

UNIVERSIDADE PAULISTA - UNIP

**HENRIQUE MENDONÇA DE SOUZA
JOSÉ VITOR ZANONI DA COSTA
KAIO DA FONSECA GONÇALVES DA SILVA**

**“DESENVOLVIMENTO DE UMA APLICAÇÃO DE SISTEMA DISTRIBUÍDO PARA
DISPOSITIVO MÓVEL”**

SÃO PAULO

2018

**HENRIQUE MENDONÇA DE SOUZA
JOSÉ VITOR ZANONI DA COSTA
KAIO DA FONSECA GONÇALVES DA SILVA**

**“DESENVOLVIMENTO DE UMA APLICAÇÃO DE SISTEMA DISTRIBUÍDO PARA
DISPOSITIVO MÓVEL”**

Atividade prática supervisionada e
apresentada ao curso Ciência da
Computação para fins de conhecimento
na área.

Orientador: Prof. Dr. César Augusto
Cardoso Caetano.

SÃO PAULO
2018

RESUMO

Neste trabalho é apresentada uma proposta de um aplicativo com informações sobre alagamentos. O aplicativo consiste em mapear as áreas que estão alagadas, se o usuário do aplicativo estiver em uma área com alagamento, ele irá informar utilizando o aplicativo através da geolocalização, dessa forma fazendo o mapeamento, se o usuário optar em consultar as áreas que estão alagadas, ele irá consultar o mapa do aplicativo onde informará as áreas alagadas supostamente registradas por outro usuário. Para efetuar o mapeamento das regiões, o aplicativo utiliza a geolocalização do celular do usuário, para que isso ocorra de forma efetiva, foi utilizada a API do Google Maps possibilitando que o celular do usuário seja localizado e mapeado. Dessa forma, obtemos resultados melhores onde os usuários se interagem entre si com troca de informações sobre regiões supostamente alagadas através do aplicativo proposto. Também foi utilizado uma interface web para fazer o armazenamento da latitude e longitude das áreas alagadas possibilitando a marcação das mesmas no mapa do aplicativo. Os resultados obtidos apresentaram-se eficientes para casos de regiões alagadas devido as fortes chuvas.

Palavras-Chave:; api; alagamento; chuva; mapa.

ABSTRACT

This work is one proposal of an application with information about floods. The application consists of mapping the areas that are allocated, whether the user makes an application with flooding, whether to use the application through the window, or who is doing the mapping, if the user is servicing as areas that are flooded, the report previous is the same as doing mapping regions, the application used the user cache, the use of mapping regions, the use of a Google API. Maps allowing the user's phone to be located and mapped. In this way, users are better obtained from the interaction between information operations on the regions supposedly flooded through the application. An interface was also used to store the latitude and longitude of flooded areas, allowing them to be tagged in the application map. The results obtained are very strict for the cases of regions affected due to heavy rains.

Keywords: api; flooding; rain; map.

Lista de Abreviaturas e Siglas

API – *Application Programming Interface.*

APP – *Application.*

ASCII – *American Standard Code for Information Interchange.*

BJSON – *Binary JSON.*

CPU – *Central Process Unit.*

CSS – *Cascading Style Sheets.*

GB – *Gigabyte.*

GHz – *Giga-Hertz.*

GPS – *Global Positioning System.*

HD – *Hard Disk.*

HTML – *HyperText Markup Language.*

HTTP – *Hypertext Transfer Protocol.*

IDC – *International Data Corporation.*

IDE – *Integrated Development Environment.*

JSON – *JavaScript Object Notation.*

MER – *Modelo de Entidade e Relacionamento.*

PWAs – *Progressive Web Apps.*

RAM – *Random Access Memory.*

SASS – *Syntactically Awesome Stylesheets.*

SDK – *Software Development Kit.*

SGBD – *Sistema de Gerenciamento de Banco de Dados.*

SOAP – *Simple Object Access Protocol.*

TB – *Terabyte.*

UML – *Unified Modeling Language.*

URI – *Uniform Resource Identifier.*

XML – *Extensible Markup Language.*

Wi-Fi – *Wireless Fidelity*.

WWW – *World Wide Web*.

Lista de Imagens

Imagem 01 – Operational System Share Worldwide	Pag. 7.
Imagem 02 – Dados.xml	Pag. 20.
Imagem 03 – Dados.txt	Pag. 21.
Imagem 04 – Blocks	Pag. 23.
Imagem 05 – Design	Pag. 24.
Imagem 06 – Funcionamento em browser	Pag. 37.
Imagem 11 – APP 1	Pag. 38
Imagem 08 – APP 2	Pag. 39.
Imagem 09 – APP 3	Pag. 40.
Imagem 10 – APP 4	Pag. 41.
Imagem 11 – APP 5	Pag. 42.
Imagem 12 – Ficha Henrique	Pag. 48.
Imagem 13 – Ficha José	Pag. 49.
Imagem 14 – Ficha Kaio	Pag. 50.

SUMÁRIO

1	OBJETIVO DO TRABALHO.....	5
2	INTRODUÇÃO.....	6
3	CONCEITOS GERAIS	9
3.1	Android.....	9
3.2	JSON.....	9
3.3	API (Application Programming Interface)	10
3.4	Sistemas Distribuídos	10
3.5	Engenharia de Software.....	11
3.6	Web Service.....	11
4	PLANO DE DESENVOLVIMENTO DA APLICAÇÃO	13
4.1	App Inventor.....	13
4.2	XML.....	13
4.3	PHP.....	15
4.4	HTML 5	15
4.5	Bootstrap.....	16
4.6	JavaScript	17
5	PROJETO	18
6	RELATÓRIO COM AS LINHAS DO CÓDIGO DO PROGRAMA	23
6.1	Aplicativo.....	23
6.2	Servidor	25
	geraxml.php.....	35
	salvar.php.....	36
7	APRESENTAÇÃO DO PROGRAMA EM FUNCIONAMENTO.....	37

8 Conclusão.....	43
9 BIBLIOGRAFIA.....	44
10 FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS.....	48

1 OBJETIVO DO TRABALHO

O objetivo principal deste trabalho é desenvolver uma aplicação baseada em sistemas distribuídos para dispositivo móvel que deverá gerenciar as informações ambientais urbanas.

O aplicativo desenvolvido permitirá que os usuários possam visualizar em um mapa onde existem pontos de alagamento, caso o mesmo esteja em meio a esta situação, ele poderá usar o aplicativo para informar aos outros usuários que aquela área também está alagada. As informações a respeito da localização dos usuários serão obtidas através da API do Google Maps, que é responsável por capturar as coordenadas do usuário quando o aplicativo é executado. Sempre que os usuários acessarem o aplicativo para informar que estão em um ponto de alagamento, as informações serão gravadas em um arquivo XML, que é a fonte de informações que serão fornecidas na interface a todos os utilizadores do serviço.

Os smartphones deverão possuir sistema operacional Android para utilizar o App, que foi escolhido por ser amplamente utilizado no mundo, e possuir uma interface agradável e intuitiva. Dispositivos com outros sistemas operacionais (iOS, Windows Phone e etc.) poderão utilizar o serviço através do Navegador web (Browser). Para que o sistema funcione corretamente é necessário que o dispositivo utilizado pelo usuário possua o recurso de GPS.

2 INTRODUÇÃO

Em um mundo globalizado, competitivo e de mudanças complexas, os aparelhos smartphones estão sendo utilizados em grande escala e tem sido cada vez mais importante na vida dos usuários.

De acordo com dados do IDC – International Data Corporation de 2013, a plataforma Android possuem aproximadamente 65% da fatia de mercado, o que representa o grande aceite da população e confiabilidade na plataforma de sistema, destacando ainda para o fato dos usuários possuírem planos e pacotes que os permitem acessar a internet através das coberturas 3 e 4G, confirmando a tendência de mercado e a inclusão tecnológica da população.

Com isso, a necessidade de desenvolver aplicativos de qualidade e que atendam as exigências, necessidades e a melhoria na qualidade de vida dos usuários vêm crescendo exponencialmente.

Os sistemas de informação na internet têm passado por uma expansão fantástica, acompanhando o crescimento da rede. Os portais fornecem informações sobre diversos ambientes de trabalho: Livrarias fornecem informações sobre obras e publicações, portais e notícias fornecem informações diversas, grandes magazines lançaram lojas virtuais que fazem a venda de seus produtos e recebem o pagamento via eletrônica.

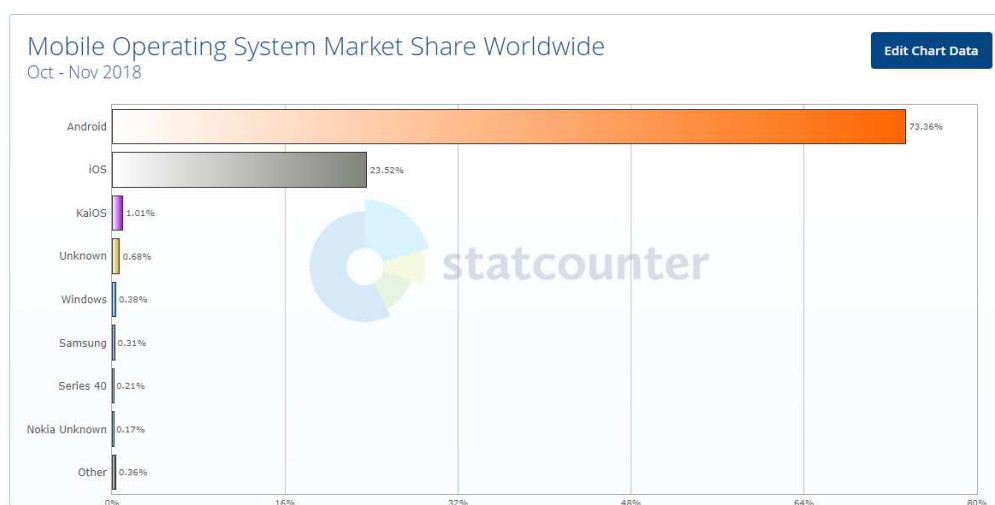
Seguindo os preceitos da Engenharia de Software, Qualidade de Software e Sistemas Distribuídos, será desenvolvida uma aplicação para dispositivo móvel que informará aos usuários os pontos de alagamentos através de um mapa. Este serviço será disponibilizado utilizando Web Service.

Antes de nos aprofundarmos nos conceitos e tecnologia de Web Services, vejamos um pouco sobre sua evolução: No ano de 2000, a W3C (World Wide Web Consortium) aceitou a submissão do Simple Object Access Protocol (SOAP). Este formato de mensagem baseado em XML estabeleceu uma estrutura de transmissão para comunicação entre aplicações (ou entre serviços) via HTTP. Sendo uma tecnologia não amarrada a fornecedor, o SOAP disponibilizou uma alternativa atrativa em relação aos protocolos proprietários tradicionais, tais como CORBA e DCOM.

No decorrer do ano seguinte, o W3C publicou a especificação WSDL. Uma nova implementação do XML, este padrão forneceu uma linguagem para descrever a interface dos Web Services. Posteriormente suplementada pela especificação UDDI (Universal Description, Discovery and Integration), que proporcionou um mecanismo padrão para a descoberta dinâmica (dynamic discovering) de descrições de serviço, a primeira geração da plataforma de Web Services foi estabelecida.

Hoje o sistema operacional Android é a plataforma de dispositivo móvel mais utilizada no mundo, conforme demonstra o gráfico abaixo:

Figura 1: Operational System Share Worldwide



Fonte: Gs StatCounter, 2018.

Com base nas informações obtidas na pesquisa sobre a utilização de dispositivos móveis, o App será desenvolvido para smartphones com sistema operacional Android, pois atenderá um número maior de usuários. Dispositivos com outros sistemas operacionais (iOS, Windows Phone e etc.) poderão utilizar o serviço através do Navegador web (Browser). Para que o sistema funcione corretamente é necessário que o dispositivo utilizado pelo usuário possua o recurso de GPS.

3 CONCEITOS GERAIS

3.1 Android

O sistema operacional Android foi criado pela Android, Inc., empresa estabelecida em 2003 na Califórnia (dentro do célebre Vale do Silício), com o intuito de desenvolver soluções de software para plataformas móveis. O Android é baseado no kernel (núcleo) do sistema operacional Linux.

Dessa forma, tornou-se (para desespero de muitos fãs “radicais” do Linux) a versão (ou “distribuição” – apesar de muitos não concordarem que o Android seja uma distribuição Linux “de facto”) mais usada no mundo do sistema operacional idealizado por Linus Torvalds.

Com a variedade de recursos do Android, seria fácil confundi-lo com um sistema operacional desktop. O Android é um ambiente em camadas baseado em kernel Linux e que inclui funções ricas. O subsistema da UI inclui:

- Janelas
- Visualizações
- Widgets para a exibição de elementos comuns como caixas de edição, listas e listas suspensas

3.2 JSON

JSON é um modelo para armazenamento e transmissão de informações em formato de texto e completamente independente de linguagem, é baseado em um subconjunto da linguagem JavaScript.

Para cada valor representado, atribui-se um nome (ou rótulo) que descreve seu significado.

O JSON é constituído em duas estruturas:

- Uma coleção de pares nome/valor.
- Uma lista ordenada de valores.

Apesar do XML ser uma linguagem mais rica e possuir maior grau de maturidade, o JSON representa informações de forma mais compacta, rápida e simples.

3.3 API (Application Programming Interface)

Uma API (Application Programming Interface) é um conjunto de comandos, funções, protocolos e objetos que os programadores podem usar para criar softwares ou interagir com sistemas externos. Ela fornece aos desenvolvedores comandos para executar operações comuns para que eles não tenham que escrever um código do zero.

Em suma, é o elemento responsável por garantir que você experimente uma fluidez perfeita de conexão. Ou seja, sem interrupções: em nenhum momento o usuário percebe que está vinculado a várias redes e transferindo dados através de diversos servidores.

3.4 Sistemas Distribuídos

Segundo TANENBAUM (2008), um Sistema Distribuído é uma coleção de computadores independentes que aparecem para o usuário como um único computador. Os Sistemas Distribuídos são baseados na Troca de Mensagens, a

diferença mais importante entre Sistemas Distribuídos e Sistemas Centralizados é o IPC (Interprocess Communication) e o fato de que nos Sistemas Centralizados a memória é partilhada.

3.5 Engenharia de Software

Existem vários conceitos de Engenharia de Software, segundo ROGER S. PRESSMAN (2011), Engenharia de Software é o conjunto de métodos, procedimento e ferramentas com o objetivo de construir software com qualidade.

Segundo NOGUEIRA (2003), podemos concluir que a Engenharia de Software é um conjunto de práticas para desenvolvimento de soluções para software.

3.6 Web Service

Em 1999, surgiu a necessidade de se padronizar a comunicação entre diferentes plataformas e linguagens de programação. Diversos padrões foram propostos, mas nenhum obteve sucesso. O Web Service surgiu como uma solução para melhorar a comunicação entre os sistemas distribuídos.

A principal característica do Web Service é a interoperabilidade, ou seja, a interoperabilidade garante que as aplicações possam trocar e processar dados geridos por outras aplicações. Portanto, Web Service é um conjunto de protocolos e padrões que servem para trocar dados entre aplicações.

O Web Service é uma classe escrita em uma linguagem suportada pela plataforma .NET que pode ser acessada via protocolo HTTP. O acesso sempre será via HTTP, mas internamente existe uma string XML que está empacotada em um protocolo SOAP (Simple Object Access Protocol). O SOAP é um modo padronizado para a transferência de dados em diversas aplicações, em XML.

4 PLANO DE DESENVOLVIMENTO DA APLICAÇÃO

4.1 App Inventor

O App Inventor é uma ferramenta originalmente criada e posteriormente cedida pelo Google ao MIT. Trata-se de uma plataforma de desenvolvimento, que parte da premissa de facilitar a criação de aplicativos, mesmo para aqueles mortais sem conhecimento avançado de programação, roda quase inteiramente no navegador. Com ele é possível criar aplicativos (ou apps), de forma simplificada. Tudo que você tem a fazer é apenas arrastar os ícones correspondentes para o espaço que simula a tela do aparelho com Android.

Com o App Inventor foi construído o aplicativo para o sistema operacional Android.

4.2 XML

O XML (eXtensible Markup Language) é um recurso que nos permite armazenar dados de forma estruturada e que serve como transporte para esses dados entre camadas diferentes.

O segredo do seu sucesso é ser um padrão aberto e amplamente aceito pela comunidade desenvolvedora.

Características do XML:

- Pode armazenar e organizar qualquer tipo de informação;
- possui padrão aberto, não sendo “amarrado” a nenhum aplicativo ou desenvolvedor;
- como o Unicode é o seu character set padrão, pode armazenar uma gama variada de símbolos e caracteres de outros idiomas;

- o XML oferece diversas maneiras de checar a qualidade e integridade do documento, usando regras de sintaxe, checagem de links internos (estrutura), comparação com modelos de documentos e definição de tipos de dados;
- formado por uma estrutura simples e não ambígua, o XML é ideal para ser lido e manipulado por pessoas e aplicativos;
- pode ser facilmente combinado com folhas de estilo (CSS) para apresentar os dados na estrutura desejada.

O XML é responsável por armazenar as coordenadas e o endereço que serão obtidos dos usuários que informarem áreas de alagamento, estas informações também serão utilizadas pelo mapa do Google, que exibirá os pontos de alagamento a partir desses dados.

4.3 PHP

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) 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. O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também tem suporte à maioria dos servidores web atualmente. Isso inclui o Apache, o IIS e muitos outros. E isso inclui qualquer servidor web que possa utilizar o binário FastCGI do PHP, como o lighttpd e o nginx. O PHP trabalha tanto como módulo quanto como um processador CGI. O PHP é responsável por receber as informações obtidas através da API do Google Maps e gerar o arquivo XML.

4.4 HTML 5

HTML (HyperText Markup Language) é o componente mais básico da web. Ela serve para definir o conteúdo e a estrutura básica de uma página web. Além do HTML, em geral outras tecnologias são usadas para descrever a apresentação/aparência (CSS) ou funcionalidade/comportamento (JavaScript) das páginas web.

HTML5 é a mais recente evolução do padrão que define o HTML. O termo representa dois conceitos diferentes:

- É uma nova versão da linguagem HTML, com novos elementos, atributos, e comportamentos

- e um conjunto maior de tecnologias que permite o desenvolvimento de aplicações e web sites mais diversos e poderosos. Este conjunto é chamado HTML5 & friends e muitas vezes abreviado apenas como HTML5.

Criada para ser utilizável por todos os desenvolvedores da Web Aberta, essa página de referências faz ligações a inúmeros recursos do HTML5, classificados em diversos grupos, baseando-se em suas funções;

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

O HTML5 foi utilizado para construir a interface do sistema.

4.5 Bootstrap

Bootstrap é o framework criado por Mark Otto e Jacob Thornton, ambos engenheiros do Twitter, é uma coleção de ferramentas para criação de aplicações web utilizando o HTML e CSS.

O Twitter Bootstrap foi utilizado para tornar a aplicação web responsiva a qualquer dispositivo.

4.6 JavaScript

É uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como node.js, Apache CouchDB e Adobe Acrobat. O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional).

O JavaScript foi utilizado para consumir o serviço do Google que retorna o endereço do usuário a partir das coordenadas obtidas pelo GPS do dispositivo. Também é responsável por consumir os dados gravados pelo PHP em XML e informar os pontos de alagamento no mapa utilizando o objeto 'XMLHttpRequest'.

5 PROJETO

Para desenvolvimento do sistema foi necessário obter informações das funcionalidades disponíveis para serem utilizadas, fornecidas pelo Google Maps API v3.

A biblioteca do Google Maps API deve ser declarada da seguinte maneira para que as classes fiquem disponíveis para serem utilizadas:

```
<script src=
http://maps.google.com/maps/api/js?
type="text/javascript"></script>
```

Classe: *google.maps.Geocoder;*

Descrição: Cria uma nova instância de um Geocoder que envia pedidos geocode para os servidores do Google.

Classe: *navigator.geolocation;*

Descrição: O método *Geolocation.getCurrentPosition()* é utilizado para capturar a posição atual do dispositivo. É utilizado também para verificar se o dispositivo suporta a utilização do método.

Classe: *google.maps.LatLng;*

Descrição: A *LatLng* é um ponto em coordenadas geográficas: latitude e longitude. Latitude varia entre -90 e 90 graus, inclusive. Valores acima ou abaixo deste intervalo será fixada ao intervalo [-90, 90]. Isto significa que, se o valor especificado é inferior a -90, ele será ajustado para -90. E se o valor for superior a 90, vai ser definido como 90. Longitude varia entre -180 e 180 graus. Valores acima ou abaixo deste

intervalo será envolvido de forma que eles fiquem dentro do intervalo. Por exemplo, um valor de -190 será convertido em 170. Um valor de 190 será convertido para -170. Isso reflete o fato de que longitudes se envolvem ao redor do globo. Embora o mapa padrão associados projeção longitude com a coordenada x do mapa, e latitude com a coordenada y, a coordenada da latitude é sempre escrito em primeiro lugar, seguido pela longitude. Observe que você não pode modificar as coordenadas de um LatLng. Se você quiser computar outro ponto, você tem que criar um novo objeto. A maioria dos métodos que aceitam objetos LatLng também aceitam um objeto LatLngLiteral, de modo que a seguir são equivalentes: `map.setCenter (novo google.maps.LatLng (- 34, 151)); map.setCenter ({lat: -34, lng: 151});` O construtor também aceita objetos literais, e converte-os às instâncias do LatLng: `myLatLng = new google.maps.LatLng ({lat: -34, lng: 151})`

Classe: *google.maps.Marker;*

Descrição: Cria um marcador com as opções especificadas. Se for especificado um mapa, o marcador é adicionado ao mapa sobre construção. Note-se que a posição deve ser definida para o marcador para exibir.

Classe: *google.maps.MapEventListener;*

Descrição: Esta classe é opaca. Ele não tem métodos e nenhum construtor. Seus casos são devolvidos a partir `addListener()`, `addDomListener()` e, eventualmente, são passados de volta para `removeListener()`.

Classe: *google.maps.Map;*

Descrição: Cria um novo mapa dentro do recipiente HTML dado, que é geralmente um elemento DIV.

Para inserir os marcadores com a função `google.maps.Marker` é utilizada `XMLHttpRequest()` que realiza requisições HTTP para acessar os dados do arquivo XML gerado pelo PHP.

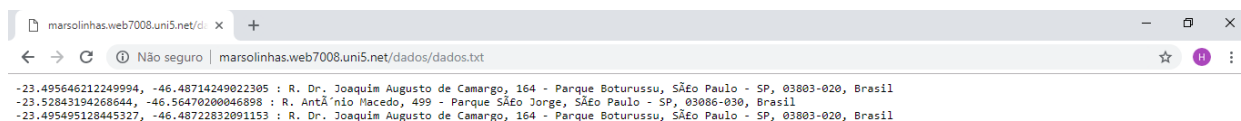
Durante criação do arquivo de dados XML é utilizada a classe DOMDocument do PHP que cria o arquivo dados.xml a partir do arquivo dados.txt que contém as informações obtidas dos usuários.

Figura 2: dados.xml



Fonte: Próprio Autor, 2018.

Figura 3: dados.txt



Fonte: Próprio Autor, 2018.

Quando o usuário clica no botão “Área alagada”, as informações obtidas através da utilização do Google Maps API são validadas no arquivo salvar.php, que validará se os campos do formulário estão vazios ou não. Se as informações forem obtidas com sucesso, as informações serão salvas no arquivo dados.txt. Caso algum erro ocorra durante este procedimento, o usuário será informado.

A classe Mobile_Detect é responsável por garantir o acesso ao sistema apenas em dispositivo móvel.

No desenvolvimento do App para sistema operacional Android, foi utilizado o componente `WebView` onde foi declarada a URL: <http://marsolinhas.web7008.uni5.net/>.

Dispositivos com outros sistemas operacionais (iOS, Windows Phone e etc.) poderão utilizar o serviço através do Navegador web (Browser).

6 RELATÓRIO COM AS LINHAS DO CÓDIGO DO PROGRAMA

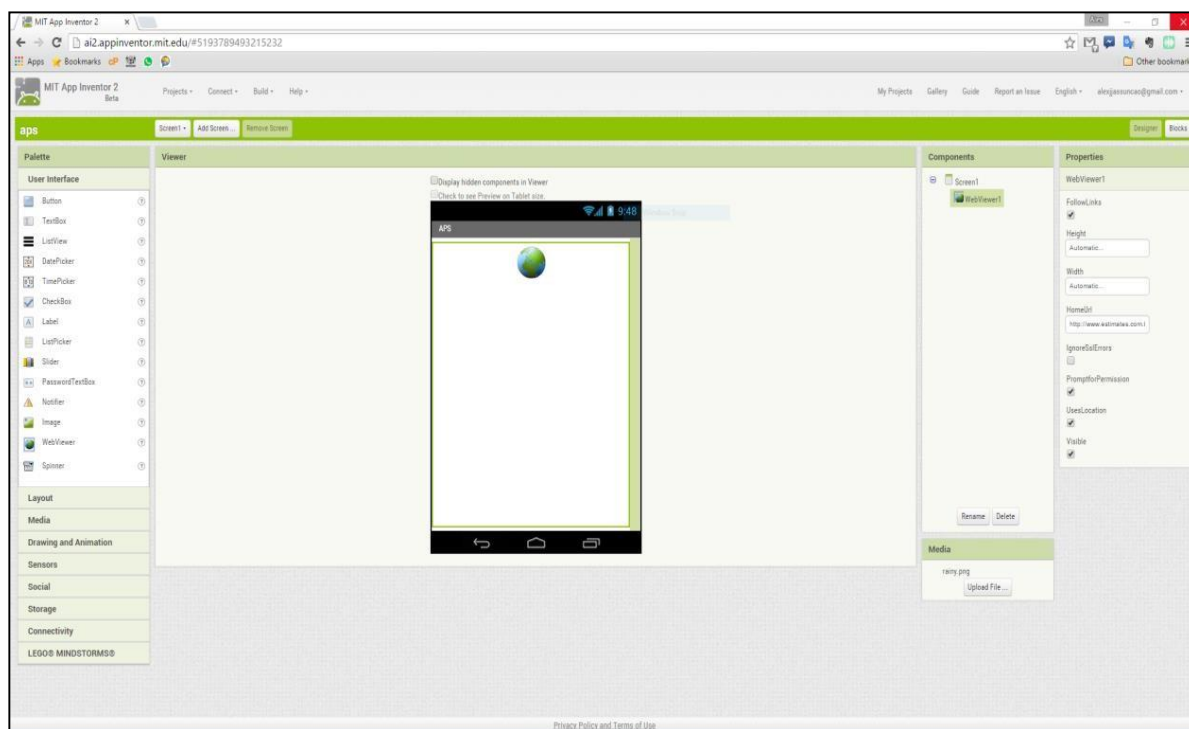
6.1 Aplicativo

Figura 4: Blocks



Fonte: Próprio Autor, 2018.

Figura 5: Design



Fonte: Próprio Autor, 2018.

6.2 Servidor

index.php

```
<?php
```

```
    require_once('lib/geraxml.php');  
    require_once('lib/Mobile_Detect.php');  
    $detect = new  
    Mobile_Detect;
```

```
?>
```

```
<!doctype html>
```

```
<html lang=pt-BR>
```

```
<head>
```

```
<meta name=robots content="noindex, nofollow"/>
```

```
<link rel=icon href=img/rainy.png>
```

```
<meta charset=UTF-8>
```

```
<meta http-equiv=X-UA-Compatible content="IE=edge">
```

```
<meta name=viewport content="width=device-width, initial-scale=1">
```

```
<title>APS - Mapa do alagamento</title>
```

```
<meta name=description content="APS"/>
```

```
<link href=css/bootstrap.min.css rel=stylesheet>
```

<!--[if lt IE 9]>

<script src=http://html5shiv-printshiv.googlecode.com/svn/trunk/html5shiv-printshiv.js></script>

<![endif]-->

</head>

<body>

<?php if (\$detect->isMobile()):?>

<div class=container>

<div class="col-lg-12 col-md-12 page-header">

<h1>APS</h1>

</div>

</div>

<div class=container>

<div class="panel panel-default">

<div class=panel-heading>Informações obtidas via Google Maps API</div>

<div class=panel-body>

<form action=lib/salvar.php class=col-xs-12 method=post>

<p class="help-block text-uppercase">

<small>Você está aqui</small>

</p>

<div class=form-group>

<label for=latitude>Latitude</label>

<input type=text class=form-control name=latitude id=latitude placeholder=Latitude
readonly=readonly>

</div>

<div class=form-group>

<label for=longitude> Longitude</label>

<input type=text class=form-control name=longitude
 id=longitude placeholder=Longitude readonly=readonly>

</div>

<div class=form-group>

<label for=longitude> Endereço</label>

<input type=text class=form-control name=endereco
 id=endereco placeholder=Endereço readonly=readonly>

</div>

<div class=form-group>

```
<button class="btn btn-primary getLocation"><span class="glyphicon glyphicon-exclamation-sign" aria-hidden=true></span> Área alagada
```

```
</button>
```

```
<p class=help-block>Clique no botão acima para informar que você está em uma área de
```

```
alagamento.</p>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class=container>
```

```
<div class="panel panel-default">
```

```
<div class=panel-heading>Mapa do alagamento</div>
```

```
<div class=panel-body>
```

```
<p class=help-block>Pontos de alagamento: <span id=pontos class=badge></span></p>
```

```
<div id=map class=col-xs-12 style=height:450px></div>
```

```
</div>
```

```
</div>
```

</div>

<div id=myModal class="modal fade bs-example-modal-sm" tabindex=-1
role=dialog aria-labelledby=mySmallModalLabel>

<div class="modal-dialog modal-sm">

<div class=modal-content>

<div class=modal-header>

```
<button type=button class=close data-dismiss=modal aria-label=Close><span aria-  
hidden=true>x</span></button>
```

```
<h4 class=modal-title id=mySmallModalLabel>APS</h4>
```

```
</div>
```

```
<div class=modal-body>
```

```
<p id=msg></p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script
```

```
src=https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js></script>
```

```
<script src=http://maps.google.com/maps/api/js? type=text/javascript></script>
```

```
<script src=js/bootstrap.min.js></script>
```

```
<?php if (isset($_GET['null']) == true): ?>
```

```
<script>$(window).load(function(){modal.innerHTML="Não conseguimos obter suas  
coordenadas, tente novamente.";$("#myModal").modal("show");});</script>
```

```
<?php endif; ?>
```

```
<script>$(document).ready(function(){var
```

```
h=new
```

```
google.maps.Geocoder;modal=document.getElementById("msg");latitude=docume
```

```

nt.
getElementById("latitude");longitude=document.getElementById("longitude");ender
ec
o=document.getElementById("endereco");pontos=document.getElementById("pont
os
");if(navigator.geolocation){navigator.geolocation.getCurrentPosition(d)}else{modal.
in nerHTML="O          seu          navegador          não
          suporta
Geolocalização.";$("#myModal").modal("show")}function          d(i){var
          j=new
google.maps.LatLng(i.coords.latitude,i.coords.longitude);latitude.value=i.coords.lati
tu de;longitude.value=i.coords.longitude;b=new
          google.maps.Marker({position:new
google.maps.LatLng(parseFloat(i.coords.latitude),parseFloat(i.coords.longitude)),m
a p:g});google.maps.event.addListener(b,"click",(function(k,l){return
function(){e.setContent("Seu

```

```

local");e.open(g,k)}})(b,c));h.geocode({location:j},function(l,k){if(k===google.maps.
Ge
ocoderStatus.OK){if(l[1]){endereco.value=l[1].formatted_address}else{window.alert
(" Nenhum resultado encontrado")}}else{window.alert("Falha:  "+k)}}})var
    g=new
google.maps.Map(document.getElementById("map"),{zoom:10,scrollwheel:false,ce
nt er:new                google.maps.LatLng(-
23.5483498,-
46.3801577),mapTypeId:google.maps.MapTypeId.ROADMAP});var
    e=new google.maps.InfoWindow();var
    h=new google.maps.Geocoder;var b,c;var
    f=new
XMLHttpRequest();f.onreadystatechange=function(){if(f.readyState==4&&f.status=
=200){a(f)}};f.open("GET","dados/dados.xml",true);f.send();function
    a(j){var
r=j.responseXML;var l=[];var k=[];var o=[];var p=[];var n=[];var q=[];var m=[];var
i=[];for(c=0;c<r.getElementsByTagName("coordenadas").length;c++){l[c]=r.getElem
e
ntsByTagName("coordenadas")[c];k[c]=l[c].childNodes[0].nodeValue;q[c]=k[c].split(":
");n[c]=q[c][0].split(",");pontos.innerHTML=k.length;o[c]=n[c][0];p[c]=n[c][1];m[c]=r.ge
tElementsByTagName("endereco")[c];i[c]=m[c].childNodes[0].nodeValue;b=new
google.maps.Marker({position:new
google.maps.LatLng(parseFloat(o[c]),parseFloat(p[c])),map:g,icon:"img/rainy.png"}
);g oogle.maps.event.addListener(b,"click",(function(s,t){return
function(){e.setContent("Coordenadas:  "+o[t]+"    "+p[t]+"    <br>
                                Endereço:
"+i[t]);e.open(g,s)}})(b,c))));</script>
<?php else: ?>

<div class=container>

```

```
<div class="col-lg-12 col-md-12 page-header">
```

```
<h1 class=text><strong>Este sistema funciona apenas em  
dispositivo móvel.</strong></h1>
```

```
</div>
```

```
</div>
```

```
<?php endif ;?>
```

```
</body>
```

```
</html>
```

geraxml.php

```
<?php
```

```
$coordenadas = file('dados/dados.txt');
```

```
$dom = new DOMDocument('1.0', 'UTF-8');
```

```
$data = $dom->createElement('data');
```

```
$dom->appendChild($data);
```

```
foreach ($coordenadas as $coordenada) {
```

```
    $dados = explode(":", $coordenada);
```

```
    $local = $dom->createElement('local');
```

```
    $cords = $dom->createElement('coordenadas', $dados[0]);
```

```
    $endereco = $dom->createElement('endereco', $dados[1]);
```

```
    $local->appendChild($cords);
```

```
    $local->appendChild($endereco);
```

```
    $data->appendChild($local);
```

```
}
```

```
file_put_contents('dados/dados.xml', $dom->saveXML());
```


salvar.php

```
<?php
```

```
if ($_POST['latitude'] != null && $_POST['longitude'] != null && $_POST['endereco']  
!= null):
```

```
    $f = fopen("../dados/dados.txt", "a+", 0);
```

```
    $linha = $_POST['latitude'] . " , " . $_POST['longitude'] . " : " . $_POST['endereco']  
    . "\n";
```

```
    fwrite($f, $linha,
```

```
    strlen($linha)); fclose($f);
```

```
    echo "<script>window.location='../index.php';</script>";
```

```
else:
```

```
    echo
```

```
    "<script>window.location='../index.php?null=true';</script>";
```

```
endif;
```

7 APRESENTAÇÃO DO PROGRAMA EM FUNCIONAMENTO

7.1 Apresentação do programa em funcionamento em um computador

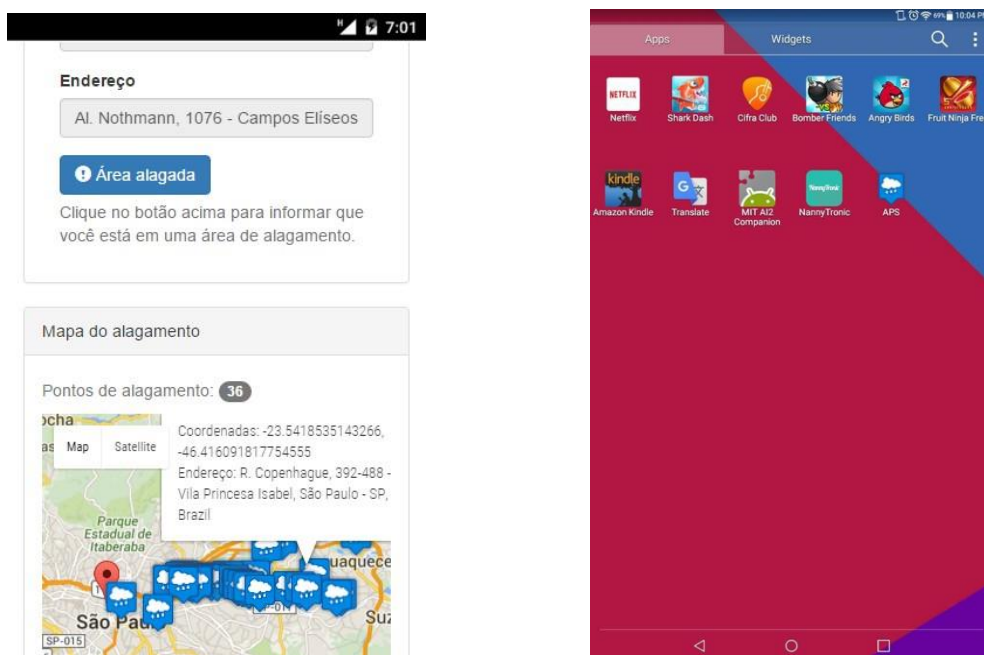
Imagem 6: Funcionamento em browser



Fonte: Próprio Autor, 2018.

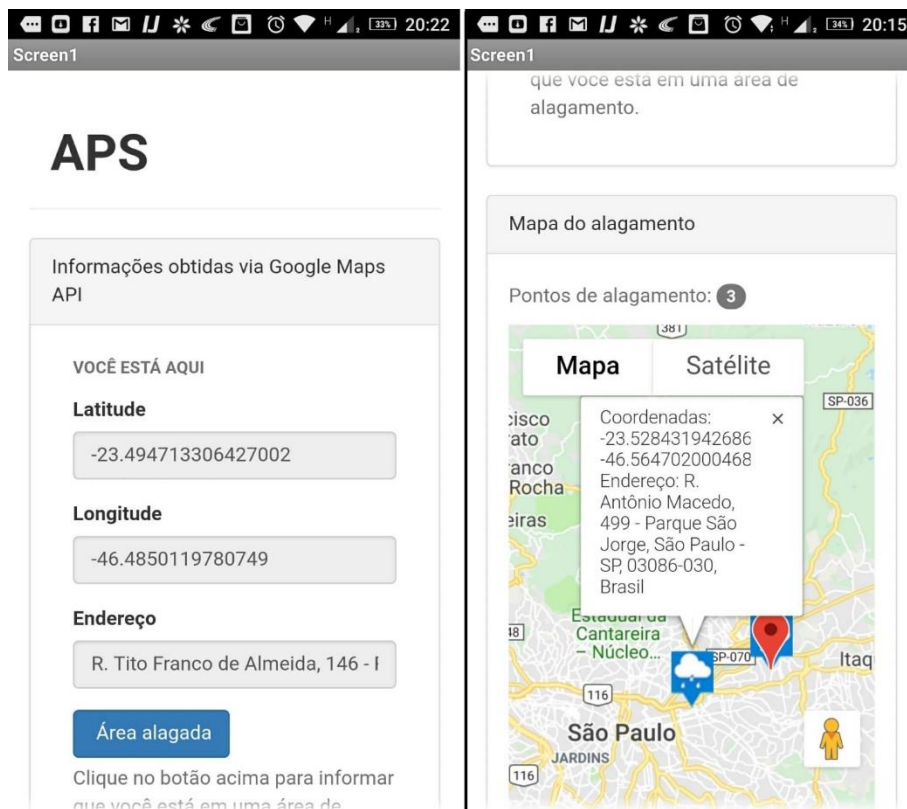
7.2 Apresentação do programa em funcionamento em um dispositivo móvel

Imagem 7: APP 1



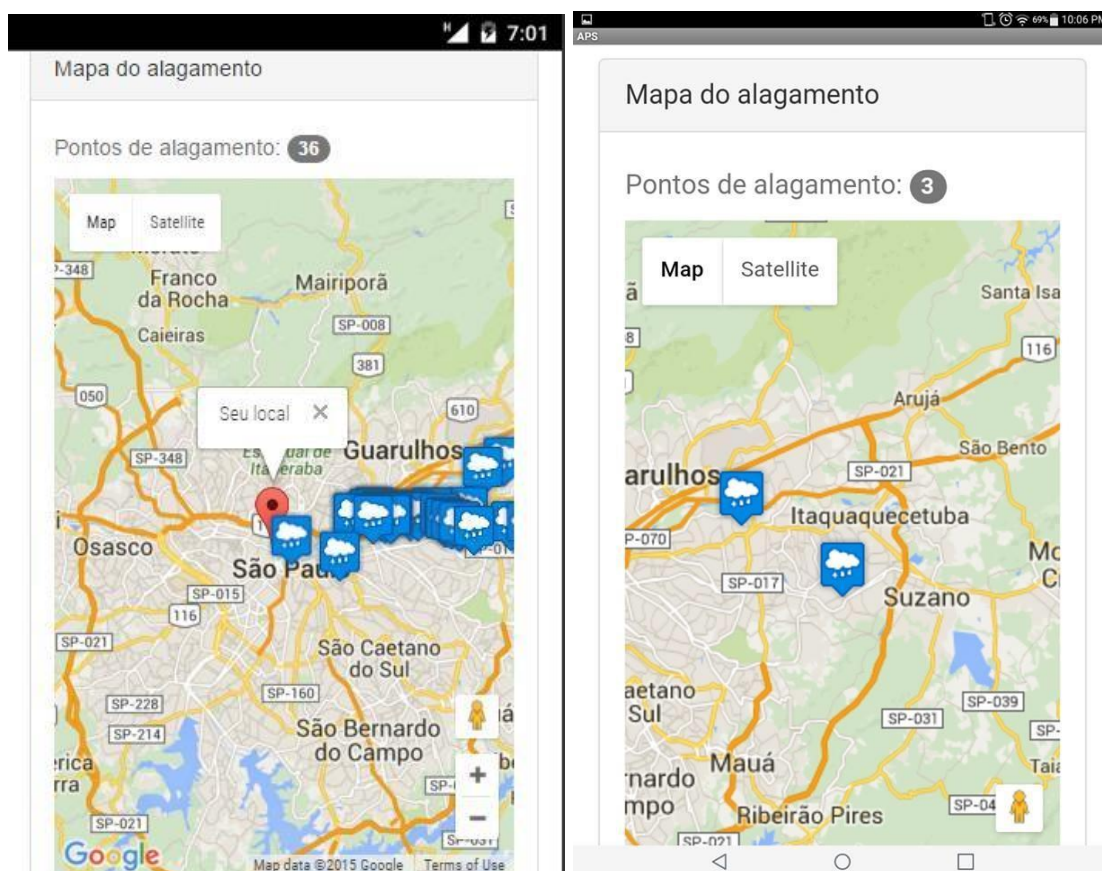
Fonte: Próprio Autor, 2018.

Imagem 8: APP 2



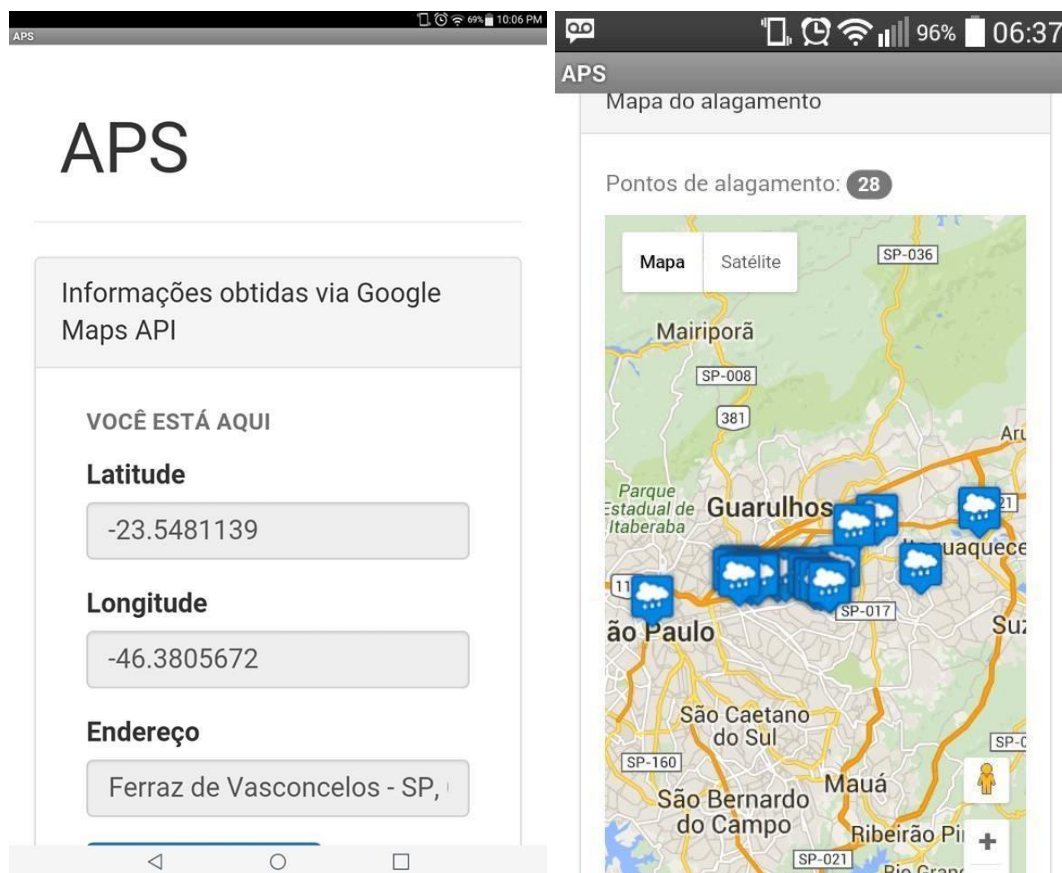
Fonte: Próprio Autor, 2018.

Imagem 9: APP 3



Fonte: Próprio Autor, 2018.

Imagem 10: APP 4



Fonte: Próprio Autor, 2018.

Imagem 11: APP 5



Fonte: Próprio Autor, 2018.

8 Conclusão

Neste trabalho abordamos o assunto Sistemas distribuídos para o desenvolvimento de uma aplicação em dispositivos móveis, onde foi criado um aplicativo que tem como objetivo gerenciar as informações ambientais urbanas, tais como chuvas fortes, tempestades, trovões, alagamentos, etc. No aplicativo é possível verificar em um mapa os pontos onde há divergência ambiental.

Criado para funcionar apenas em dispositivos móveis com sistema Android, foi utilizada a linguagem de programação PHP e também API do Google Maps para capturar coordenadas.

Conclui-se que o desenvolvimento do projeto proporcionou um estudo completo sobre Sistemas distribuídos, possibilitando o grupo a desenvolver as técnicas necessárias para a criação de um aplicativo aceitável no mercado de trabalho utilizando somente os recursos necessários garantindo assim o custo e a efetividade do software em questão.

Os métodos aprendidos em aulas foram extremamente fundamentais para o desenvolvimento e conclusão do projeto.

Portanto os conceitos de Sistemas Distribuídos são para o desenvolvimento de uma aplicação móvel que atenda as necessidades do usuário e que tenha um ótimo desempenho.

9 BIBLIOGRAFIA

Engenharia de Software, Uma abordagem profissional. Disponível em:

<<https://fateczlads.files.wordpress.com/2014/08/engenharia-de-software-7c2b0-edic3a7c3a3o-roger-s-pressman-capc3adtulo-1.pdf>>

Acesso em: 10/10/2018

Tanenbaum, Andrew Stuart. *Sistemas Distribuídos: princípios e paradigmas*. 2 ed. Pearson, 2015.

Introdução ao Desenvolvimento Android. Disponível em:

<<https://www.ibm.com/developerworks/br/library/os-android-devel/index.html>>.

Acesso em: 11/10/2018.

Introdução ao Android. Disponível em: <<https://www.devmedia.com.br/introducao-ao-android/28375>>. Acesso em: 11/10/2018

Saiba o que é e para que serve uma API. Disponível em:

<<https://usemobile.com.br/o-que-e-uma-api/>>. Acesso em: 11/10/2018

JSON. Disponível em: <<http://www.json.org/json-pt.html>>. Acesso em: 09/10/2018.

Introdução ao Formato JSON. Disponível em:

<<http://www.devmedia.com.br/introducao-ao-formato-json/25275>>. Acesso em: 09/10/2018.

Sistemas Distribuídos. Disponível em:

<<http://www.cin.ufpe.br/~sd/disciplinas/sd/pos/aulas/Apresentacao.pdf>>. Acesso

em: 09/10/2018.

Sistemas Distribuídos. Disponível em:

<<http://www.noginfo.com.br/moodle/mod/resource/view.php?id=3746>>. Acesso em: 09/10/2018.

Engenharia de Software. Disponível em:

<<http://www.noginfo.com.br/moodle/course/view.php?id=2>>. Acesso em: 09/10/2018.

“A Implementação da Engenharia de Requisitos como fator crítico de sucesso para as pequenas empresas de desenvolvimento de software”. Disponível em: <<http://www.noginfo.com.br/moodle/mod/resource/view.php?id=3>>. Acesso em: 09/10/2018.

“Metodologia de Desenvolvimento de Sistemas de Informação para Internet”. Disponível em:

<<http://www.noginfo.com.br/moodle/mod/resource/view.php?id=7>>. Acesso em: 09/10/2018.

“Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI - Parte1”. Disponível em: <<http://www.devmedia.com.br/introducao-as-tecnologias-web-services-soa-soap-wsdl-e-uddi-parte1/2873>>. Acesso em: 09/10/2018.

“Web Services com REST e SOAP”. Disponível em:

<<http://pt.slideshare.net/CCCarl/web-service-soap-vs-rest>>. Acesso em: 09/10/2018.

“Web Services”. Disponível em:

<<https://msdn.microsoft.com/pt-br/library/cc564893.aspx>>.

Acesso em: 09/10/2018.

“São Paulo e as enchentes”. Disponível em:

<<http://planetasustentavel.abril.com.br/noticia/cidade/sao-paulo-enchentes-causas-tratamento-acoes-678098.shtml>>. Acesso em: 09/10/2018.

“App Inventor”. Disponível em: <<http://appinventor.mit.edu/explore/about-us.html>>. Acesso em: 09/10/2018.

Introdução ao XML. Disponível em: <<https://www.devmedia.com.br/introducao-ao-xml-parte-01-03/8231>>. Acesso em: 09/10/2018.

HTML. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 09/10/2018.

HTML5. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>>. Acesso em: 10/10/2018

O que é o PHP?. Disponível em: <https://secure.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 11/10/2018.

O que o PHP pode fazer?. Disponível em:

<https://secure.php.net/manual/pt_BR/intro-whatcando.php>. Acesso em: 10/10/2018.

JavaScript. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 11/10/2018

“O que é o Framework Twitter Bootstrap ?”. Disponível em:

<<http://www.vrsys.com.br/blog/15-tecnologia/58-o-que-e-o-framework-twitter-bootstrap>>. Acesso em: 09/10/2018.

“Google Maps Javascript API”. Disponível em:

<<https://developers.google.com/maps/documentation/javascript/3.exp/reference>>. Acesso em: 09/10/2018.

