

Sincronismo de Dados Entre um Sistema Web e um Sistema para Dispositivos Móveis, sem Conexão à Internet

Silvério Oliveira de Paula

Faculdade de Tecnologia de Garça (FATEC) - contato@silverio.eti.br

João Baptista Cardia Neto

Faculdade de Tecnologia de Garça (FATEC) - joao.cardia@fatec.sp.gov.br

Larissa Pavarini da Luz

Faculdade de Tecnologia de Garça (FATEC) - larissa.luz01@fatec.sp.gov.br

Resumo

No Brasil, devido ao elevado grau de insatisfação dos usuários quanto à qualidade dos serviços prestados pelas operadoras de telefonia móvel, incluindo o acesso à Internet, onde em diversas regiões do país o sinal é ineficiente ou até mesmo inexistente, optou-se por desenvolver uma aplicação da qual o uso da Internet foi descartado. A presente aplicação consiste em um Sistema Web, que atenda micro e pequenas empresas que trabalham com entrega de produtos, desenvolvido utilizando-se a linguagem de programação PHP com *Framework CodeIgniter* e banco de dados MySQL. Além do Sistema Web, foi criado um aplicativo para dispositivos móveis Android, que apoiará a entrega dos pedidos efetuados além de, ao se sincronizar, atualizar com novos pedidos por outros clientes. A fim de reduzir custos e evitar possíveis quedas ou oscilações durante a entrega de produtos em regiões remotas, principalmente naquelas onde não há conexões de dados móveis, os dados serão armazenados no próprio dispositivo e posteriormente sincronizados com o servidor através da rede local.

Palavras-chaves: Telefonia móvel. Sistema Web. Android. Web Service.

Data synchronization between a web system and a system for mobile devices, without Internet connection

Abstract

In Brazil, due to the high degree of user dissatisfaction with the quality of the services provided by the mobile operators, including Internet access, where in many regions of the country the signal is inefficient or even non-existent, it was decided to develop an application of which the use of the Internet has been discarded. The present application consists of a Web System, which serves micro and small companies that work with product delivery, developed using the PHP programming language with *Framework CodeIgniter* and MySQL database. In addition to the Web System, an application was created for Android mobile devices, which will support the delivery of the orders made and, when synchronizing, will update with new orders by other customers. In order to reduce costs and prevent possible downtime or oscillations during product delivery in remote regions, especially those where there are no mobile data connections, the data will be stored on the device itself and later synchronized with the server through the local network.

Key-words: Mobile telephony. Web system. Android. Web Service.

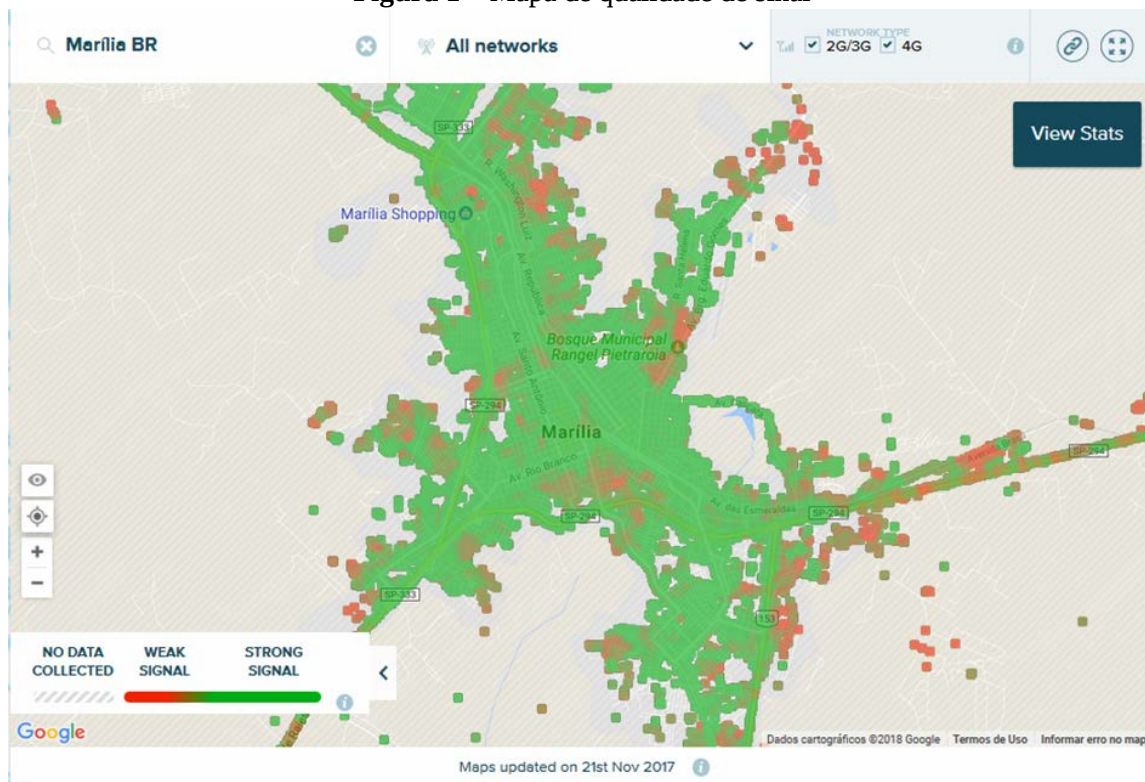
1. Introdução

Segundo IDEC (Instituto Brasileiro de Defesa do Consumidor), em sua análise da Internet móvel, mostram um cenário ruim para a qualidade das redes 2G e 3G no Brasil. De acordo com a análise, tendo como critério a velocidade download e utilizando-se do aplicativo *OpenSignal*, concluiu-se que o Brasil oferece um serviço com qualidade 30% inferior a média mundial e o investimento em infraestrutura das redes móveis no interior ainda são inferiores aos que são feitos nas capitais.

Segundo estudo da *OpenSignal*, que levantou dados de 95 países, disponibilidade de 4G e qualidade do sinal deixam a desejar no Brasil. (CAPELAS, 2016).

A Figura 1 mostra o mapa da qualidade de sinal, gerado no site da *OpenSignal*, exibindo as áreas de cobertura onde o sinal é mais fraco ou inexistente (pontos vermelhos), incluindo todas as operadoras, na cidade de Marília, interior de São Paulo.

Figura 1 – Mapa de qualidade de sinal



Fonte: *OpenSignal* (2018)

Quanto mais distante estamos do centro da cidade, menor é a intensidade do sinal, ocorrendo então falhas e oscilações da Internet, o que justifica a pesquisa de uma solução alternativa para as empresas que trabalham com entregas de produtos em regiões mais remotas e desejam controlar seus pedidos e clientes através de dispositivos móveis e persistir essas informações em uma base de dados centralizada.

2. Objetivo

Reduzir custos e evitar problemas durante a entrega de produtos, descartar o uso da Internet e a impressão de documentos, manter um banco de dados central e fazer uso de infraestrutura de baixo custo para o servidor.

Através da rede local, via roteador *Wireless*, evitar problemas de inconsistência ou corrupção de dados durante a transmissão e recepção entre o Sistema *Web* e os dispositivos móveis com sistema operacional Android, mantendo a centralização e a integridade dos dados.

2.1 Específico

Desenvolver um Sistema *Web* e um aplicativo para dispositivos móveis com sistema operacional Android, para gerenciamento de endereços, clientes, produtos, rotas, dispositivos, controle de pedidos e entregas que, através da rede local, via roteador *Wireless*, envie pedidos a entregar para os dispositivos móveis, onde os entregadores farão uso desses dispositivos para controlar suas entregas de forma eficiente.

O Sistema *Web*, por meio de um *Web Service*, manterá o banco de dados central, instalado no servidor, sincronizando informações com os dispositivos móveis.

Justifica-se o desenvolvimento utilizando-se a plataforma *Web*, mesmo funcionando em uma rede local, devido a tendência mundial da Computação em Nuvem, onde a migração do presente sistema para esta nova plataforma poderá ocorrer com maior compatibilidade.

3. Metodologia

Uma pesquisa de satisfação e qualidade dos serviços de telefonia móvel foi realizada através do método quantitativo e qualitativo, por meio da aplicação de um questionário online estruturado e fechado, disseminado em redes sociais. Foram considerados para efeito estatístico e atualização da teoria sobre a Internet móvel oscilante no Brasil, somente as pessoas residentes no interior e Capital do Estado de São Paulo.

Em uma entrevista informal, junto ao proprietário de uma microempresa que trabalha com vendas e entregas de produtos de limpeza, foi possível fazer levantamento e análise de requisitos com foco no gerenciamento de pedidos e entrega de produtos, onde foi sugerido a não utilização da Internet, com o objetivo de reduzir custos. No presente projeto a microempresa receberá o nome fictício de Pedidos.lan.

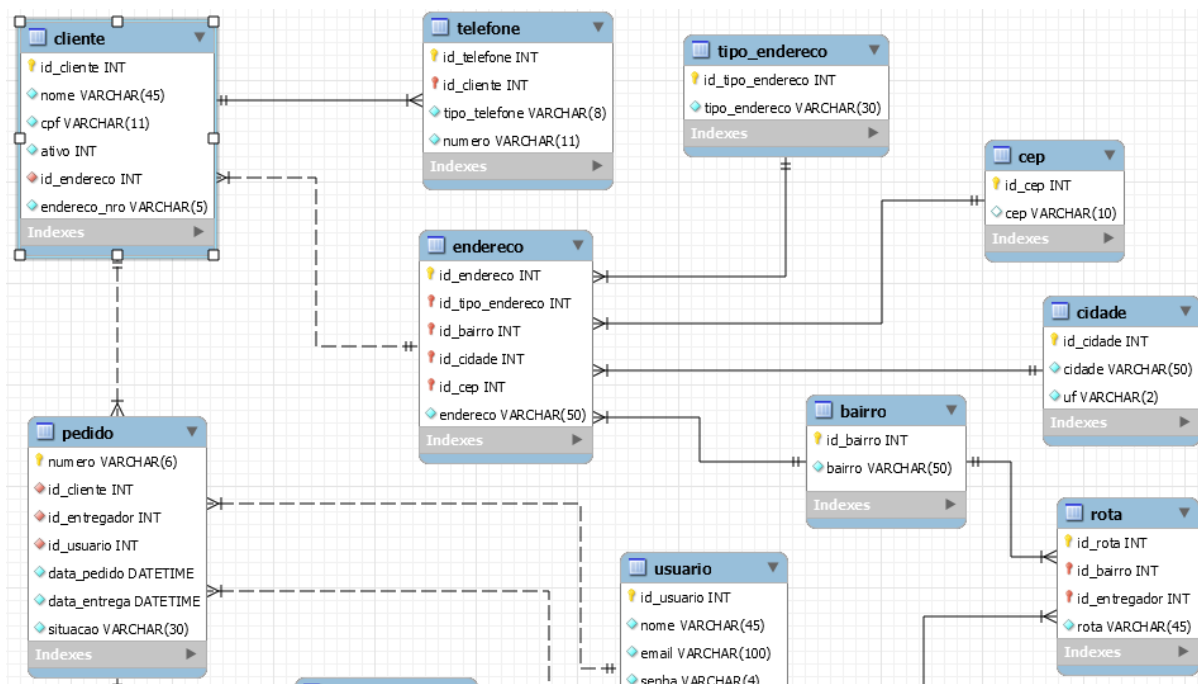
4. Desenvolvimento

4.1 Banco de Dados Mysql - Servidor

Utilizou-se o Sistema Gerenciador de Banco de Dados (SGBD) MySQL como banco principal, instalado no servidor, com foco no controle de endereços, onde considerou-se de suma importância para o controle de pedidos e entregas e sua estrutura parcial é exibida na Figura 2.

Figura 2 – Estrutura Parcial do Banco de Dados Principal

Sincronismo de Dados entre um Sistema Web e um Sistema para Dispositivos Móveis, sem Conexão à Internet



Fonte: Própria do Autor (2018)

4.2 Banco de Dados SQLite – Dispositivos Móveis Android

Para os dispositivos móveis, utilizou-se o banco de dados SQLite, nativo do sistema operacional Android, para que nele fossem armazenados os pedidos a entregar. Esses pedidos, após concluídas as entregas, são excluídos após sincronização e atualização dos dados com o banco principal através da rede local, via roteador Wireless.

A Figura 3 mostra a estrutura do banco de dados criada para os dispositivos Android.

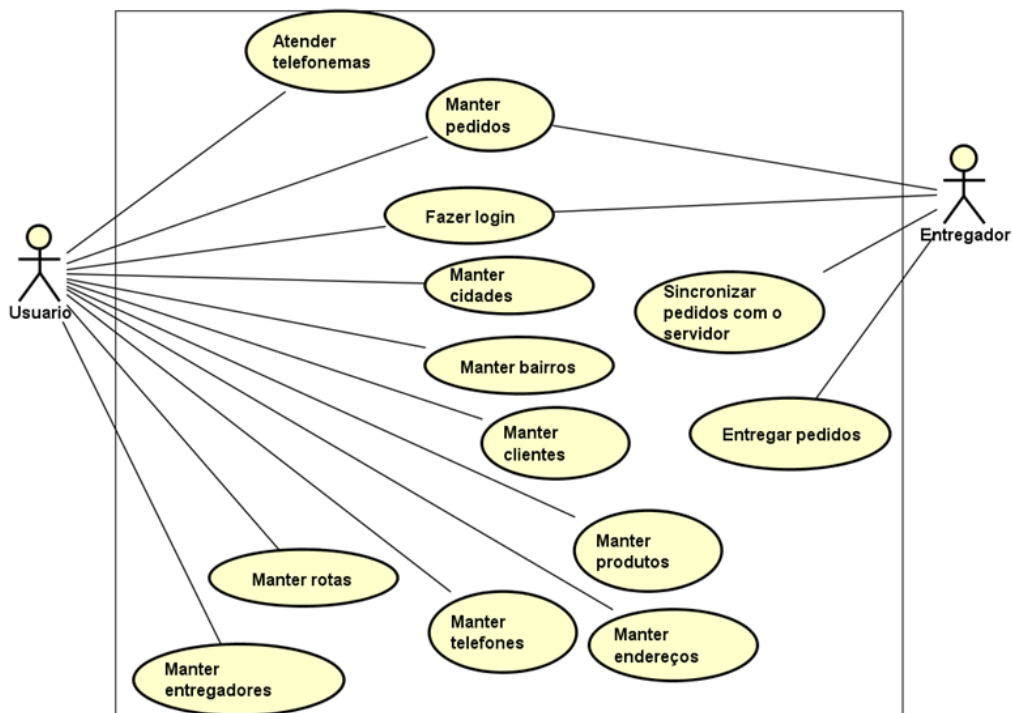
Figura 3 – Estrutura do Banco de Dados SQLite

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL("CREATE TABLE pedidos(" +  
        +"numero TEXT PRIMARY KEY, "  
        +"data_pedido TEXT, "  
        +"data_entrega TEXT, "  
        +"horas_entrega TEXT, "  
        +"nome_cliente TEXT, "  
        +"tipo_endereco TEXT, "  
        +"endereco TEXT, "  
        +"endereco_nro TEXT, "  
        +"situacao TEXT);");  
  
    db.execSQL("CREATE TABLE itens(" +  
        +"pedido_numero TEXT, "  
        +"id_produto TEXT, "  
        +"qtde TEXT, "  
        +"descricao TEXT, "  
        +"valor TEXT, PRIMARY KEY (pedido_numero,id_produto), FOREIGN KEY (pedido_numero) REFERENCES pedidos (numero));");  
  
    db.execSQL("CREATE TABLE produtos(" +  
        +"id_produto TEXT PRIMARY KEY, "  
        +"descricao TEXT, "  
        +"valor TEXT);");  
  
    db.execSQL("CREATE TABLE enderecos(" +  
        +"id_endereco TEXT PRIMARY KEY, "  
        +"endereco TEXT);");  
}
```

Fonte: Própria do Autor (2018)

O diagrama de caso de uso da Figura 4, demonstra o resultado obtido, após entrevista informal, o levantamento e análise de requisitos, da microempresa Pedidos.lan e descreve as principais funcionalidades do Sistema Web.

Figura 4 – Diagrama de Caso de Uso Pedidos.lan



Fonte: Própria do Autor (2018)

Entre os requisitos analisados, destacou-se a não utilização da Internet e a probabilidade de que, quando as entregas estiverem em andamento, outros clientes, que não os já cadastrados no sistema, queiram adquirir produtos. Sendo assim, esses novos clientes deverão ser inseridos

no banco de dados do dispositivo móvel para posteriormente serem inseridos no banco de dados central, motivos pelas quais reforçaram a criação de um banco de dados SQLite para os dispositivos móveis.

4.3 Servidor com Sistema Operacional Centos Linux

De acordo com Martini (2013) “um servidor é um computador ou hardware que fornece um serviço ininterrupto na rede” e, em caráter experimental, utilizou-se um computador simples, com hardware básico, com o Sistema Operacional CentOS Linux, que segundo Ferreira (2008) “o Linux é um clone de Unix criado como uma alternativa barata e funcional para quem não está disposto a pagar o alto preço de um sistema Unix comercial ou não tem um computador suficientemente rápido”. Para maior performance, não foi utilizada nenhuma interface gráfica.

Utilizou-se um computador *Desktop* simples, possuindo apenas uma interface *ethernet* com o endereço IP 192.168.0.50 fixo.

4.4 Framework CodeIgniter e o Padrão MVC

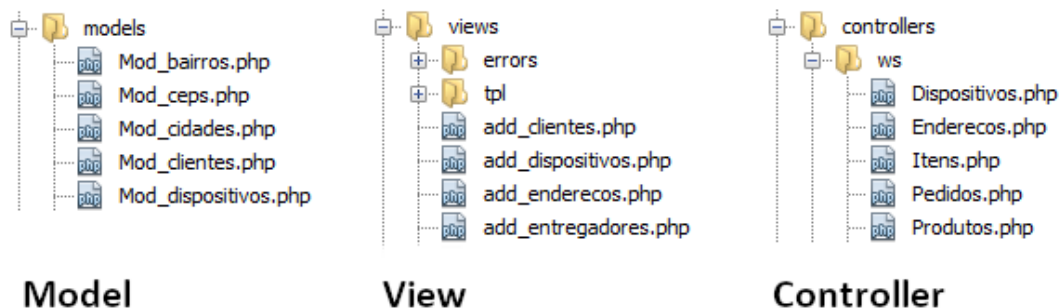
“O CodeIgniter [...] é um *framework MVC open source*, escrito em PHP e mantido atualmente pelo *British Columbia Institute of Technology* e por uma grande comunidade de desenvolvedores ao redor do mundo.” (ANTUNES, 2016). O CodeIgniter nos oferece a estrutura MVC pronta para uso além de várias bibliotecas que aceleram o processo de desenvolvimento.

“MVC é um padrão de design de projetos que divide as tarefas em três camadas distintas de código: modelagem de objetos, apresentação e controle de fluxo e carregamento de recursos.” (CABARDO, 2015). A distribuição em camadas nos auxilia enormemente na manutenção do sistema, bem como da organização dos arquivos. O CodeIgniter foi utilizado para abstrair e diminuir a quantidade de código a ser escrito com a linguagem de programação PHP, que segundo sua documentação “é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento Web e que pode ser embutida dentro do HTML”.

Não foi necessária nenhuma instalação especial para o CodeIgniter, bastando apenas o download e descompactação do arquivo no diretório apropriado no servidor Web, obtendo-se a estrutura de diretórios e arquivos prontos para uso. Destaca-se os diretórios *model*, *views* e *controller* depois da instalação conforme a Figura 5.

Figura 5 – Diretórios Model, Views e Controller do CodeIgniter

Sincronismo de Dados entre um Sistema Web e um Sistema para Dispositivos Móveis, sem Conexão à Internet



Fonte: Própria do Autor (2018)

4.5 Web Service Rest e Json

Segundo a documentação do NETBEANS (2018):

REST (*Representational State Transfer*) é um estilo de arquitetura para sistemas de hipermídia distribuídos, como a *World Wide Web*. O conceito dos recursos identificados por identificadores de recursos universais (URIs) é central para a arquitetura RESTful. Esses recursos podem ser manipulados usando uma interface padrão, como HTTP, e as informações são trocadas usando representações desses recursos.

Para a sincronização de dados entre servidor e dispositivos móveis utilizou-se um *Web Service* REST e formato de respostas JSON.

“O ponto mais forte do JSON é seu formato de dados simples. Não é preciso muito espaço de armazenamento [...] o que o torna ideal para dispositivos menos potentes, como telefones.” (MITCHELL, 2016).

A Figura 6 exibe o código executado pelo *Web Service* Rest que retorna um *array* de pedidos no formato JSON, requisição feita pelo dispositivo móvel que opera como um consumidor dos recursos Rest.

Figura 6 – Código executado pelo *Web Service*

```
public function listar($id_rota)
{
    $listaPedidos = array();
    $pedidos = $this->mod_pedidos->getAllLista($id_rota);
    foreach ($pedidos->result() as $pedido) {
        $data_pedido = date('d/m/Y', strtotime($pedido->data_pedido));
        $data_entrega = date('d/m/Y', strtotime($pedido->data_entrega));
        $temp = [
            'numero'=>$pedido->numero,
            'data_pedido'=>$data_pedido,
            'data_entrega'=>$data_entrega,
            'cliente_nome'=>$pedido->cliente_nome,
            'tipo_endereco'=>$pedido->tipo_endereco,
            'endereco'=>$pedido->endereco,
            'endereco_nro'=>$pedido->endereco_nro
        ];
        array_push($listaPedidos, $temp);
    }
    echo json_encode($listaPedidos);
}
```

Fonte: Própria do Autor (2018)

A Figura 7 exibe uma tela do dispositivo móvel que fez a requisição de uma lista de pedidos ao Servidor Web, através do *Web Service Rest* obtendo uma listagem de pedidos que devem ser entregues.

Figura 7 – Tela dispositivo móvel com uma listagem de pedidos

Sincronismo de Dados entre um Sistema Web e um Sistema para Dispositivos Móveis, sem Conexão à Internet



Fonte: Própria do Autor (2018)

4.6 Hardware utilizado para o Servidor

Computador com processador Intel Core 2 Duo E7200, 4GB de memória RAM DDR2, disco rígido de 160GB SATA 2, roteador wireless e tablets 7" com sistema operacional Android.

4.7 Segurança – Acesso ao Sistema Web pelos Dispositivos Móveis

Para acesso ao Sistema Web, os dispositivos móveis devem ser previamente cadastrados, obtendo-se deles o endereço *mac address*. Somente dispositivos cadastrados podem acessar o Sistema Web, conforme a Figura 8.

Figura 8 – Segurança e Acesso ao Sistema Web

A imagem é uma captura de tela de uma tela de cadastro de dispositivos. No topo, há uma barra cinza com o título "Dispositivos" e o status "editando". Abaixo, há um formulário com três campos de texto: "Endereço MAC" (contendo "00:d0:a9:2f:3d:99"), "Marca" (contendo "Genesis") e "Modelo" (contendo "GT-7340"). No rodapé, há dois botões: "Salvar" (em azul) e "Cancelar" (em cinza).

Fonte: Própria do Autor (2018)

Outro recurso utilizado foi obter o endereço *mac address* do roteador Wireless. Com isso foi possível fazer com que os dispositivos móveis pudessem se conectar corretamente somente no roteador utilizado na rede local. A Figura 9 exibe o trecho de código de como isso foi executado.

Figura 9 – Trecho de Código Obtendo o *Mac Address* do Roteador

```
WifiManager wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE);
WifiInfo wifiInfo = wifiManager.getConnectionInfo();
String s = wifiInfo.getBSSID();
String ss = s.replaceAll("[^0-9]", "").substring(2, 6);

macRouter = Integer.parseInt(ss);

if (macRouter == 2127) {
```

Fonte: Própria do Autor (2018)

5. Resultados e Discussões

A utilização de um computador simples e considerado até mesmo obsoleto, a não utilização da Internet móvel e a ausência de impressões em papel, reduziram consideravelmente os custos operacionais para pequenas e microempresas, além de, para uma futura implementação, agilizar o processo de entrega de produtos devido ao controle de rotas que será incorporado a aplicação.

Escolhemos o Linux, em sua distribuição CentOS, por ser considerado um sistema operacional leve, que funciona praticamente em qualquer computador, independente do processador e arquitetura de hardware e possui licença de uso livre baseado na GNU GPL (*General Public License*).

O *Web Service Rest* e o formato JSON mostraram-se eficientes durante o envio e recepção de dados entre o Sistema Web e os dispositivos móveis, não ocorrendo falhas ou corrupção de dados durante os testes.

Ferramentas Web são o padrão de desenvolvimento escolhido para o projeto, com pensamento voltado para o paradigma da Computação em Nuvem, onde acredita-se ser o futuro dos sistemas de informação. Porém, “Por outro lado, há um consenso de que um dos grandes entraves para uma adoção ainda mais abrangente desse paradigma está nas atuais limitações de rede internet”. (ENDLER, VITERBO, FONSECA, 2011).

Os resultados da pesquisa de satisfação e qualidade dos serviços de telefonia móvel das operadoras de modo geral, aplicada por meio de um questionário *online*, no período de 14 de maio de 2018 a 10 de junho de 2018, ainda revelam um alto nível de falhas e altos valores cobrados pelas operadoras.

5.1 Resultados da Pesquisa de Satisfação e Qualidade dos Serviços de Telefonia Móvel

Tabela 1 – Total de participantes por cidade – Total de participantes

Cidade/UF	Total de Participantes por Cidade
Bauru/SP	1
Garça/SP	15
Herculândia	1
Marília/SP	59
Palmital/SP	1

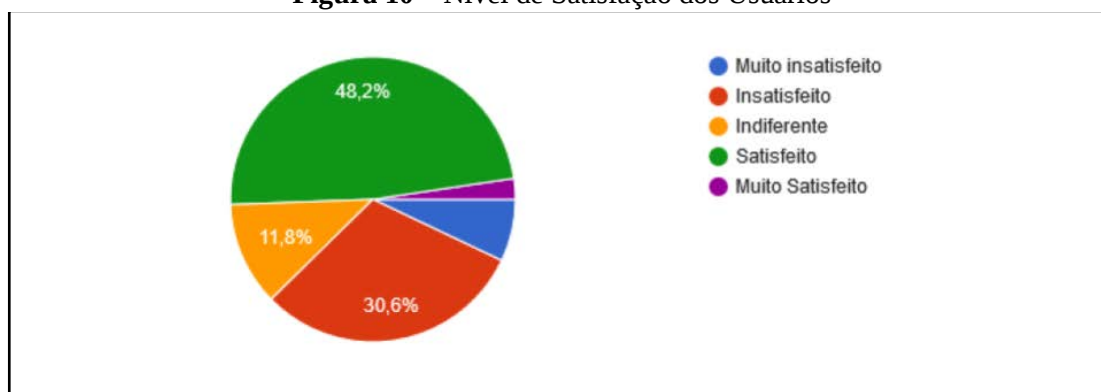
Piracicaba/SP	1
Pompéia/SP	1
Presidente Prudente/SP	1
São Paulo/SP	3
Vera Cruz/SP	2
Total de Participantes	85

Fonte: Própria do Autor (2018)

A pesquisa exibida na Tabela 1 acima, corresponde ao total de 85 participantes e o total de participantes por cidade, destinada a residentes no Estado de São Paulo, visando principalmente as cidades do interior e em particular a cidade de Marília, onde se localiza a microempresa entrevistada.

Quanto ao nível de satisfação dos usuários sobre a qualidade dos serviços prestados, obteve-se um resultado equilibrado, conforme a Figura 8.

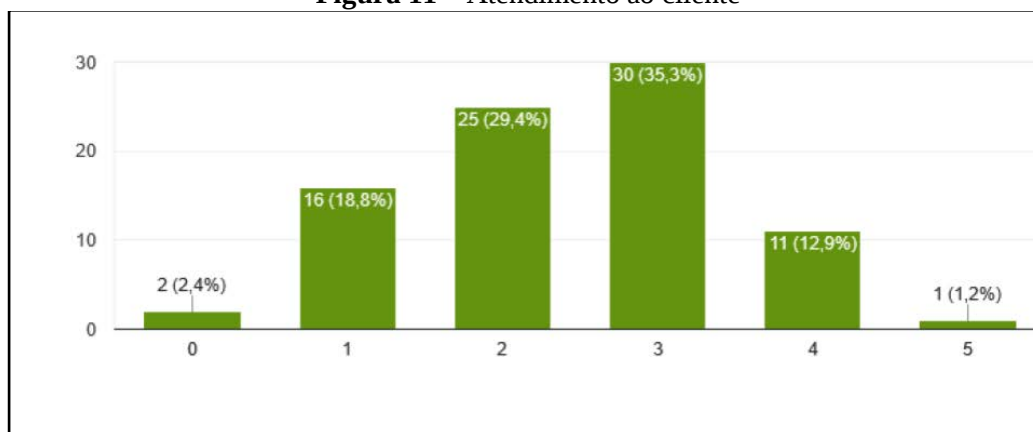
Figura 10 – Nível de Satisfação dos Usuários



Fonte: Própria do Autor (2018)

Sobre o atendimento das operadoras aos clientes, para qualquer tipo de solicitação, onde foram atribuídas notas de 0 a 5, notou-se nota 3 como valor predominante, conforme a Figura 11.

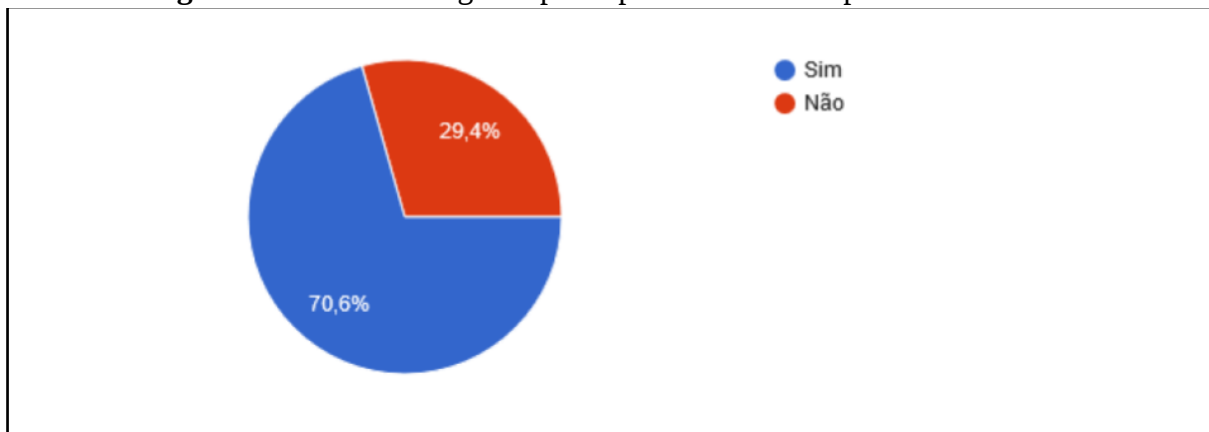
Figura 11 – Atendimento ao cliente



Fonte: Própria do Autor (2018)

Questionados se tiveram algum tipo de problema, como exemplo, desconexão ou ausência de sinal, em um período de 30 dias, 70% disseram que sim conforme a Figura 12.

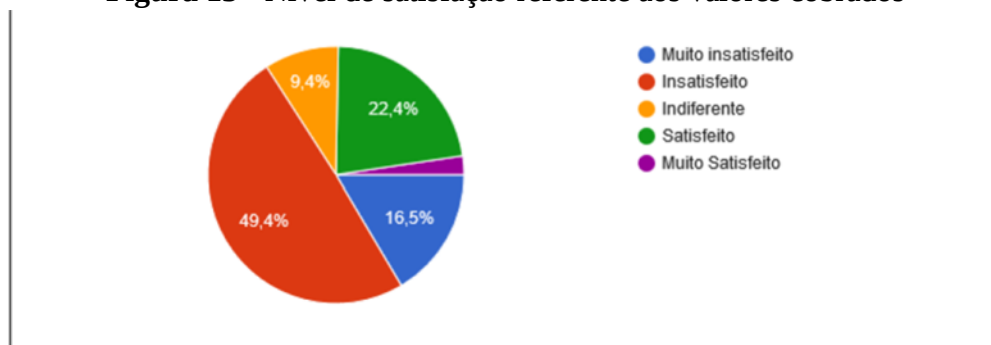
Figura 12 - Se houve algum tipo de problema em um período de 30 dias



Fonte: Própria do Autor (2018)

Quanto aos valores cobrados pelos serviços prestados, notou-se um número elevado de pessoas insatisfeitas ou muito insatisfeitas, conforme a Figura 13.

Figura 13 - Nível de satisfação referente aos valores cobrados



Fonte: Própria do Autor (2018)

REFERÊNCIAS

ANTUNES, Jonathan Lamim. **CodeIgniter: Produtividade na criação de aplicações web em PHP**. São Paulo: Casa do Código, 2016. 1 p.

CABARDO, Ademir C. **Criando um E-commerce com CodeIgniter**. 1.ed. São Paulo: Novatec Editora, 2015. 28 p.

CAPELAS, Bruno. O Estado de São Paulo, ago. 2016. **Internet móvel do Brasil é 57ª mais rápida**. Disponível em: < <http://link.estadao.com.br/noticias/cultura-digital,internet-movel-do-brasil-e-57-mais-rapida,10000069978> >. Acesso em: 10 mar. 2018.

ENDLER, Markus; VITERBO, José; FONSECA, Hubert. **Perspectivas e Desafios da Computação em Nuvem na Internet do Futuro**. 2011. Disponível em: <<http://www-di.inf.puc-rio.br/~endler/paperlinks/TechReports/CPqD-rel3.pdf>>. Acesso em: 12 mar. 2018.

FERREIRA, E. Rubem. **Linux Guia do Administrador do Sistema**. 2. ed. São Paulo: Novatec, 2008. 24 p.

IDEC (Instituto Brasileiro de Defesa do Consumidor), fev. 2016. **Internet Móvel no Brasil**. Disponível em: <<http://idec.org.br/pdf/analise-internet-movel-brasil-2g-3g.pdf>>. Acesso em: 10 mar. 2018.

VANDRESEN, Rogério Schueroff; MAGALHÃES, Willian Barbosa. **Conceitos e Aplicações da Computação em Nuvem**. Universidade Paranaense (UNIPAR). Disponível em: <<http://ftp.unipar.br/~seinpar/2013/artigos/Rogério%20Schueroff%20Vandresen.pdf>>. Acesso em: 10 mar. 2018.

MARTINI, Luciano Andress. **Linux para Servidores: Da instalação à virtualização**. Santa Cruz do Rio Pardo, São Paulo: Editora Viena, 2013. 38 p.

MITCHELL, Lorna Jane. **PHP Web Services**. 2.ed. Estados Unidos, 2016, 48 p.

NETBEANS Docs & Support. **Introdução aos Webservices RESTful**. Disponível em: <https://netbeans.org/kb/docs/websvc/rest_pt_BR.html>. Acesso em 5 abr. 2018.