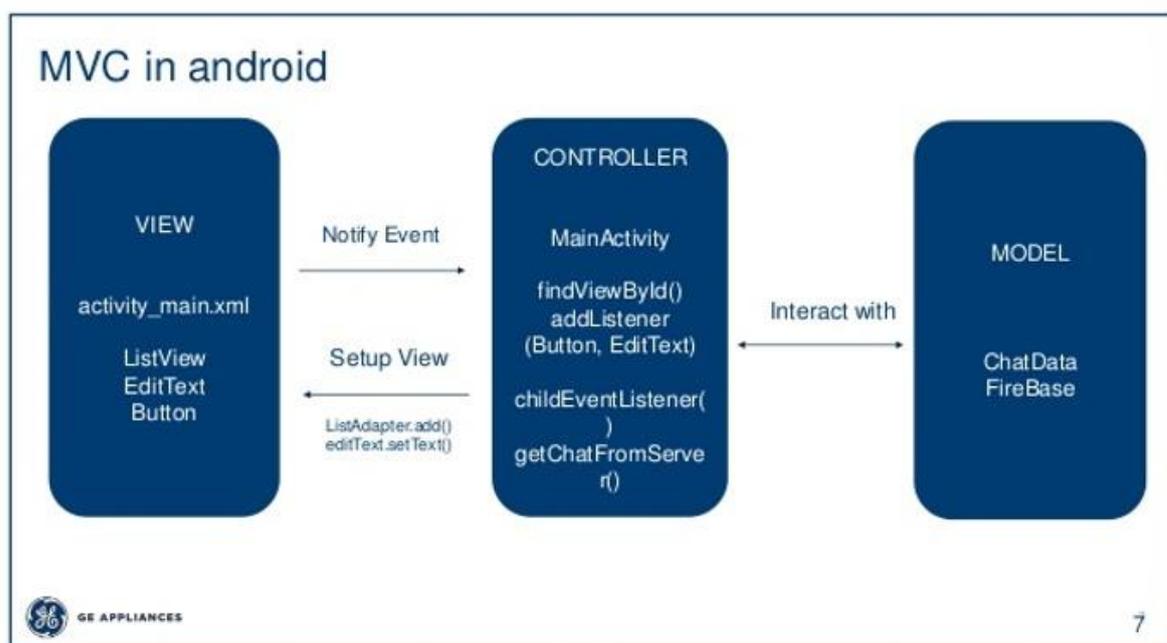
Todas as interfaces do aplicativo serão definidas em arquivos XML, as telas serão criadas a partir de uma classe que herda de um componente chamado *Activity*. *Java* é a linguagem de programação responsável pelas operações e métodos que serão realizados pelo aplicativo durante o seu uso. A ligação entre a *interface* em XML e seus elementos com o

código *Java* da *Activity* será feita através de um arquivo especial, com o nome apenas de *R.java* que fica na pasta *gen* (LECHETA, 2013).

O banco de dados será o *SQLite* que é o mais indicado para informações com a mesma estrutura e que podem ser organizadas em tabelas a serem consultadas. O *SQLite* é simples e consome poucos recursos, ele é bastante usado em dispositivos embarcados e dispõe de algumas bibliotecas, como o *Room* por exemplo (LECHETA, 2013).

A conexão *web* será feita através do *JSON* que é um formato mais utilizado de retorno de *web-services*, devido a sua simplicidade em comparação com *XML*. O *Android* possui bibliotecas padrão para lidar com o *JSON*. Existem duas classes que são utilizadas como o *JSONObject* e *JSONArray*. A *Retrofit* será a API utilizada na implementação da transmissão dos dados entre o aplicativo e o servidor (LECHETA, 2013).

Figura 25 - Arquitetura do Sistema



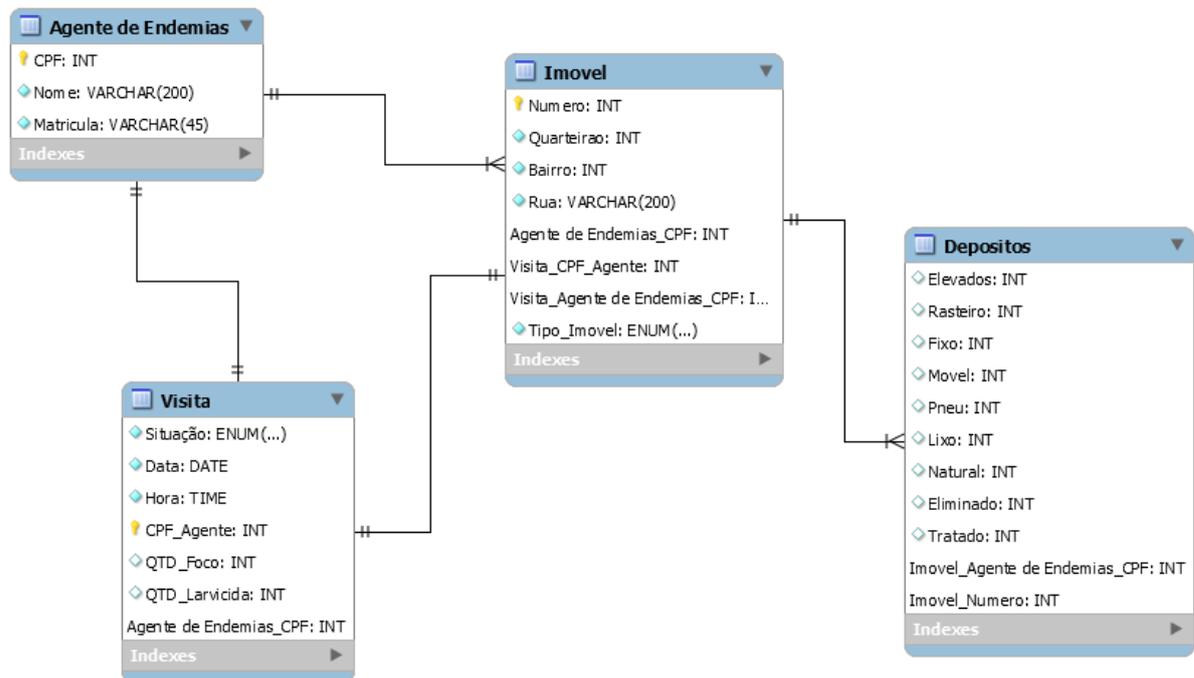
Fonte: Codementor (2018).

3.3.5 Modelo lógico de dados

Este modelo lógico de dados mostra todas as relações existentes no dia-dia de trabalho dos Agentes de Combate a Endemias, onde o agente realiza a visita programada de acordo com os dados do imóvel, como quarteirão e bairro, data e hora e a situação da visita, se o

imóvel está aberto, se está fechado ou se o residente recusa a visita. Se o imóvel estiver aberto, o agente irá fazer a inspeção de todos os depósitos, eliminar ou tratar alguns destes depósitos, coletar as informações e orientar o morador.

Figura 26 - Modelo Lógico de Dados do Aplicativo



Fonte: Elaborada pelo autor a partir do software MySQL Workbench (2020).

3.4 Planejamento (Release e Iteração)

Após a conclusão das atividades de inicialização, foi dado início a fase de planejamento do projeto. Com base no tempo que falta para apresentação, foram definidas etapas com tempo determinado afim de cumprir o prazo e finalizar a entrega do projeto.

3.4.1 Plano de Release

O plano de *release* consiste na definição dos *releases* e a partir de então dividi-los em iterações (GARCIA; LIMA; FERREIRA et al., 2007). A seguir, através dos Quadros 4, 5 e 6, serão mostrados alguns *releases* finalizados dentro dos seus devidos períodos.

Quadro 5 - Release 01 - (06/08/2019) Gerente: Raniery

Iteração	User Store	Período
Iteração - 01	US01	06/08/2019 – 29/08/2019

Fonte: Elaborado pelo autor (2019).

Quadro 6 - Release 02 - (05/09/2019) Gerente: Raniery

Iteração	User Store	Período
Iteração - 02	US02	29/08/2019 – 05/09/2019

Fonte: Elaborado pelo autor (2019).

Quadro 7 - Release 02 – (05/09/2019 – 12/09/2019) Gerente: Raniery

Iteração	User Store	Período
Iteração - 03	US03	05/09/2019 – 12/09/2019

Fonte: Elaborado pelo autor (2019).

3.4.2 Plano de iteração

O plano de iteração consiste em dividir as *User Stories* em atividades e torná-las responsabilidades de todos os membros da equipe. No caso deste projeto, apenas um dos membros é o responsável por todas as iterações do projeto.

Iteração – 01

US01 – Preparar o ambiente para uso da ferramenta de desenvolvimentos do aplicativo:

TA1.1	Pesquisar pela ferramenta de desenvolvimento Android Studio.
TA1.2	Pesquisar pelos principais requisitos de sistema.

Atividade	Descrição	Responsável	Estimativa de tempo (h)	Tempo Real	Status
A1.1	Baixar e instalar o JDK (Java Development Kit)	Raniery	2	2	Concluído
A1.2	Baixar, instalar e configurar o Android Studio	Raniery	2	2	Concluído

Fonte: Elaborado pelo autor (2020).

Iteração – 02

US02 – Implementar as funcionalidades da *activity* de acesso e segurança

TA2.1	Informar o login e a senha correto (acesso ao sistema confirmado com sucesso).
TA2.2	Informar login ou senha inválidos (o acesso ao sistema não pode ser efetuado e exibir uma mensagem de erro).

Atividade	Descrição	Responsável	Estimativa de tempo (h)	Tempo real	Status
A2.1	Implementar a	Raniery	3	3	Concluído

	<i>activity</i> de acesso e segurança				
A2.2	Implementar as funcionalidades responsável pelo de acesso e segurança	Raniery	6	7	Concluído

Fonte: Elaborado pelo autor (2020).

Iteração – 03

US03 – Implementação da *activity* de visualização do sistema

TA3.01	Visualizar o acesso ao itinerário.
TA3.02	Visualizar o acesso as visitas pendentes.
TA3.03	Visualizar o relatório de visitas e trabalhos.
TA3.04	Visualizar o acesso as informações do aplicativo.

Atividade	Descrição	Responsável	Estimativa de tempo (h)	Tempo real	Status
A3.1	Implementar a <i>activity</i> de visualização inicial	Raniery	1	2	Concluído
A3.2	Implementar as funcionalidades de acesso ao itinerário	Raniery	1	2	Concluído
A3.3	Implementar as funcionalidades de acesso as visitas pendentes	Raniery	1	2	Concluído
A3.4	Implementar as funcionalidades de acesso ao relatório de visitas e trabalhos	Raniery	1	2	Concluído
A3.5	Implementar o método de acesso as informações do aplicativo	Raniery	1	2	Concluído

Fonte: Elaborado pelo autor (2020).

Segue como apêndice os demais testes de aceitação e atividades relacionadas ao plano de iteração que foram previamente planejados, mas que não foram concluídos.

3.4.3 Big chart

O *big chart* pode ser visto como uma análise quantitativa do andamento do projeto a partir do recolhimento de métricas. As métricas são definidas pelo gerente de projeto, de acordo com os pontos principais a serem analisados (GARCIA; LIMA; FERREIRA et al., 2007).

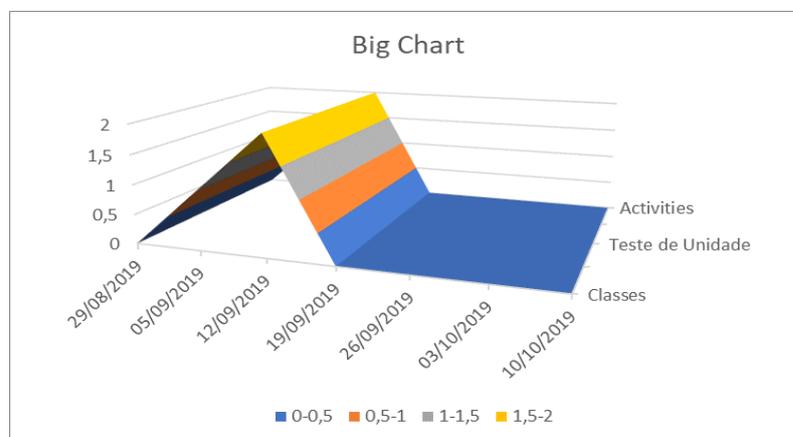
Quadro 8 - Big Chart

Big Chart					
Data	Classes	Teste de Aceitação	Teste de Unidade	User Stories	Activities
29/08/2019	0	0	0	0	0
05/09/2019	1	1	1	1	1
12/09/2019	2	2	2	2	2
19/09/2019	-	-	-	-	-
26/09/2019	-	-	-	-	-
03/10/2019	-	-	-	-	-
10/10/2019	-	-	-	-	-

Fonte: Elaborado pelo autor (2020).

Na figura 27 é exibido o Big Chart, um gráfico que mostra o progresso das etapas de desenvolvimento das Classes, Teste de Aceitação, Teste de Unidade, User Stories e Activities. De acordo com o gráfico, as etapas de desenvolvimento seguiam progredindo até a data 19/09/2019, quando sofreram uma estagnação.

Figura 27 - Big Chart



Fonte: Elaborada pelo autor a partir do Excel (2020)

3.4.4 *Análise de riscos*

Em 06 de agosto de 2019, foi iniciado o desenvolvimento do projeto, todas as tarefas foram organizadas estrategicamente em um gerenciador de projeto *online* chamado *Trello*, onde todos os passos referentes ao desenvolvimento do aplicativo foram devidamente registrados com a data de início e término de cada tarefa.

Primeiramente foi feito um estudo de todas as tecnologias que envolve o desenvolvimento do aplicativo, foram pesquisados assuntos envolvendo um dos principais sistemas operacionais para dispositivos móveis, o *Android*, a ideia era saber tudo sobre essa plataforma e suas principais características com o objetivo de entender esse sistema operacional, as suas características e suas relações com todos os aplicativos para ele desenvolvido.

Em seguida foi feito um estudo envolvendo os principais tópicos especiais para desenvolvimento de aplicações para dispositivos móveis, depois de estudar tudo sobre o sistema operacional *Android*, a ideia é aprender tudo sobre as principais técnicas de programação para essas plataformas.

Para concluir a fase de estudos das tecnologias foi feito um estudo sobre a principal ferramenta utilizada por desenvolvedores para projetar o aplicativo, o ambiente de desenvolvimento *Android Studio*, essa ferramenta é extremamente sofisticada para quem deseja desenvolver aplicativos para o sistema *Android*. Foram realizados estudos sobre todos os seus componentes e sobre os requisitos para a instalação.

Contudo, o aprendizado das tecnologias demanda um tempo considerável e o prazo para concluir o projeto era algo em torno de dois meses, mas tudo foi previamente organizado para ser concluído dentro do prazo estipulado, desta forma o projeto foi iniciado.

No dia 19 de agosto de 2019, logo após os estudos das tecnologias, foram feitos os levantamentos de todos os requisitos para obter todas as informações sobre os profissionais Agentes de Combates a Endemias, tudo sobre o programa de controle do mosquito *Aedes Aegypti* e a logística em torno da organização responsável por toda a Vigilância Ambiental da cidade de Patos-PB. O objetivo era entender como os profissionais realizavam a coletas das informações, onde as guardava e para onde elas eram enviadas, pois o aplicativo iria automatizar todas essas tarefas para torná-las mais rápida e dinâmica. O prazo para conclusão do aplicativo era o dia 10 de outubro de 2019.

Em seguida foi criado um diagrama de classe explicando toda a dinâmica do trabalho dos ACEs, demonstrando como cada profissional fazia o seu trabalho em cada visita feita,

apresentando todos os detalhes. Isso auxilia o desenvolvedor na hora de criar as classes e rotinas que deverá ser feita pelo aplicativo.

No dia 29 de agosto de 2019, depois de estudar sobre as tecnologias, com todos os requisitos e informações adquiridas e diagrama de classe pronto, era hora de começar a desenvolver o aplicativo, de acordo com a logística elaborada, o aplicativo irá ser feito por partes, a primeira parte de aplicativo a ser feita será a *activity* de autenticação, o prazo para concluir esta tarefa é o dia 05 de setembro de 2019.

Essa *activity* levou mais de uma semana para ser feita, mais tempo do que previsto, devido alguns problemas envolvendo a equipe responsável pela parte técnica, a equipe era composta por apenas duas pessoas, mas apenas uma realizava todas as tarefas atribuídas a uma equipe de desenvolvimento de *software*, o outro era um consultor técnico, responsável em fornecer informações e conselhos detalhados ao desenvolver, o fato de apenas uma única pessoa ser a responsável por todo o desenvolvimento do sistema, causava sobrecarga e comprometia a organização do trabalho.

Outro problema que vinha ocorrendo era a indisponibilidade de tempo de um dos componentes da equipe de desenvolvimento, o consultor, ele é um profissional experiente com um amplo conhecimento na área de desenvolvimento de aplicativos para dispositivos móveis, mas o seu tempo era muito corrido durante a semana, impossibilitando consultas diárias, as consultorias só poderiam ser feitas de formas semanais e um curto espaço de tempo que na maioria das vezes não sanava todas as dúvidas.

Nesse ritmo iniciou a produção da próxima tarefa, a *Activity* de apresentação do aplicativo, essa é a *interface* responsável por apresentar o sistema ao usuário com os acessos ao itinerário, relação de visitas pendentes, arquivos com a relação de resumos de visitas domiciliares realizadas e relatório parcial de trabalho e as informações referentes ao aplicativo. Essa *Activity* é bem mais simples e fácil de ser feita e levou um tempo menor do que estimado.

A próxima tarefa é a *Activity* de acesso a lista de bairro, bairros e imóveis, essa *interface* é bem mais complicada pois envolve algumas funções de manipulação dessas listas como adição, pesquisa, edição e exclusão de alguns desses itens e também é a responsável por coletar as informações e realizar os cálculos a cada conclusão dos trabalhos.

Por ser uma *interface* mais difícil que as demais, a sua conclusão foi por diversas vezes adiada pela ocorrência de inúmeros erros e seu desenvolvimento e obrigando o desenvolvedor a ter que realizar várias consultas e estudos, mas a trabalho não avançava.

Além do desconhecimento da tecnologia e da limitação da equipe, um novo risco começou a se manifestar, o responsável pelo desenvolvimento do projeto começou a apresentar problemas de saúde que acabou comprometendo o desenvolvimento do trabalho, esses problemas de saúde iniciaram na metade do mês de setembro e foram se agravando no decorrer do prazo para conclusão do aplicativo.

Com isso o prazo para conclusão do projeto foi adiado até a data limite para apresentação do projeto, fazendo com o que a conclusão chegasse na data limite de entrega, em um dado momento, o programador não pôde mais prosseguir com o trabalho devido um agravamento do seu problema de saúde e tendo que abandonar o trabalho. Dessa maneira, todo o planejamento foi comprometido e o desenvolvimento do aplicativo teve de ser interrompido por prazo indeterminado tendo a sua conclusão e apresentação adiada.

No primeiro semestre de 2020, houve uma tentativa de concluir o projeto, mas os problemas anteriores ainda persistiam, principalmente com a relação à saúde do desenvolvedor que acabou sendo diagnosticado com um problema crônico renal do qual tem que se submeter a seções de hemodiálise três vezes por semana.

Somando a esses problemas mencionados, um outro acabara de surgir, no final do ano de 2019 passado e início do ano de 2020, uma pandemia começou a se alastrar de formar assustadora em todo mundo, causando um verdadeiro caos sanitário e econômico, fazendo inúmeras vítimas e parando o planeta por completo. Com todos esses riscos, a equipe decidiu então interromper definitivamente a produção do projeto e apresentar o que até então tinha sido produzido, cuja disposição destes riscos pode ser melhor entendida pela síntese presente no Quadro 8 a seguir:

Quadro 9 - Tabela de Riscos

Data	Risco	Prioridade	Responsável	Status	Providência/Solução
06/08/2019	Desconhecimento da Tecnologia	Média	Raniery	Superado	Estudar um pouco mais sobre as tecnologias envolvidas
29/08/2019	Limitação da Equipe	Alta	Raniery	Vigente	Aproveitar a disponibilidade de tempo possível de cada membro
10/10/2019	Ruptura do Desenvolvimento em virtude de indisponibilidade de Recurso Humanos (Colapso)	Alta	Raniery	Vigente	Suspender por indeterminado a conclusão do projeto
17/03/2020	Pandemia	Alta	Raniery	Vigente	Interromper a realização do projeto e apresentar o que foi feito até momento

Fonte: Elaborado pelo autor (2020).

3.5 Considerações Finais

Neste capítulo, foi apresentado a proposição arquitetural para desenvolvimento do aplicativo Itinerário Eletrônico, fazendo uso da metodologia de desenvolvimento, o easYProcess (YP) seguindo as principais etapas: definição de papéis, conversas com o cliente, inicialização e planejamento para uma posterior implementação do aplicativo. A seguir a Conclusão do trabalho com as considerações finais, contribuições da pesquisa, limitações da pesquisa e trabalhos futuros.

4 CONCLUSÃO

O presente capítulo apresenta as considerações finais do trabalho, seguindo das contribuições e limitações desta pesquisa, bem como sugestões para trabalhos futuros.

4.1 Considerações Finais

Este trabalho apresentou uma proposição arquitetural para o desenvolvimento de um aplicativo chamado Itinerário Eletrônico, que é um sistema para dispositivos móveis com o sistema operacional *Android*, que objetiva automatizar a coleta, a análise, o processamento, o armazenamento, exibição e envio via *web* das informações obtidas pelos Agentes de Combate a Endemias da cidade de Patos na Paraíba, bem como facilitar o trabalho da coordenação responsável pelo trabalho destes profissionais.

Foi uma experiência bem-sucedida de projeto, seguida de uma experiência má sucedida de produto com ênfase na gestão de risco, porém, com a entrega/proposição de um projeto arquitetural capaz de ser disponibilizado ao setor de TI da Secretaria Municipal de Saúde de Patos, com vista a sua futura implementação e implantação como forma de utilidade pública gratuitamente prestada.

4.2 Contribuições da Pesquisa

Esta pesquisa contribui para a proposição de sistema que irá dinamizar os trabalhos de coletas de dados feitas pelos ACEs da cidade de Patos-PB, explicando com detalhes, todas as especificidades principais do aplicativo bem como apresentando todas as etapas e sugestões necessárias como organização e definição dos papéis da equipe e os planejamentos e medidas preventivas para os eventuais riscos que possam ocorrer durante a etapa de desenvolvimento.

4.3 Limitações da Pesquisa

A pesquisa foi marcada por algumas limitações, o desconhecimento das tecnologias envolvidas impôs que um dos membros tivesse que estudar sobre o sistema operacional *Android*, sobre a ferramenta de desenvolvimento de aplicativos *Android Studio* e sobre as

principais práticas de desenvolvimentos de aplicações móveis. A limitação da equipe composta por dois membros onde um deles auxiliava e outro desenvolvia seguindo as orientações do primeiro fez com que o desenvolvimento tivesse pequenos atrasos. E a ruptura do desenvolvimento por conta da indisponibilidade dos recursos humanos em virtude de um problema de saúde adquirido por um dos membros, fizeram com que o trabalho sofresse várias interrupções, obrigando a equipe a mudar a sua estratégia de finalização do trabalho.

4.4 Trabalhos Futuros

Entre diversas possibilidades, pode-se destacar:

- A possibilidade de transformação deste trabalho em um produto final capaz de corresponder às expectativas do trabalho e colocar em prática o desenvolvimento aplicativo Itinerário Eletrônico.
- Desenvolver uma versão *web* do Itinerário Eletrônico que se comunique com todas as versões *Mobile* instaladas em todos os dispositivos utilizados por cada ACEs para que as informações possam ser acompanhadas em tempo real pela gestão da coordenação de Vigilâncias Ambiental da cidade de Patos-PB e gerando um relatório geral com todas as informações de todos os Agentes de Combate a Endemias.
- Propor uma outra versão *Mobile* para outras plataformas, não só a *Android* como também desenvolver uma versão final que seja capaz de ser implantada em outros municípios, não somente Patos, com o objetivo de expandir a tecnologia.

REFERÊNCIAS

BRASIL. MINISTÉRIO DA SAÚDE. FUNDAÇÃO NACIONAL DE SAÚDE. **Dengue instruções para o pessoal de combate ao vetor: manual de normas técnicas**. 3. ed., rev. Brasília: Ministério da Saúde: Fundação Nacional de Saúde, 2001. 84p.

BRASIL. MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE. DEPARTAMENTO DE VIGILÂNCIA DAS DOENÇAS TRANSMISSÍVEIS. **Plano de Contingência Nacional para a Febre de Chikungunya**. Brasília: Ministério da Saúde, 2014. 48 p.

BRASIL. MINISTÉRIO DA SAÚDE. SECRETARIA DE VIGILÂNCIA EM SAÚDE. DEPARTAMENTO DE VIGILÂNCIA DE DOENÇAS E AGRAVOS NÃO TRANSMISSÍVEIS E PROMOÇÃO DA SAÚDE. **Saúde Brasil 2014: uma análise da situação de saúde e das causas externas**. Brasília: Ministério da Saúde, 2015. 462 p.

CARVALHO, L. Android cresce no Brasil e aumenta a distância para iOS e Windows Phone. **Olhar Digital**: 2017. Disponível em < <https://olhardigital.com.br/noticia/android-cresce-no-brasil-e-aumenta-distancia-para-ios-e-windows-phone/68023>>. Acesso em: 14 nov. 2018.

CASTRO, R. **Habitação, Saneamento Básico e a proliferação de Dengue, Zika e Chikungunya nas Favelas**. Fundação Oswaldo Cruz: 2016. Disponível em: <https://rededengue.fiocruz.br/noticias/524-habitacao-saneamento-basico-e-a-proliferacao-de-dengue-zika-e-chikungunya-nas-favelas?fbclid=IwAR0Dw3BcsJC42qWBAVtwd33hCce06MEq6porFsMUTg_zmmhrDWYP3zA9jTw>. Acesso em: 14 nov. 2018.

CROCKFORD, D. Introducing JSON. **Json.org**: 2018. Disponível em: <<https://www.json.org/>>. Acesso em: 14 nov. 2018.

FERREIRA, C.F.; MOTA, R.D. **Comparando Aplicação Web Service REST e SOAP**. Unipar: 2014. Disponível em: <[http://web.unipar.br/~seinpar/2014/artigos/pos/Cleber_de_F_Ferreira_Roberto_Dias_Mota%20\(1\).pdf?fbclid=IwAR0kchGRjttUoNUnBjDwd78geT-r802IIAog-enYpzuSJ9IWOApbUMIXeY4](http://web.unipar.br/~seinpar/2014/artigos/pos/Cleber_de_F_Ferreira_Roberto_Dias_Mota%20(1).pdf?fbclid=IwAR0kchGRjttUoNUnBjDwd78geT-r802IIAog-enYpzuSJ9IWOApbUMIXeY4)>. Acesso em: 14 nov. 2018.

FRÓES, M. **Utilizando Retrofit no Android**. 2016. Disponível em: <https://matheusfroes.wordpress.com/2016/03/27/utilizando-retrofit-no-android/?fbclid=IwAR1K3yKKOe-EMxBzBKUBPo9fwJGzbBFfeARcfRfiJWRree_JKkViOILQpAOA>. Acesso em: 14 nov. 2018.

G1 NOTÍCIAS. Brasil tem 62 mortes por dengue em 2018. **G1.globo.com**: 2018. Disponível em: <<https://g1.globo.com/bemestar/noticia/brasil-tem-62-mortes-por-dengue-em-2018.ghtml>>. Acesso em: 14 nov. 2018.

LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3 ed. São Paulo: Novatec, 2013. 824 p.

MOURA, A. S.; ROCHA, R. L. **Endemias e Epidemias: dengue, leishmaniose, febre amarela, influenza, febre maculosa e leptospirose**. Belo Horizonte: Nescon/ufmg, 2012. 78 p.

SILVA, L. J. O Controle das Endemias no Brasil e suas histórias. **Revista Ciência e Cultura**, v. 55, n. 1, p. 44-47, 2003.

TELE SÍNTESE. Há 5 Bilhões Assinantes de Celular no Mundo. **Portal Terra**: 2017. Disponível em: <<http://www.telesintese.com.br/ha-5-bilhoes-assinantes-de-celular-no-mundo/>>. Acesso em: 14 nov. 2018.

TOLEDO JÚNIOR, A. C. C. **Pragas e Epidemias: histórias de doenças infecciosas**. Belo Horizonte: Folium Editora, 2006. 152 p.

LUNA, Expedito J. A.; SILVA JUNIOR, Jarbas Barbosa da. Doenças transmissíveis, endemias, epidemias e pandemias. In: FUNDAÇÃO OSVALDO CRUZ (Brasil). Secretaria de Assuntos Estratégicos da Presidência da República. **A saúde no Brasil em 2030: prospecção estratégica do sistema de saúde brasileiro: organização e gestão do sistema de saúde**. Rio de Janeiro: Fiocruz/ipea/ministério da Saúde/secretaria de Assuntos Estratégicos da Presidência da República, 2013. Cap. 4. P. 126-176. ISBN 978-85-8110-016-6. Disponível em: <<http://books.scielo.org/id/8pmm/pdf/noronha-9788581100166-06.pdf>>. Acesso em: 30 de out. de 2018

DENGUE, vírus e vetor. Fiocruz 2018. Disponível em: <<http://www.ioc.fiocruz.br/dengue/textos/longatraje.html>> Acesso em 30 de out de 2018

TOSIN, C. Conhecendo o Android. **Dicas-L**: 2011. Disponível em: <http://www.dicas-l.com.br/arquivo/conhecendo_o_android.php?fbclid=IwAR0_bxbIXk87DbEpci1CgJP35VRbr1VT5kpy4us381NoHr1QmFj1JvL8IWk#.W-xQcDhKjIW>. Acesso em: 14 nov. 2018.

VASCONCELOS, P. F. C. Febre Amarela. **Revista da Sociedade Brasileira de Medicina Tropical**, n. 36, v. 2, p. 275-293, mar-abr, 2003.

Mais de 5 bilhões de pessoas usam aparelho celular, revela pesquisa. **Agencia Brasil**, Brasília, 08 de setembro de 2019. Disponível em:<<https://agenciabrasil.ebc.com.br/geral/noticia/2019-09/mais-de-5-bilhoes-de-pessoas-usam-aparelho-celular-revela-pesquisa>>. Acesso em: 11 de julho de 2020

A evolução da tecnologia mobile e seu reflexo para a área de RH. **LG**, 2019 Disponível:<<https://www.lg.com.br/blog/evolucao-tecnologia-mobile-rh/>>. Acesso em: 25 de julho de 2020

Conheça o Android Studio. **Developers**, 2020 Disponível: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em: 01 de agosto de 2020

GARCIA, F. P.; LIMA, A. H. G.; FERREIRA, D. de S.; JÚNIOR, F. L. L.; ROCHA, G. R. C. da; MENDES, G. W. D.; PONTES, R. F. de; ROCHA, V. K. da H.; DANTAS, V. F.

easYProcess – Um Processo de Desenvolvimento de Software. Universidade Federal de Campina Grande. Campina Grande: 2007.

Arquitetura da Plataforma. **Developers**, 2020 Disponível: <https://developer.android.com/guide/platform?hl=pt-br>. Acesso em 14 de dez de 2020.

Desenvolvimento de App: escolhendo a linguagem de Programação. **Tecmundo**. 2020 Disponível em: <https://www.tecmundo.com.br/software/204629-desenvolvimento-app-escolhendo-linguagem-programacao.htm>. Acesso em 14 de dez de 2020.

Guia de Flutter. **Devmedia**. 2020 Disponível: <https://www.devmedia.com.br/guia/flutter/40713>. Acesso em 14 de dez de 2020.

React Native. **Devmedia**. 2020 Disponível: <https://www.devmedia.com.br/react-native/>. Acesso em 14 de dez de 2020.

Web service REST versus SOAP. **Devmedia**. 2020 Disponível: <https://www.devmedia.com.br/web-services-rest-versus-soap/32451>. Acesso em 14 de dez de 2020.

Android Retrofit: primeiros passos com retrofit API. **Devmedia**. 2020 Disponível: <https://www.devmedia.com.br/android-retrofit-primeiros-passos-com-a-retrofit-api/31857>. Acesso em 14 de dez de 2020.

APÊNDICE A – Iterações do Projeto Que Não Foram Concluídas

Iteração – 04

US04 – Implementação da activity lista de Bairros:

TA4.1	Visualizar da lista de Bairros.
TA4.2	Cadastrar bairro com todos dados corretos (bairro cadastrado com sucesso).
TA4.3	Cadastrar bairro sem informar os dados obrigatórios (o aplicativo deve informar que o cadastro não deve ser efetuado).
TA4.4	Cadastrar bairro com os dados inválidos (o aplicativo deve informar que o cadastro não deve ser efetuado).
TA4.5	Editar o cadastro de um bairro informando novos dados corretos (edição efetuada com sucesso).
TA4.6	Editar o cadastro de um bairro sem informar os dados obrigatórios (o aplicativo deve informar que a edição não deve ser efetuada).
TA4.7	Editar o cadastro de um bairro informando os dados inválidos (a edição não deve ser efetuada).
TA4.8	Excluir um bairro de relação clicando na opção excluir e depois confirmar (o bairro não deve ser mais exibido).
TA4.9	Acessar a lista de bairros clicando em cima do bairro selecionado.

Atividade	Descrição	Responsável	Estimativa de Tempo (h)	Tempo Real	Status
A4.1	Implementar a activity de acesso a lista de bairros	Raniery	5	-	-
A4.2	Implementar o CRUD de bairros	Raniery	5	-	-
A4.3	Implementar as funcionalidades de acesso a	Raniery	5	-	-

	lista de quarteirões				
--	-------------------------	--	--	--	--

Iteração – 05

US05 – Implementação da activity lista de Quarteirões:

TA5.1	Visualizar a lista de quarteirões.
TA5.2	Cadastrar quarteirão a lista com todos os dados corretos (cadastro de quarteirão realizado com sucesso).
TA5.3	Cadastrar quarteirão a lista sem informar os dados obrigatórios (o cadastro não deve ser efetuado pelo aplicativo).
TA5.4	Cadastrar quarteirão a lista informando dados inválidos (o cadastro não deve ser efetuado pelo aplicativo).
TA5.5	Editar o cadastro de um quarteirão informando novos dados corretos (edição efetuado com sucesso).
TA5.6	Editar o cadastro de um quarteirão sem informar os dados obrigatórios (edição não deve ser efetuada pelo aplicativo).
TA5.7	Editar o cadastro de um quarteirão informando dados inválidos (a edição não deve ser efetuada pelo aplicativo).
TA5.8	Excluir um quarteirão de relação clicando na opção excluir e depois confirmar (o quarteirão não deve ser mais exibido).
TA5.9	Acessar a relação de imóveis clicando em cima do quarteirão selecionado.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A5.1	Implementar a activity de acesso a lista de quarteirões	Raniery	5	-	-

A5.2	Implementar o CRUD de Quarteirões	Raniery	5	-	-
A5.3	Implementar as funcionalidades de acesso a lista de imóveis	Raniery	5	-	-

Iteração – 06

US06 – Implementação da activity relação de Imóveis:

TA6.1	Visualizar a relação de imóveis.
TA6.2	Cadastrar imóvel a relação com todos os dados corretos (cadastro de imóvel realizado com sucesso).
TA6.3	Cadastrar imóvel sem informar os dados obrigatórios (o aplicativo deve informar que o cadastro não deve ser efetuado).
TA6.4	Cadastrar imóvel a relação informando dados inválidos (o cadastro não deve ser efetuado pelo aplicativo).
TA6.5	Editar o cadastro de um imóvel informando novos dados corretos (edição efetuado com sucesso).
TA6.6	Editar o cadastro de um imóvel sem informar os dados obrigatórios (edição não deve ser efetuada pelo aplicativo).
TA6.7	Editar o cadastro de um imóvel informando dados inválidos (a edição não deve ser efetuada pelo aplicativo).
TA6.8	Excluir um imóvel de relação clicando na opção excluir e depois confirmar (o imóvel não deve ser mais exibido).
TA6.9	Acessar o resumo diário de visitas clicando em cima do imóvel selecionado.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
------------------	------------------	--------------------	----------------------------	-------------------	---------------

A6.1	Implementar a activity de acesso a relação de imóveis	Raniery	5	-	-
A6.2	Implementar o CRUD de Imóveis	Raniery	5	-	-
A6.3	Implementar as funcionalidades de acesso ao resumo diário de visitas	Raniery	5	-	-

Iteração – 07

US07 – Implementação das funcionalidades da activity do resumo diário de visita:

TA7.01	Marcar a opção “Fechado” (a visita não poderá ser feita e as suas informações não poderão ser atualizadas).
TA7.02	Marcar a opção “Recusado” (a visita não poderá ser feita e as informações não poderão ser atualizada).
TA7.03	Marcar a opção “Aberto” (a visita poderá ser feita e as informações referentes ao imóvel serão atualizadas).
TA7.04	Atualizar o tipo da visita para “Normal” automaticamente.
TA7.05	Atualizar a data e a hora de visita automaticamente.
TA7.06	Informar a quantidade de depósitos do tipo “A1” inspecionados.
TA7.07	Informar a quantidade de depósitos do tipo “A2” inspecionados.
TA7.08	Informar a quantidade de depósitos do tipo “B1” inspecionados.
TA7.09	Informar a quantidade de depósitos do tipo “C1” inspecionados.
TA7.10	Informar a quantidade de depósitos do tipo “D1” inspecionados.
TA7.11	Informar a quantidade de depósitos do tipo “D2” inspecionados.
TA7.12	Informar a quantidade de depósitos do tipo “E1” inspecionados.
TA7.13	Informar a quantidade de depósitos eliminados.
TA7.14	Informar a quantidade de depósitos tratados.

TA7.15	Informar a quantidade de focos capturados.
TA7.16	Informar a quantidade de larvicida utilizado.
TA7.17	Seguir adiante com as visitas.
TA7.18	Fechar todas as visitas e encerrar os trabalhos (o boletim foi fechado com sucesso).
TA7.19	Selecionar os imóveis com as opções de visitas "Fechado" ou "Recusado" e adicionar a relação de imóveis com visitas pendentes.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A7.1	Implementar a activity de acesso ao resumo diário de visitas	Raniery	5	-	-
A7.2	Implementar as funcionalidades de coleta, fechamento e armazenamento dos dados obtidos durante cada visita domiciliar	Raniery	5	-	-

Iteração – 08

US08 – Implementação das funcionalidades da activity de Pendência:

TA8.1	Visualizar a relação de bairros.
TA8.2	Acessar a relação de quarteirões ao clicar em um bairro.
TA8.3	Visualizar a relação de quarteirões.
TA8.4	Acessar a relação de imóveis ao clicar em um quarteirão.

TA8.5	Visualizar a relação da imóveis pendentes.
TA8.6	Acessar o resumo diário de visitas domiciliares pendentes ao clicando em um imóvel não visitado.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A8.1	Implementar a activity de acesso a lista de bairros com imóveis pendentes	Raniery	5	-	-
A8.2	Implementar as funcionalidades de acesso a lista de quarteirões com imóveis pendentes	Raniery	5	-	-
A8.3	Implementar a activity de acesso a lista de quarteirões com imóveis pendentes	Raniery	5	-	-
A8.4	Implementar as funcionalidades de acesso a lista de imóveis pendentes	Raniery	5	-	-
A8.5	Implementar a activity de acesso a lista de imóveis pendentes	Raniery	5	-	-
A8.6	Implementar as funcionalidades de acesso ao	Raniery	5	-	-

	resumo diário de visitas				
--	--------------------------	--	--	--	--

Iteração – 09

US09 – Implementação das funcionalidades da Activity do resumo diário de visitas pendentes:

TA9.01	Marcar a opção “Aberto” (a visita poderá ser feita e as informações referentes ao imóvel serão atualizadas).
TA9.02	Atualizar o tipo da visita para “Repasse” automaticamente.
TA9.03	Atualizar a data e a hora de visita automaticamente.
TA9.04	Informar a quantidade de depósitos do tipo “A1” inspecionados.
TA9.05	Informar a quantidade de depósitos do tipo “A2” inspecionados.
TA9.06	Informar a quantidade de depósitos do tipo “B1” inspecionados.
TA9.07	Informar a quantidade de depósitos do tipo “C1” inspecionados.
TA9.08	Informar a quantidade de depósitos do tipo “D1” inspecionados.
TA9.09	Informar a quantidade de depósitos do tipo “D2” inspecionados.
TA9.10	Informar a quantidade de depósitos do tipo “E1” inspecionados.
TA9.11	Informar a quantidade de depósitos eliminados.
TA9.12	Informar a quantidade de depósitos tratados.
TA9.13	Informar a quantidade de focos capturados.
TA9.14	Informar a quantidade de larvicida utilizado.
TA9.15	Seguir adiante com as visitas.
TA9.16	Fechar todas as visitas e encerrar os trabalhos (o boletim foi fechado com sucesso).

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
------------------	------------------	--------------------	----------------------------	-------------------	---------------

A9.1	Implementar a activity de acesso ao resumo diário de visitas pendentes	Raniery	5	-	-
A9.2	Implementar as funcionalidades de coleta, fechamento e armazenamento dos dados obtidos durante cada visita aos imóveis pendentes	Raniery	5	-	-

Iteração – 10

US10 – Implementação das funcionalidades da activity Arquivo:

TA10.1	Exibir uma relação de resumo de visitas normais.
TA10.2	Exibir uma relação de resumo de visitas pendentes.
TA10.3	Acessar resumo parcial de trabalho ao clicar em “Parcial de Trabalho”.
TA10.4	Exibir resumo parcial de trabalho.
TA10.5	Realizar o fechamento do ciclo de trabalho informando um login válido (fechamento realizado com sucesso).
TA10.6	Realizar o fechamento do ciclo de trabalho informando um login inválidos (fechamento não deve ser feito e uma mensagem de erro será exibida).
TA10.7	Armazenar o relatório de trabalho no banco de dados do dispositivo.
TA10.8	Enviar as informações e uma cópia do resumo de trabalho para um servidor.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A10.1	Implementar a activity de acesso ao	Raniery	5	-	-

	Arquivos de visitas e trabalhos				
A10.2	Implementar as funcionalidades de acesso ao Arquivos e de fechamento, armazenamento e envio	Raniery	5	-	-

Iteração – 11

US11 – Implementação das funcionalidades da activity de visualização das informações sobre o aplicativo:

TA11.1	Visualização das informações referentes ao aplicativo.
TA11.2	Voltar para a tela inicial do aplicativo.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A11.1	Implementar a activity de visualização das informações sobre o aplicativo	Raniery	5	-	-
A11.2	Implementar as funcionalidade de informações sobre o aplicativo	Raniery	5	-	-

Iteração – 12

US12 – Implementação das funcionalidades de armazenamento das informações coletadas:

TA12.1	Armazenar os dados coletados de cada imóvel visitado.
TA12.2	Armazenar os resumos de visitas de cada dia de trabalho.

Atividade	Descrição	Responsável	Estimativa de Tempo	Tempo Real	Status
A12.1	Implementar o banco de dados	Raniery	5	-	-