



UNIVERSIDADE FEDERAL DO ACRE
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**DESENVOLVIMENTO DE UM APLICATIVO MOBILE PARA A CÂMARA
MUNICIPAL DE RIO BRANCO**

RIO BRANCO
2019

GABRIEL FIGUEIREDO BEZERRA

**DESENVOLVIMENTO DE UM APLICATIVO MOBILE PARA A CAMÃRA
MUNICIPAL DE RIO BRANCO**

Projeto de estágio supervisionado
apresentado como exigência parcial para
obtenção do grau de bacharel em
Sistemas de Informação da Universidade
Federal do Acre.

Prof. Orientador: Prof.^a Dra. Catarina de
Souza Costa

**RIO BRANCO
2019**

LISTA DE FIGURAS

Figura 1- Modelo Cascata	15
Figura 2- Desenvolvimento Incremental.....	16
Figura 3 - Protótipo Evolutivo	17
Figura 4 - React Component e React DOM.....	27
Figura 5 - Cronograma Gantt	31

SUMÁRIO

1 APRESENTAÇÃO	5
2 PROBLEMA DA PESQUISA	7
3 OBJETIVOS DA PESQUISA	9
3.1 OBJETIVO GERAL.....	9
3.2 OBJETIVOS ESPECÍFICOS.....	9
4 JUSTIFICATIVA DA PESQUISA	11
5 FUNDAMENTAÇÃO TEÓRICA	13
5.1 ENGENHARIA DE SOFTWARE.....	13
5.1.1 PROCESSOS DE SOFTWARE	14
5.1.2 ENGENHARIA DE REQUISITOS	18
5.1.3 MODELAGEM DE SISTEMAS	19
5.2 FERRAMENTAS.....	20
5.2.1 LINGUAGEM DE PROGRAMAÇÃO	20
5.2.2 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS.....	21
5.2.3 SISTEMA MOBILE	22
5.3 CONSIDERAÇÕES DO CAPÍTULO	23
6 PROCEDIMENTOS METODOLÓGICOS.....	24
6.1 CLASSIFICAÇÃO METODOLÓGICA	24
6.2 JAVASCRIPT.....	25
6.3 REACT	26
6.4 REACT NATIVE.....	27

6.5 FIREBASE	28
7 ESBOÇO DOS CAPÍTULOS E SEÇÕES	29
8 CRONOGRAMA	31
REFERÊNCIAS BIBLIOGRÁFICAS	33

1 APRESENTAÇÃO

A Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD C) realizada em 2017, expôs que a utilização de microcomputadores caiu entre 2016 e 2017 e em contrapartida, a utilização de telefone celular teve crescimento, de 92,6% para 93,2% dos domicílios. A PNAD teve como objetivo obter informações sobre Tecnologia da Comunicação e da Informação – TIC tendo como foco aspectos de acesso à internet e a televisão e posse de telefone móvel celular para uso pessoal. Na mesma pesquisa foi comprovado que o telefone é o mais utilizado para se ter acesso a internet, 98,7% dos domicílios, sendo que 43,3% usam apenas este meio para utilizar a internet (IBGE, 2017).

Existe a necessidade de os órgãos públicos expandirem os meios para comunicação com o povo através de mecanismos mais abrangentes. Esta é uma das premissas da Lei nº 12.527/2011, mais conhecida como Lei de Acesso a Informação (BRASIL, 2011). A Lei criou mecanismos que possibilitaram o acesso de informações públicas dos órgãos públicos e entidades, sem que as mesmas tenham de apresentar motivos para recebimento de tais informações. De acordo com Brasil (2011):

Art. 3º Os procedimentos previstos nesta Lei destinam-se a assegurar o direito fundamental de acesso à informação e devem ser executados em conformidade com os princípios básicos da administração pública e com as seguintes diretrizes:

II - divulgação de informações de interesse público, independentemente de solicitações;

III - utilização de meios de comunicação viabilizados pela tecnologia da informação;

Art. 5º É dever do Estado garantir o direito de acesso à informação, que será franqueada, mediante procedimentos objetivos e ágeis, de forma transparente, clara e em linguagem de fácil compreensão.

Desta forma, esse estágio visa o desenvolvimento de um aplicativo mobile para a Câmara Municipal de Rio Branco- Acre utilizando o *Framework React Native*. Para isso foi necessário a criação de telas e utilização de bibliotecas, API'S e banco de dado, que permitirá a utilização de serviços, tais como, informações acerca do sistema de apoio ao processo legislativo, cadastro de problemas, realização de cadastro de usuário, visualizar informações sobre leis, entre outros.

2 PROBLEMA DA PESQUISA

O problema a ser sanado é o fato de ser necessário abrir acesso ao conteúdo informacional do governo para a população, está necessita de acesso e qualidade informacional, para assim poder de forma mais satisfatória e eficiente realizar o exercício da democracia (FIGUEIREDO; SANTOS, 2013).

De acordo com Zanotello (2010 apud ALLI,2004):

A informática, a robótica, a inteligência artificial, a domótica, os novos materiais e as comunicações conduzem a uma variação dos meios e modos de trabalho e de troca social. A tecnologia é um elemento decisivo do progresso, configurador de um novo espaço histórico e cultural. As tecnologias da comunicação, o conhecimento em tempo real no âmbito planetário e a facilidade de intercâmbio dos conhecimentos produzem uma onda de transformações das estruturas e dos conceitos culturais.⁶ (tradução livre)

Logo uma administração transparente permite um melhor engajamento da população, pois este permite que o cidadão se torne um agente ativo do processo decisório, permitindo a ele o poder de se manifestar com aquilo que não acha coerente (ASSIS; VILLA ,2003).

A informação disponível atualmente da Câmara Municipal de Rio branco encontra-se atualmente apenas disponível em um site que não se apresenta de fácil utilização para uma população que encontra-se, cada vez mais usuária apenas de

dispositivos mobile. Para que a população tenha acesso as informações que necessita, torna-se necessário a utilização de um aplicativo para melhorar a interação da população com o órgão legislativo que se encontra competente a resolução de problemas (IBGE, 2017).

É importante ter ciência de que a própria administração pública se encontra em um processo de atraso e ineficiência no que diz respeito a gerar respostas rápidas à manutenção necessária na cidade de Rio Branco - Acre. Segundo Lima, Ramos e Fernandes (2006) a falta de informações e os receios por parte da administração em utilizar novas tecnologias faz com que se perpetue uma gerência ineficiente, não compactuando assim para que se utilize procedimentos mais arcaicos e ineficiente o que aumenta o gasto com recursos.

Diante do contexto apresentado, este trabalho tem como foco expandir os meios pelos quais a população pode adquirir informações e se comunicar com o seu órgão legislativo Municipal em prol de fomentar a resolução de problemas estruturais pertinentes em suas cidades.

3 OBJETIVOS DA PESQUISA

Os objetivos, geral e específicos, da pesquisa são listados nas seções a seguir.

3.1 OBJETIVO GERAL

Criar uma plataforma *Mobile* que possa divulgar as informações da Câmara Municipal de Rio Branco – Acre utilizando a Framework *React Native*.

3.2 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral do presente estágio, será necessário alcançar os seguintes objetivos específicos:

- a) Estudar sobre a Framework *React Native*
- b) Criar uma interface em *React Native* para o aplicativo.

- c) Passar as informações do site existente para o aplicativo através de um Web Service.
- d) Criação e implantação de uma funcionalidade que atendam aos requisitos impostos pela câmara, no que tange a identificação de problemas estruturais na cidade de Rio Branco.
- e) Realizar testes na aplicação.

4 JUSTIFICATIVA DA PESQUISA

Atualmente a Câmara Municipal de Rio Branco- Acre busca formas de expandir os meios para divulgar suas informações, para tal utiliza-se de um site e um canal no Youtube. Na busca de como melhorar significativamente a prestação do serviço público eletrônico e tornar o ente mais ágil, transparente e eficiente na divulgação de conteúdo informacional a mesma busca a implantação de um aplicativo.

As instituições governamentais do poder legislativo têm como base de atuação a participação popular, uma vez que os representantes são eleitos democraticamente para representar a população. Deste modo é de fundamental importância que os cidadãos, como foco da atuação da entidade, tenham mecanismos que facilitem o acesso as informações referentes as atuações de seus representantes.

Com o aplicativo a população terá mais um meio de interação com o poder legislativo municipal, conseguindo assim mais uma forma obter informações da instituição e até mesmo melhorar a comunicação. Sendo assim o trabalho é importante na fiscalização e pressão em cima dos vereadores, sendo mais uma forma de cobrar destes o bom desempenho de suas funções e através disso ver mudanças no espaço em que habitam.

Dessa forma, o propósito deste estágio foi estudar as ferramentas e aplicar conhecimentos adquiridos no decorrer da graduação, para o desenvolvimento e

implantação de um aplicativo mobile utilizando a *Framework React Native*. Consequentemente, suprimindo as necessidades encontradas, a qual seria, a necessidade de expandir a divulgação de conteúdo informacional da Câmara Municipal de Rio Branco – Acre para outras mídias.

5 FUNDAMENTAÇÃO TEÓRICA

Este capítulo contém os conceitos que fundamentam o presente trabalho. A Seção 2.1 aborda os conceitos de Engenharia de Software utilizados. A Seção 2.2 apresenta as ferramentas que foram utilizadas no desenvolvimento do site. Por fim, a Seção 2.3 discorre as considerações finais deste capítulo.

5.1 ENGENHARIA DE SOFTWARE

A Engenharia de Software abrange todo o projeto de desenvolvimento de um software, sendo assim é uma mistura de engenharia de sistema e de hardware, alcançando um conceito maior do que a programação individual, oferecendo dessa forma uma base para a construção de software de alta qualidade de forma mais produtiva (SOMMERVILLE, 2011)

Segundo Pressman (2011) a Engenharia de Software introduz ao desenvolvedor do processo os detalhes de como fazer o software com alta qualidade, os quais são alcançados segundo três elementos fundamentais, são eles: metas, ferramentas e procedimentos. Seguindo essa ordem os métodos seriam aqueles que proporciona detalhes de como o processo será feito, englobando um

conjunto de tarefas como: planejamento , análise de requisitos, projeto de estrutura e dados, arquitetura de programa, codificação, teste e manutenção. Já a camada de tarefa consta como a responsável pelo apoio automatizado ou semi-automatizado aos métodos e por fim, os procedimentos que conta como a ponte entre método e procedimento.

5.1.1 PROCESSOS DE SOFTWARE

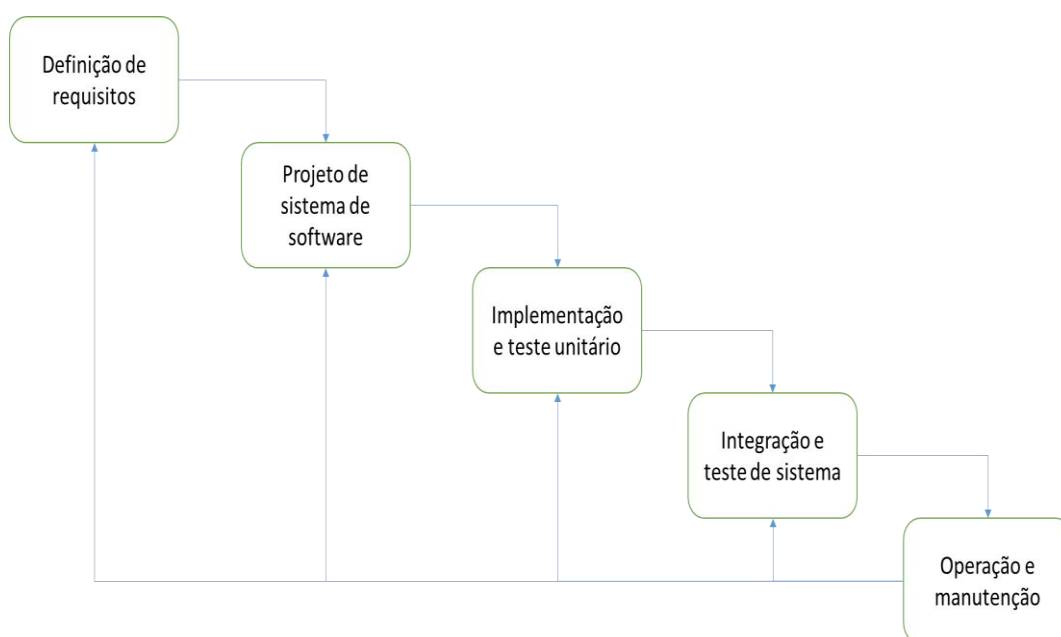
Um processador de software é o conjunto de atividades que por fim resultam num produto de software. Segundo SOMMERVILLE (2011) para simplificar e padronizar o processo de software, temos os modelos de software, os quais apresentam uma visão particular do processo, por assim dizer não se apresenta uma descrição definitiva, mas sim, abstrações que podem ser usadas para explicar diferentes formas de desenvolvimento de software.

Um dos modelos clássicos e mais utilizados é o modelo cascata, representado graficamente na Figura 1, que se apodera de uma abordagem sistemática e sequencial para produzir um software, se inicia no topo da representação e vai descendo os níveis no decorrer do ciclo de vida do desenvolvimento do software, esse modelo abrange as seguintes atividades (SOMMERVILLE, 2011):

- a) Análise e definição de requisitos: Neste nível é estabelecido os requisitos de todas as funcionalidades do sistema por meio de consultas aos usuários.
- b) Projeto de sistema e software: Esta etapa é focada na coleta dos requisitos que dizem respeito, especificamente, ao software, para que assim seja entendida a natureza dos programas que serão construídos.
- c) Implementação e teste unitário: consta como a verificação das especificações das unidades.

- d) Integração e teste de sistema: as unidades individuais do programa são colocadas para funcionar em conjunto e testadas como um sistema acabado para verificar se funciona como foi especificado.
- e) Operação e manutenção: Manutenção do sistema, possível melhora ou expansão.

Figura 1- Modelo Cascata



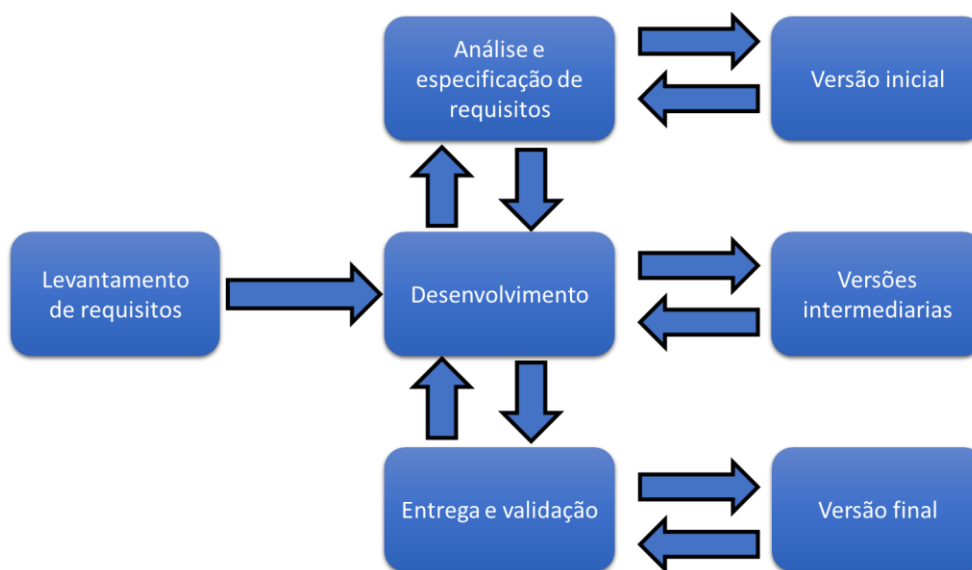
Fonte: Adaptado de Sommerville (2011).

Apesar de ter sido um dos modelos mais amplamente utilizados, o modelo cascata se encontra em defasagem, uma vez que, os projetos raramente seguem um fluxo tão preciso e sequencial, é difícil para o cliente colocar tudo que deseja explicitamente, pois há incertezas naturais no começo do projeto e ainda faz com que o cliente tenha de esperar muito para ter uma versão de trabalho do software em mãos (PRESSMAN, 2011).

Outro modelo existente é o processo de desenvolvimento incremental, apresentado graficamente na Figura 2, no qual, a ideia é desenvolvida através de versões que são julgadas pelo usuário até se chegar ao que é desejado pelo

mesmo, esse tipo de abordagem está presente em métodos ágeis. Nesse modelo são dados pequenos passos que são homologados um a um até se chegar na versão final, isso é muito útil, pois se for constatado um erro ou desejar-se realizar uma mudança no escopo, isso não impactará tão bruscamente o desenvolvimento do software (SOMMERVILLE, 2011).

Figura 2- Desenvolvimento Incremental



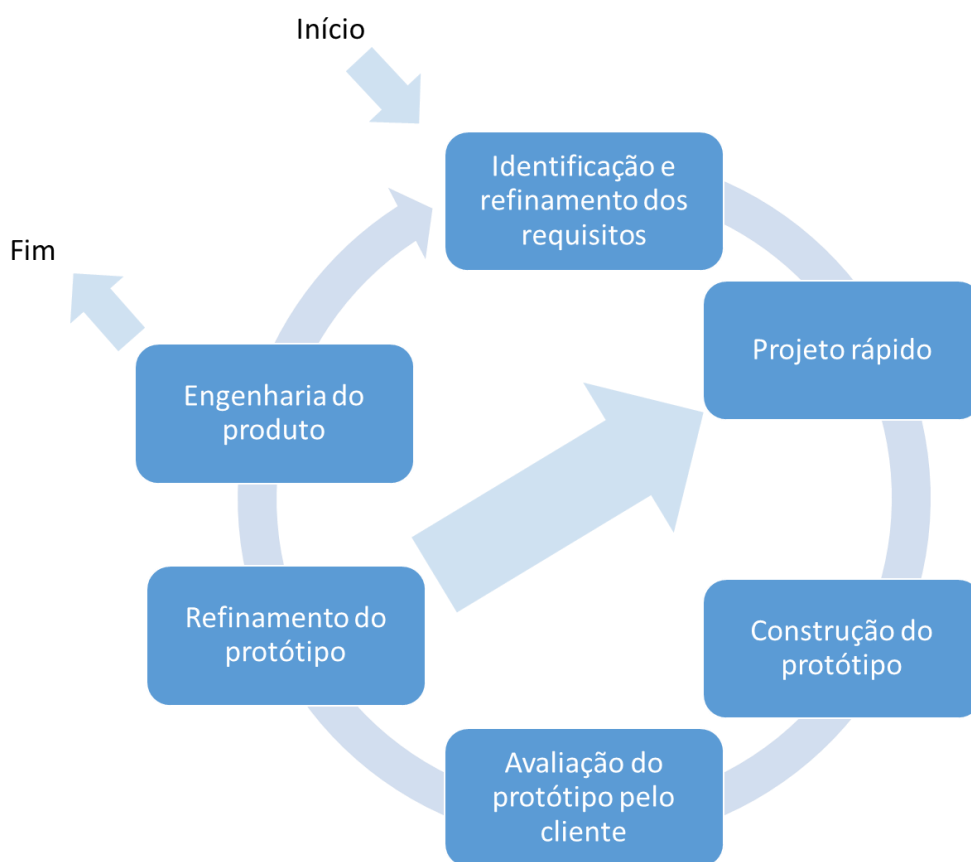
Fonte: Adaptado de Sommerville (2011).

Porém tal modelo ainda apresente problemas, como, por exemplo: os requisitos não se encontrarem tão estáveis ou completos como se esperava, ou ainda não ser possível a entrega regular de informações para os gerentes, uma vez que, o processo é ágil, para manter tamanha comunicação iria acarretar muitos gastos, e ainda as constantes mudanças podem acabar por tornar oneroso o processo, pois acabam corrompendo a estrutura do software (SOMMERVILLE, 2009).

Por fim, será falado sobre o modelo de prototipagem evolutiva, apresentado graficamente na Figura 3, esse modelo se pauta em desenvolver ou idealizar um software em baixo nível e ir lhe encorpando, inserindo suas funcionalidades no decorrer do projeto. A prototipagem evolutiva é flexível na definição dos requisitos,

os mesmos podem ser definidos no decorrer do processo, e por isso, o cliente tem de estar mais próximo o que gera melhor visibilidade da produção de seu software, porém isso cobra ao processo uma gestão sofisticada, e uma documentação bem-feita para que o produto não se perca ao longo da prototipagem (FILHO, 2000).

Figura 3 - Protótipo Evolutivo



Fonte: Adaptado de Pressman (2011).

Apesar das qualidades encontradas nesse modelo, ele também apresenta erros, entre eles, consta o apego do cliente ao protótipo, o que torna inviável uma construção mais robusta do software e acarretará em problemas futuros. O desenvolvedor pode também ocasionar situações que causarão problemas futuros no produto final, a partir do momento que escolhe uma linguagem com a qual é mais próximo para fazer o protótipo, mas por conta de sua proximidade com a linguagem

a perpétua no software ao invés de buscar uma linguagem de programação que se adapta as necessidades encontradas nos requisitos do produto (PRESSMAN, 2011).

Conforme exposto acima, o modelo que foi escolhido para a execução desse trabalho foi a prototipagem evolutiva, uma vez que tem a característica de ter uma aproximação mais abrangente com o cliente e garantir que será possível mudanças no processo de criação do software enquanto esse já se encontrar em desenvolvimento (PRESSMAN, 2011).

5.1.2 ENGENHARIA DE REQUISITOS

A engenharia de requisitos está pautada na busca por entender o que o cliente quer realmente no seu produto, segundo Sommerville (2011) esse se trata do processo de descoberta e refinamento que se tem durante a documentação de requisitos. Durante esse processo o cliente tenta trazer a luz o que ainda e não está claro para ele, tentando assim deixar de forma mais concretas o que realmente deve ser desempenhado pelo software.

Os requisitos são importantes pois garantem que o software seja entregue corretamente, de forma a garantir que se gaste recurso apenas com o que e extremamente necessário, evitando assim criação errônea de componentes (PRESSMAN ,2011). Os requisitos podem ser classificados em duas características, sendo elas requisitos funcionais e não funcionais, o primeiro se trata das respostas do sistema a ação do usuário já o segundo está relacionado com as propriedades que provém do sistema (FILHO, 2000).

5.1.3 MODELAGEM DE SISTEMAS

O modelo busca fazer uma representação simplificada do mundo real, logo a modelagem de sistema busca promover uma melhor comunicação entre programador e usuário, facilitando a demonstração sobre os vários produtos a desenvolver, bem como avaliação de custos e dados através de representações ordenadas sobre o sistema (PEREIRA, 2011).

Os modelos normalmente fazem o uso de notação gráfica para tentar repassar qual seria o funcionamento do sistema (PRESSMAN, 2011). Os modelos fazem uso de linguagens para descrever as exigências sem ambiguidade e assim dar os meios para se realizar os propósitos do sistema, a modelagem permite, ainda, que seja feita a documentação do sistema, conseguindo assim se registrar as características intrínsecas do mesmo (SILVA; VIDEIRA, 2001).

Existem várias linguagens que atendem satisfatoriamente a tarefa de modelagem, a que será retratada aqui é a UML (Unified Modeling Language), pois esta faz bem a tarefa de criar um modelo conciso, com uma linguagem que pode ser facilmente entendível por humanos e interpretada pelas máquinas (PEREIRA, 2011).

Os diagramas são as ferramentas que dissecam os problemas e expõem as possibilidades de interligação dos elementos de uma forma lógica ou estrutural. A UML, apresenta diferentes tipos de diagramas, que são (SILVA; VIDEIRA, 2001):

- a) Diagrama de caso de uso que capta o sistema na visão de quem o utiliza;
- b) Diagrama de classe que apresenta um modelo de alto grau de abstração;
- c) Diagrama de interação entre objetos que permite se ter noção da dinâmica de funcionamento de um sistema.

- d) Diagrama de componente e diagrama de instalação que permite a visualização dos componentes físicos.

Os diagramas que são efetivamente usados nesse trabalho são os de caso de uso e classe.

5.2 FERRAMENTAS

Nesta seção serão abordadas as ferramentas utilizadas para o desenvolvimento do estágio supervisionado, bem como a Linguagem de Programação, o Sistema de Gerenciamento de Banco de Dados e o Sistema Mobile.

5.2.1 LINGUAGEM DE PROGRAMAÇÃO

A linguagem de programação é uma formalização que permite com que o computador entenda o programador, ou seja, para um computador executar qualquer função é necessário que haja uma comunicação entre o programador e a máquina, é o intermediador dessa tarefa é a linguagem de programação (GABBRIELLI; MARTINI, 2010).

A linguagem pode ser dividida em baixo nível ou alto-nível, a primeira representa a linguagem de máquina, a qual é representada apenas em linguagem binária constituída de regras, a qual, é mais difícil para a interpretação humana, já a segunda se apresenta como algo mais próxima da linguagem humanos. Quando se está programando necessita-se de uma boa comunicação entre esses dois tipos de linguagem, que é realizada por um compilador, que terá a função de tornar a linguagem de alto nível interpretável pela máquina (FERRARI; CECHINEL, 2008).

A Linguagem de Programação que será usada nesse trabalho é JavaScript, mais detalhes serão dados na seção 6.4.

5.2.2 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Um sistema de gerenciamento de banco de dados é a junção de softwares que tem a função de gerenciar a coleção de dados, em suma, e basicamente um sistema que faz o armazenamento e recuperação de informações de uma maneira mais conveniente e eficiente (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Os bancos de dados são imprescindíveis para o armazenamento de dados em uma organização, pois é extremamente difícil manter uma administração dos dados sem esse tipo de software, uma vez que, esse sistema economiza muito em recursos, análise, manutenção de dados (DATE, 2004).

Com o advento do Big Data, que pode ser melhor classificado como um grande conjunto que precisam ser processados e armazenados, os dados estão se expandindo a uma taxa cada vez maior e esse aumento exponencial fez com que não fosse mais viável usar os bancos de dados relacionais, uma vez que, estes estavam sendo levados ao limite (MARZ; WARREN, 2015).

Diante desse contexto de mudanças radicais, foi necessário a utilização do banco de dados NoSQL, não relacional, que permitem velocidade, flexibilidade e escalabilidade ao armazenar e acessar dados não estruturados. Segundo Marz e Warren (2015) o NoSQL apresenta 4 paradigmas:

- a) Chave-valor apresentam um modelo simples que armazena e procura um dado usando um identificador.
- b) Orientado a documento é um modelo expande a ideia de chave-valor, pois organizam um conjunto de chaves-valores em uma estrutura lógica, isso propicia um melhor desempenho e flexibilidade.
- c) Família de coluna é um paradigma que apresenta um conjunto de famílias de colunas, estas são compostas em colunas relacionais e as

linhas e colunas são identificadas como chaves e o registro de data e hora permite diferenciar as múltiplas versões de um mesmo dado.

- d) Gráficos apresenta uma estrutura de nós que se conectam, os nós são os bancos de dados e as conexões são as saídas ou entradas, esse modelo é adequado para modelagem de rede.

Esse trabalho fará uso do banco de dados Firebase, isso será melhor discutido na seção 6.3 , que faz uso de um paradigma chave valor para armazenar os dados oriundos do aplicativo.

5.2.3 SISTEMA MOBILE

Um sistema operacional móvel (Mobile OS) se apresenta como um software plataforma que permite com que programas possam ser executados em dispositivos móveis (OKEDIRAN et al., 2014).

Aparelhos mobile, na atualidade, são um dos eletrônicos mais consumidos pela sociedade, e todos necessitam de um sistema operacional para conseguir rodar seus serviços, principalmente porque atualmente tais sistemas se encontram bem mais complexos e robustos, possuindo até muitas características que antes apenas um computador teria (OKEDIRAN et al., 2014).

Quando ocorreu a mudança de sistemas analógicos para digitais, os celulares sofreram mudanças para conseguir interagir com serviços mais avançados, como o uso de mensagens curtas e conexão com a internet (FITZEK; REICHERT, 2007).

Os smartphones atuais são projetados para que desenvolvedores externos possam conseguir desenvolver aplicações para os dispositivos, e através dessas

características os usuários podem conseguir novos aplicativos conectando-se as lojas de aplicativo disponíveis (OKEDIRAN et al., 2014).

No contexto descrito acima será construído um aplicativo mobile usando a *Framework React Native*, que será melhor discutida na seção 6.2.

5.3 CONSIDERAÇÕES DO CAPÍTULO

Com base em todas as informações apresentadas neste capítulo, é possível ter uma dimensão sobre toda a informação que está contido nesse estágio, pode-se ter uma compreensão através da base teórica do funcionamento das ferramentas que serão empregadas bem como entender a necessidade de toda a documentação que será gerada.

6 PROCEDIMENTOS METODOLÓGICOS

Esse capítulo trata das metodologias e ferramentas que serão utilizadas neste estágio. A Seção 6.1 é realizada uma classificação metodológica. A Seção 6.2 define a linguagem de programação JavaScript. A Seção 6.3 define o framework React. A seção 6.4 apresenta o framework React Native. Por fim a seção 6.5 trata do banco de dados Firebase

6.1 CLASSIFICAÇÃO METODOLÓGICA

O trabalho produzido pode se encaixar em diversas conjunturas, segundo as formas clássicas esse trabalho pode ser classificado segundo seus objetivos, abordagem do problema, natureza e procedimento técnico (GIL ,2008).

No que tange aos objetivos esse trabalho pode entrar na classificação de pesquisa exploratória, pois se embasa na inexistência ou quantidade diminuta de dados disponíveis, logo o que resta e aperfeiçoar ideias e construção de conjecturas, ou seja, se debruçar em cima de respostas antecipadas (SANTOS, 2016).

No que diz respeito a abordagem do problema esse trabalho pode ser classificado como uma pesquisa qualitativa, pois o objetivo aqui disposto é apenas classificar um conjunto de observações (MARCONI; LACATOS, 2017).

No que tange a natureza, o trabalho se enquadra nos quesitos de uma pesquisa aplicada, uma vez que tem por objetivo gerar conhecimento e sanar problemas específicos (GIL, 2008).

Por fim os objetivos do trabalho, levando em conta o contexto do trabalho o objetivo que mais o capta é a pesquisa exploratória, pois tal pesquisa não faz uso de técnicas quantitativas, buscando uma familiaridade com o problema e gera hipóteses (GIL, 2008).

6.2 JAVASCRIPT

JavaScript é uma linguagem de programação que foi criada com o objetivo de realizar programação em navegadores gráficos de tal forma que fosse possível fazer aplicativos Web de forma mais prática (BROWN, 2016).

Segundo Minnick e Holland (2015) as linguagens de programação podem ser divididas em dois tipos: compiladas e interpretadas, as compiladas são aquelas em que o programador escreve o código para depois executá-lo em um compilador (está presente em linguagens como o C, C++, entre outros), já as interpretadas são aquelas que não precisam passar pelo compilador e já vão direto à máquina para ser executado, esse é o caso do JavaScript (MINNICK; HOLLAND, 2015).

O JavaScript apresenta-se como uma linguagem altamente liberal, com uma ideia de design que buscava ser atrativa e facilitar a utilização, para que assim conseguisse mais adeptos, podendo ser estes programadores experientes ou iniciantes na área da programação. (BROWN, 2016).

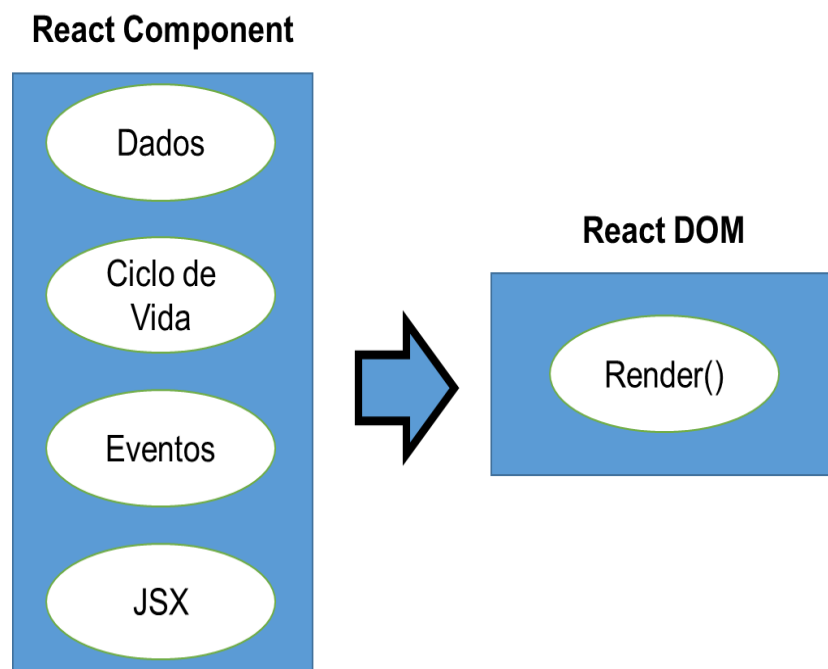
6.3 REACT

O *React* é uma biblioteca para construir interfaces de usuário, não se encaixando como um framework que irá lidar com tudo, mas sim como uma camada de visualização (BODUCH, 2017).

Segundo Boduch (2017) o *React* trabalha com API's, tal forma de funcionamento dispensa a necessidade de o programador possuir conhecimento acerca do funcionamento total deste framework. Este framework na base de seu funcionamento *apresenta-se como uma partição de duas API's* maiores, sendo elas: DOM e o *React Component API*.

Como representado na Figura 4, o *DOM* é usado para realizar a renderização de uma página web, já o *React Component API*, são as partes da página que são efetivamente renderizadas pelo *React DOM*. O *React Component* é dividido em quatro áreas, são elas (BODUCH, 2017):

- a) Dados que são provenientes de algum lugar, para o componente não é importante saber de onde, e é processado.
- b) O ciclo de vida que são métodos implementados para responder às mudanças que acontecem durante os processos do componente.
- c) Eventos que é o código que o programador escreve para responder as ações do usuário.
- d) JSX que consta como a sintaxe dos componentes *React* usados para descrever a interface do usuário.



Fonte: Adaptado de Boduch (2017).

6.4 REACT NATIVE

O *React Native* é um framework que utiliza de JavaScript para escrever aplicativos móveis nativos para IOS e Android (BODUCH, 2017). O código feito nesse framework pode ser, em sua maior parte, compartilhado entre as plataformas, o que diminui recursos e tempo no desenvolvimento do aplicativo mobile (EISENMAN, 2016).

O *React Native* funciona através de um método que faz uso de chamadas assíncronas para o sistema operacional móvel subjacente, cuja a lógica para funcionamento faz uso de JavaScript e da API React (esta API é fundamentalmente a mesma do React (seção 6.3), porém tem a diferença de não apresentar o DOM, mas sim a API assíncrona) (BODUCH, 2017).

Segundo Eisenman (2016) o *React Native* apresenta-se bem intuitivo para programadores com um certo nível de experiência na fabricação de aplicativos mobile, por conta de suas ferramentas de desenvolvedor e mensagens de erros

relevantes, o que faz com que a *Framework* seja bem robusta, um exemplo de ferramenta importante a ser citado e uma das características do JavaScript (citada na seção 6.1), como está se enquadra nos parâmetros de uma linguagem de programação interpretada não é necessário recriar o aplicativo para ver as alterações, sendo necessário apenas atualizar a aplicação.

6.5 FIREBASE

O *Firebase* se apresenta como um banco de dados NoSQL, que utiliza do paradigma chave-valor, que possibilita o uso de muitos serviços pertencentes ao Google para desenvolvedores mobile/web, incluindo autenticação de usuário, armazenamento e hospedagem de dados em tempo real, entre outros (FIREBASE, 2019).

O *Firebase* apresenta dois bancos de dados: *Realtime Database* e *Cloud Firestore*, o primeiro é um banco de dados que apresenta baixa latência e eficiência para a utilização em aplicativos mobile que necessitam de gerenciamento de banco de dados em tempo real, já o segundo é o mais recente banco de dados desenvolvido pela Google que demonstrou resultados melhores que o *Realtime Firebase*, no que tange clareza na utilização, qualidade de consulta e rapidez (FIREBASE, 2019).

No trabalho a ser realizado o banco de dados que será efetivamente usado será o *Realtime Firebase*, um banco de dados com hospedagem em nuvem, que tem a capacidade de armazenar dados no formato JSON e sincronizar tais dados com todos os clientes conectados. O banco de dados a ser citado tem a capacidade de mesmo off-line conseguir armazenar os eventos que estão ocorrendo, e quando se conecta a uma rede descarrega todos os dados no repositório sem causar nenhum conflito (FIREBASE, 2019).

7 ESBOÇO DOS CAPÍTULOS E SEÇÕES

O relatório de estágio seguirá a seguinte estrutura:

1. INTRODUÇÃO

1.1 PROBLEMA DA PESQUISA

1.2 OBJETIVOS DA PESQUISA

1.2.1 Objetivo geral

1.2.2 Objetivos específicos

1.3 JUSTIFICATIVA DA PESQUISA

1.4 METODOLOGIA

1.5 ORGANIZAÇÃO DO TRABALHO

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Engenharia de Software

2.1.1 Processos de Software

2.1.2 Engenharia de Requisitos

2.1.3 Modelagem de Sistema

2.2 FERRAMENTAS

2.2.1 Linguagem de Programação

2.2.2 Sistema de Gerenciamento de Banco de Dados

2.2.3 Sistema Mobile

2.3 Considerações do Capítulo

3 ESTÁGIO SUPERVISIONADO

3.1 INSTITUIÇÃO

3.2 ESTRUTURA ORGANIZACIONAL

3.3 ATIVIDADES DESENVOLVIDAS

3.3.1 Levantamento de Requisitos

3.3.2 Modelagem de Caso de Uso

3.3.3 Criação de Protótipos

3.3.4 Desenvolvimento da Aplicação Mobile

3.3.5 Desenvolvimento de BackEnd

3.3.6 Desenvolvimento do Site

3.3.7 Testes

3.4 RESULTADOS OBTIDOS

4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES

4.1 CONSIDERAÇÕES FINAIS

4.2 RECOMENDAÇÕES

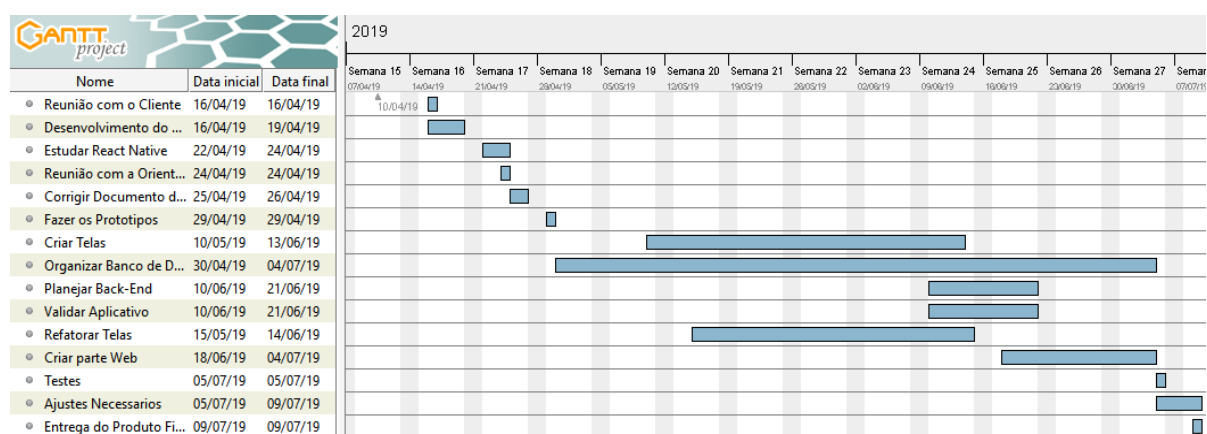
REFERÊNCIAS BIBLIOGRÁFICAS

8 CRONOGRAMA

O cronograma do estágio foi desenvolvido com base nos princípios prototipação evolutiva de software. A Figura 5 apresenta as fases que compõem o cronograma.

A primeira parte do estágio consta com a parte de levantamento de requisitos, estudo das tecnologias a serem usadas e prototipação, este período corresponde o período entre 16/04/2019 a 29/04/2019.

Figura 5 - Cronograma Gantt



Fonte: elaboração própria

Após essa primeira etapa, será dado início o período para construção das telas, e em concorrência a está a refatoração de telas, a organização do banco de dados que acontecerão ao longo do projeto

Nas etapas finais será criada a parte Web, que ocorrerá entre 18/06/2019 até 04/07/2019, e por fim serão feitos os testes e ajustes finais para a entrega ser realizada no dia 09/07/2019.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSIS, M. M. A.; VILLA, T. C. S. **O controle social e a democratização da informação: um processo em construção**. Revista Latino-Americana de Enfermagem, Ribeirão Preto, v.11, n.3, p.376-382, 2003. Disponível em : <<http://www.scielo.br/pdf/rlae/v11n3/16549.pdf>>. Acesso em: 02 de jun. de 2019.

BODUCH, A. **Ract and React Native**. Birmingham: Packt Publishing, 2017.

BRASIL. (18 de Novembro de 2011) **LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011**. *Lei de Acesso à Informação*. Disponível em: <<https://www2.camara.leg.br/legin/fed/lei/2011/lei-12527-18-novembro-2011-611802-norma-pl.html>>. Acesso em: 02 de jun. de 2019.

BROWN, E. **Learning Javascript: Add Sparkle and Life to your Web Page**. California: O'Reilly Media, 2016.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. Rio de Janeiro: Elsevier, 2004.

FERRARI, F.; CECHINEL, C. **Introdução a algoritmos e programação**. Bagé: Universidade Federal do Pampa, 2008.

FILHO, W. P.P. **Engenharia de Software: fundamentos, métodos e padrões**. Rio de Janeiro: LTC, 2000.

FIREBASE. **Firestore RealTime Database**. Disponível em: <<https://firebase.google.com/docs/database?hl=pt-BR>>. Acesso em: 02 de jun. de 2019.

FITZEK, F. H. P; REICHERT, F. **Mobile Phone Programming and its Application to Wireless Networking**. Spring, 2007.

GABBRIELLI, M; MARTINI, S. **Programming Languages: Principles and Paradigms**. Londres: Springer-Verlag London Limited, 2010.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Editora Atlas SA, 2008.

LIMA, J. P; RAMOS, R. A. R; FERNANDES, L. **A Prática de Gestão de Pavimentos em Cidades Médias Brasileiras**. In: PLURIS 2006: anais do 2º Congresso Luso-Brasileiro de Planeamento Urbano Regional Integrado Sustentável, 2. 2006, Braga. Anais... Braga,1985.

MARCONI, M. A; LAKATOS, E. M. **Metodologia Científica**. 7.ed. São Paulo: Editora Atlas Ltda, 2017.

MARZ, N; WARREN, J. **Big Data: Principles and Best Practices of Scalable Real-Time Data Systems**. Nova York: Manning Publications, 2015.

MINNICK, C; HOLLAND, E; **Coding with JavaScript for Dummies**. Nova Jersey: John Wiley & Sons, 2015.

PEREIRA, L. A. M. **Análise e Modelagem de Sistemas com a UML**. Rio de Janeiro,2011.

PRESSMAN, R. S. **Engenharia de software**. São Paulo: Pearson, 2011. Londres, 2010.

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

SILVA, A. M. R; VIDEIRA, C. A. E. **UML, Metodologias e Ferramentas CASE**. Porto: Centro Atlântico, 2001.

ZANOTELLO, S. **A tecnologia da informação no governo e as compras via internet**. Revista de Estudos sobre área de Direito. 2010. Disponível em : <http://www.portal.anchieta.br/revistas-e-livros/direito/pdf/direito14_5.pdf>. Acesso em: 02 de jun. de 2019.