

PROTÓTIPO DE PREVISÃO DE TEMPO PARA AMBIENTE AGRÍCOLA DE BAIXO CUSTO

ROCHA, A. S.*

LUCAS M.†

Resumo

As técnicas de sensoriamento remoto, por meio das quais é possível obter as variáveis ambientais e meteorológicas em tempo real, tornaram-se uma ferramenta indispensável para obter uma agricultura de precisão. O principal objetivo deste trabalho é fazer um estudo sobre as tecnologias disponíveis de sistemas embarcados, desenvolvendo um software IHM de aquisição de dados meteorológicos e ambientais, para ser utilizado em ambiente agrícola. Foi feita uma pesquisa bibliográfica para fundamentar os conteúdos a serem aplicados, como, por exemplo, os dispositivos de sistemas embarcados e sensoriamento remoto, que demonstrou as melhores variáveis a serem utilizadas como referência. Comprovou-se que a tecnologia é um meio poderoso para obter melhorias em cenários imprevisíveis.

Palavras chaves: Software de aquisição, Web Site, Banco de Dados.

Remote sensing techniques, through which real-time environmental and weather variables can be obtained, have become an indispensable tool for precision agriculture. The main objective of this work is to study the available technologies of embedded systems, developing an IHM software for the acquisition of meteorological and environmental data, to be used in an agricultural environment. A bibliographical research was done to base the contents to be applied, such as embedded systems devices and remote sensing, which demonstrated the best variables to be used as reference. It has been proven that technology is a powerful means to get improvements in unpredictable scenarios.

Keywords: Acquisition software, Web Site, Database.

I Introdução

Estudos relatam que o setor rural/agrícola apresenta uma carência de tecnologia nas áreas de sistemas de previsão de tempo local, pois os altos custos de equipamento inviabilizam a sua compra. Para isso, foi desenvolvido um sistema versátil, que possui um custo relativamente baixo quando

comparado à concorrência, apresentando resultados de previsão análogos aos de mercado. Neste setor, existem vários sensores para as mais diversas medições, como umidade do ar, velocidade do vento e nível pluviométrico de chuvas, porém, existe a necessidade de se cruzar estas informações e transformá-las em conhecimento. Devido a esse cenário, foi desenvolvido um software gerenciador online na *Framework* Getz, responsável por receber os dados e transformá-los em gráficos de rápida compreensão.

Outra relevância que estimulou a escolha do tema foi perceber que o sistema desenvolvido pode acompanhar situações climáticas, a partir dos dados gerados por um dispositivo portátil micro controlado que capta sinais climáticos e ambientais utilizando sensores direcionados para esse fim. Os sinais são amostrados e transmitidos, por meio de um subsistema de comunicação sem fios, a um *Web Service*¹ [21], para sua visualização em tempo real.

O sistema deverá receber estas informações e:

- armazená-los em um banco de dados;
- mostrar os dados em uma interface própria;
- mostrar gráficos com base no banco de dados.

Para a validação, foi gerada uma curva de temperatura e umidade da latitude 19° 44' 54"S e longitude 47° 55' 55"W, e confrontado com as curvas do site weather.com.

II Aplicações e Linguagens

Nos dias atuais, muitos meios de comunicação se propagam pela sociedade, entretanto é indiscutível que a internet se evidencia como o principal modo de comunicação. Ela cresce a todo segundo, com novos sites, usuários, bancos de dados e aplicações, dentre outros fatores. A internet é uma rede de computadores que interconecta todos os dispositivos de computação ao redor do mundo. O notório crescimento foi conduzido pela alta quantidade de computadores e pela diversidade de serviços online [9].

A criação de uma boa interface necessita da integração de vários processos, tais como a comunicação entre o *hardware*, que realiza a leitura dos dados e o software gerenciador, usado como interface gráfica ao usuário.

A partir desses requisitos, a aplicação assume papel como ferramenta de interação entre usuário/máquina, facilitando a tomada de decisões.

¹Conjunto de métodos acessados e invocados por outros programas utilizando tecnologias Web

*ROCHA, A. S., Universidade de Uberaba (UNIUBE), Uberaba, Minas Gerais, Brasil, adamisstarling@gmail.com

†LUCAS. M., Orientador, Universidade de Uberaba (UNIUBE), Uberaba, Minas Gerais, Brasil, marcelo.lucas.eng@uniube.br

A Diferenças entre Aplicação Convencional e Web

A dinâmica constante via internet ocasiona a necessidade da criação de web sites e serviços, em que tudo se interconecta e deve estar sempre disponível para, a qualquer momento, ser acessado.

Com essa dinâmica, diferenciam-se dois tipos de aplicação: convencional ou desktop e web. As convencionais ou desktop são uma solução na qual se tem instalado, na máquina do usuário, um cliente, e, por sua vez, consome esses acessos via internet. Porém, essa solução apresenta problemas com relação a tamanho e tipo de sistema operacional usado pelo computador desse usuário. Já a aplicação web não precisa ser instalada e não depende de um sistema operacional e de espaço para ser instalado, bastando apenas que o usuário possua um navegador e uma conexão estável com a internet.

Assim, uma aplicação web se diferencia da convencional pela alta disponibilidade, sem muitos vínculos, e também pelo fato de essa estrutura garantir uma maior organização referente a atualizações, pois toda a aplicação fica concentrada em um único servidor, evitando erros de programação e descontinuação. Devido a esses fatores, a aplicação web ganha notoriedade e foi a escolhida para este projeto.

B Ferramentas para Desenvolvimento

O desenvolvimento de aplicações *web* acontece a partir de ferramentas concisas para a criação do serviço ou produto. Torna-se necessária a utilização de vários padrões, como o MVC (Modelo Visual e Controle), que, de acordo com Ramos, separa uma aplicação em 3 camadas distintas, e subdivide as regras de negócio, de forma a menos impactar, caso existam muitas alterações em uma das camadas definidas [14].

Outra ferramenta necessária é o banco, de dados para abstrair e guardar as informações lidas. Para a manipulação de dados dentro desse banco, será usado o SQL ² que consiste em pesquisar e manipular dados.

Além dessas ferramentas principais, será usada também a *Framework*, Getz que cuidará das funcionalidades e dos padrões, e o Notepad++, para a manipulação dos códigos HTML e PHP, que serão gerados pelo Getz.

B.1 Linguagem de Programação Web – HTML e PHP

Dentre as linguagens de programação mais usuais, o HTML é sugerido nas aplicações web, considerando a manipulação de imagens, gráficos e multilinks. O HTML é uma programação independente, sem instalações adicionais e compatível basicamente com vários padrões, inclusive o CSS ³, que facilita sua estilização e manipulação visual. [15, 4]

²(*Structured Query* Linguagem ou Linguagem de Consulta Estruturada)

³(*Cascading Style Sheets* ou folhas de estilos em cascata)

Sabendo da diferença entre as aplicações convencionais e a web, esta última necessita de um perfeito sincronismo para enviar e receber dados. Uma tecnologia aplicada web que pode contribuir bastante para essa dinâmica é o PHP. Essa linguagem de programação permite a comunicação entre Cliente/Servidor e também a inserção e/ou geração de códigos em páginas de HTML. Nessas condições, essa tecnologia vai diretamente ao encontro do HTML, facilitando as aplicações web [16].

B.2 Padrão Modelo Visual e Controle (MVC)

O MVC ("Modelo", "Visual" e "Controle") nada mais é do que um padrão de arquitetura que se divide em 3 camadas distintas:

- Visual: na qual acontece a interação direta com o usuário;
- Modelo: que manipula diretamente os dados do banco de forma direta ;
- Controle: tem o papel de entremeio das camadas "Visual" e "Modelo", e também aplica as regras de negócio.

B.3 Banco de Dados - SQL

Dentre as ferramentas para aplicação web, o banco de dados é imprescindível, pois, é onde as informações serão guardadas e gerenciadas . O SQL é uma linguagem para se manipular banco de dados relacionais, e muitos sistemas SGBD's ⁴ o utiliza para manipular dados, usando uma interface. [17].

Sendo assim, esta linguagem é utilizada de forma a inserir, remover e atualizar dados do banco, de forma direta, simplesmente usando um SGBD e a linguagem SQL para realizar tais tarefas.

B.4 Getz Framework

Com o alto índice de dinâmica, as aplicações web devem oferecer ferramentas de alto desempenho. A *Framework* Getz é construída utilizando o padrão MVC, escrito na linguagem PHP, que ajuda, de forma rápida e organizada, no desenvolvimento de projetos web, gerando todo o código de forma automática, havendo apenas a necessidade de manipular regras e o visual [3].

B.5 Notepad++

O Notepad++ é um excelente editor de códigos PHP, CSS, Java Script, dentre outras linguagens de programação. É muito leve e sua construção é focada em gastar o mínimo possível de CPU, e, assim, reduzir a emissão de dióxido de carbono por consumo de energia elétrica [18].

O Notepad++ é regido pela licença GPL Free, portanto um programa *Open Source*.

⁴Sistema de Gerenciamento de Banco de Dados Relacionais

B.6 Variáveis coletadas

O projeto consiste na aquisição de dados via web para estações meteorológicas. No entanto, para se coletar dados reais, haveria a necessidade de uma estação meteorológica profissional que realizasse a coleta, porém tais dados serão gerados baseando-se nos dados do site weather.com, que são colhidos via estação profissional, conforme a figura 1.

id	data_hora	umidade	temperatura	cadastrado	modificado
1	2017-11-30 09:00:00	80	22	2017-11-30 08:35	2017-11-30 08:40
2	2017-11-30 09:15:00	78	22	2017-11-30 09:04	2017-11-30 09:04
3	2017-11-30 09:30:00	76	23	2017-11-30 09:04	2017-11-30 09:04
4	2017-11-30 09:45:00	75	23	2017-11-30 09:04	2017-11-30 09:04
5	2017-11-30 10:00:00	73	23	2017-11-30 09:04	2017-11-30 09:04
6	2017-11-30 10:15:00	72	24	2017-11-30 09:05	2017-11-30 09:05
7	2017-11-30 10:30:00	71	24	2017-11-30 09:05	2017-11-30 09:05
8	2017-11-30 10:45:00	70	24	2017-11-30 09:05	2017-11-30 09:05
9	2017-11-30 11:00:00	69	24	2017-11-30 09:05	2017-11-30 09:05
10	2017-11-30 11:15:00	69	25	2017-11-30 08:55	2017-11-30 08:55
11	2017-11-30 11:30:00	69	25	2017-11-30 09:14	2017-11-30 09:14
12	2017-11-30 11:45:00	68	25	2017-11-30 09:14	2017-11-30 09:14
13	2017-11-30 12:00:00	67	25	2017-11-30 09:14	2017-11-30 09:14
14	2017-11-30 12:15:00	66	25	2017-11-30 09:14	2017-11-30 09:14
15	2017-11-30 12:30:00	65	26	2017-11-30 09:14	2017-11-30 09:14
16	2017-11-30 13:00:00	64	26	2017-11-30 08:50	2017-11-30 08:50
17	2017-11-30 13:15:00	63	26	2017-11-30 08:54	2017-11-30 08:54
18	2017-11-30 13:30:00	63	26	2017-11-30 08:54	2017-11-30 08:54
19	2017-11-30 13:45:00	63	26	2017-11-30 08:54	2017-11-30 08:54
20	2017-11-30 14:00:00	62	26	2017-11-30 08:55	2017-11-30 08:55

Figura 1: Dados modificados para amostragem

Para realização dos testes, duas variáveis ambientais serão analisadas:

- Temperatura - Medição da temperatura em graus Celsius (Celsius);
- Umidade do Ar - Medição da umidade relativa do ar - de modo indireto - em porcentagem (%);

As variáveis foram escolhidas por serem as mais importantes de uma estação meteorológica e porque muitas análises e previsões podem ser feitas pela comparação destas medidas, ao longo de determinados períodos de tempo. A indicação do vento não será feita porque sua implementação apresentou alto custo.

III Metodologia

A Montagem da Interface Gráfica para Usuário

Toda a interface será gerada a partir da *Framework Getz*, porém, existe a necessidade de se usar um banco de dados já montado e bem estruturado, de preferência que remeta, o mais próximo possível, da regra de negócio, uma vez que a *Framework* fará toda a engenharia reversa do bando de dados.

A.1 Banco de Dados

Para aplicação web, foi pensado um banco de dados que fosse simples e de uso comum, uma vez que tal sistema deveria ser hospedado na web. Por motivo, foi escolhido o banco de dados MySQL, que é relacional e sua estrutura é amigável.

Para a utilização da *Framework Getz* e necessária a importação do SQL básico(`db_min_pt-BR.sql`) que ele disponibiliza dentro da pasta `_doc` para o MySQL.

Outro ponto importante foi a criação da tabela chamada `medicoes`, em que foi definida a coluna `id`, com auto incremento, e a coluna `data_hora`, para gravar a hora da leitura. Também foram criadas as colunas `umidade` e `temperatura`, para, respectivamente, gravar estas variáveis. E ainda existe a necessidade de se criar duas colunas para atender aos requisitos da *Framework Getz*, que são: `cadastrado` e `modificado` para gravar as datas respectivas. A figura 2 mostra a tabela medições criada no banco de dados MySQL.

medicoes					
id:	int				
data_hora:	datetime				
umidade:	varchar				
temperatura:	varchar				
cadastrado:	datetime				
modificado:	datetime				

Figura 2: Tabela de medição

A.2 Configuração Inicial do Servidor PHP

É importante salientar que a *Framework Getz* necessita de algumas configurações adicionais para com o PHP, uma vez que utiliza a tecnologia de "URL Amigável", que facilita a compreensão e memorização do usuário. Então se deve ativar esta opção do apache. Para isso, vá até o arquivo `.http.conf` ou `php.ini` do servidor php e retire o '#' da frente da frase '#LoadModule rewrite_module modules/mod_rewrite.so' e/ou '#mod_rewrite', e também da frase '#LoadModule headers_module modules/mod_headers.so' e/ou '#mod_headers';

A.3 Elaboração do Web Site

A elaboração do web site inicia-se a partir da *Framework Getz*, no qual se deve, inicialmente, realizar algumas configurações para que ela faça a engenharia reversa do banco e gere todo o contexto.

A configuração inicial se dá pela alteração do arquivo `settings.xml`, que se encontra na pasta `_bin`, conforme a Figura 1. Deve-se alterar `@PROJECT` para o nome do projeto, `http://www.@DOMAIN`, para o domínio a ser hospedado, e existe a possibilidade de se configurar um servidor local e um oficial, alterando os valores, `@SERVER`, `@USERNAME`, `@PASSWORD`, `@DATABASE`.

Após alteradas estas configurações, existe a necessidade de se abrir um browser(navegador de internet) e realizar a chamada descrita no site 'http://localhost/project/_bin/boot.php', lembrando-se de que esta URL deve ser executada localmente, uma vez que o teste é realizado em ambiente de teste local.

Ao executar a URL no navegador, será gerado um arquivo com o nome `reveng.exe`, na pasta `_bin`, que deverá

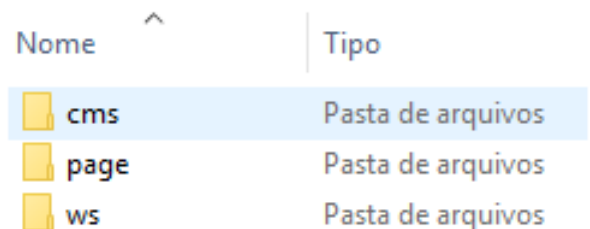
Algoritmo 1: Arquivo de configuração do Getz *Framework*

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Settings -->
3 <settings>
4 <!-- Project name -->
5 <project>_PROJECT</project>
6 <domain>http://www._DOMAIN</domain>
7 <!-- Local connection -->
8 <localhost>
9 <server>_SERVER</server>
10 <username>_USERNAME</username>
11 <password>_PASSWORD</password>
12 <database>_DATABASE</database>
13 </localhost>
14 <!-- Official connection -->
15 <official>
16 <server>_SERVER</server>
17 <username>_USERNAME</username>
18 <password>_PASSWORD</password>
19 <database>_DATABASE</database>
20 </official>
21 </settings>
```

ser executado para que toda a engenharia reversa seja feita e todos os arquivos sejam gerados.

Com toda a engenharia reversa gerada, dentro da pasta do Getz é importante salientar que o Getz gera 3 módulos bem distintos na pasta **Mod**, conforme a figura 3, sendo:

- cms: Módulo referente ao sistema de segurança que o próprio Getz tem, acessando a URL: 'http://localhost/project/cms/'
- page: Módulo referente ao web site (inicialmente está vazio com somente um **index.php**, porém, dá liberdade para trabalhar da forma que desejar.)
- ws: Módulo referente a Web Service para requisições via 'GET' e 'POST'



Nome	Tipo
cms	Pasta de arquivos
page	Pasta de arquivos
ws	Pasta de arquivos

Figura 3: Módulos gerados pelo Getz *Framework*

A.4 Webservice

O Getz, ao ser executado, cria o módulo "WS", que nada mais é que um web service que aceita requisições por URL

amigável[25], e também, o envio de dados, diretamente, para a base dados, não sendo necessária praticamente nenhuma alteração adicional para se utilizá-lo.

A.5 Gráficos

Todos os gráficos foram criados com a biblioteca da própria *Framework* Getz, que permite montar gráficos com dados em uma lista, que podem e/ou não estar em um banco de dados.

Para renderizar na página os gráficos, optou-se por realizar a média da hora lida para se montar apenas um ponto, uma vez que é possível que se tenha mais de uma leitura por hora e todas as variáveis envolvidas não têm alterações bruscas.

Para a geração dos gráficos foi utilizada a função de gráficos (graphic) do Getz, que baseia-se no elemento Canvas do HTML5⁵. Para iniciar, foi necessária a coleta dos dados, que foram relacionados às datas, com os valores da variável em questão, e passados para a função, com os seguintes parâmetros, de acordo com a tabela 1.

Tabela 1: Tabela de parâmetros da Função "grafic".

Parâmetro	Tipo	Definição
id	String	Identificação do elemento HTML.
lines	Array	Lista de Valores em questão.
columns	Array	Lista de datas.
preValue	String	Mascara anterior aos Valores.
posValue	String	Mascara posterior aos Valores.
convertToDecimal	Boolean	Se caso os valores forem 'Double' setar true para manipular a mascara.
lineColor	String	Cor da linha a ser mostrado.

E, com isso, foi gerado um gráfico, de acordo com cada variável definida, como o exemplo da figura 4.

⁵(HyperText Markup Language - Linguagem de Marcação de Hipertexto) [23]

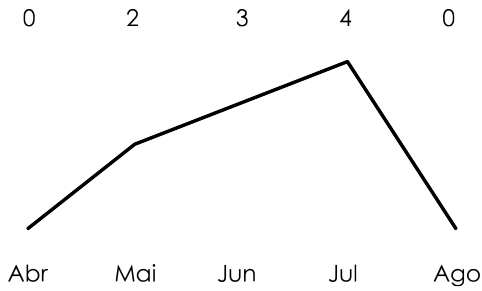


Figura 4: Exemplo de Gráfico gerado pela função "grafic", da *Framework Getz*

IV Resultados

Ao coletar todos os dados via Web Service, os gráficos mostraram satisfatoriamente as curvas ao logo do tempo. Todos os gráficos são gerados com sua variável colhida na parte superior e a hora que ela foi gerada, na parte inferior, reforçando que, caso exista mais de uma leitura por hora, foi feita a média, para que se tenha uma maior precisão na variável colhida. Sendo assim, a temperatura não se alterou tão abruptamente entre uma hora e outra, conforme a figura 5.

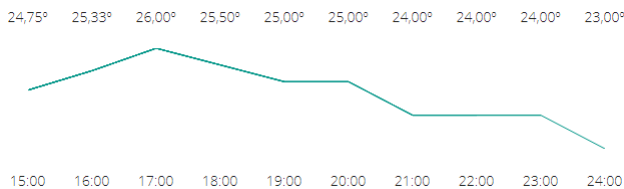


Figura 5: Gráfico de temperatura de acordo com o tempo, gerado a partir da *Framework Getz* e dos dados do site Weather.com

A figura 6 mostrou que, ao se aproximar de 12, horas a umidade diminuiu e, com o passar do tempo, aumentou novamente.

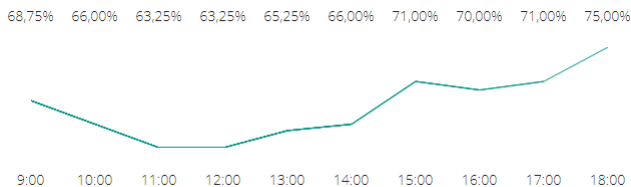


Figura 6: Gráfico de Umidade de acordo com o tempo, gerado a partir da *Framework Getz* e dos dados do site Weather.com

V Discussão

Os Gráficos gerados são, de fácil entendimento[24], porém, sem uma aplicação específica, sendo possível outras aplicações consumirem tais informações. Cruzando dois ou mais gráficos, um aspecto que é importante salientar é que todos os dados foram retirados do site weather.com, mas ao se colher dados de um dispositivo físico, os gráficos ficaram vulneráveis a falhas, uma vez que dependerá apenas da precisão e qualidade dos sensores que irão colher esses dados.

Outro aspecto importante é a linguagem PHP que, além de ser muito versátil com relação à sua forma de programar, tem uma das principais vantagens: o seu baixo custo computacional e sua fácil alocação. A maioria dos servidores encontrados na web consegue executá-la de forma satisfatória, tanto nas versões mais antigas quanto na mais atual, sendo prática e rápida a implementação de um web service para colher estes dados.

VI Conclusão

A agricultura, com o passar do tempo, está cada vez mais fadigada, dependendo muito do seu espaço físico. Com o aumento da população, essa necessidade é cada vez mais forçada a ter um maior rendimento em uma menor área e a agricultura de precisão se torna cada vez mais indispensável nos campos. E este sistema vem para contribuir com este meio, haja vista que o sistema recebeu dados e mostrou gráficos que são mais próximos à realidade, facilitando a tomada de decisões. E, por realizar a média das variáveis colhidas, aumenta a precisão lida pelos sensores e diminui ainda mais a possibilidade de erro.

Referências

- [1] SOUZA, Z. M. et al. A. Análise dos atributos do solo e da produtividade da cultura de cana-de-açúcar com o uso da geoestatística e árvore de decisão. *Cienc. Rural*, v. 40, n. 4, 2010.
- [2] SCHONEMANN, E.; MATTHIAS, B. (2011). Springer, T. A new Approach for GNSS Analysis in a Multi-GNSS and Multi-Signal Environment. *Journal of Geodetic Science*. v. 1, n. 3, p.204-214.
- [3] SAKAMOTO, Mario. Gets. Website, 2017. Disponível em: <<http://wtag.com.br/getz/contents/4/>> Acessado em: 3 jun. 2017
- [4] MDN. Css. Disponível em: <<https://developer.mozilla.org/pt-br/docs/web/css>>. Acesso em: 13 nov. 2017.
- [5] OLIVEIRA, R. R. Uso do Microcontrolador ESP8266 para Automação Residencial. 42f. Tese de Doutorado - Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2017.

- [6] FOROUZAN, Behrouz A. Comunicação de Dados e Redes de Computadores. 3.ed. Porto Alegre: Bookman, 2006. 840p.
- [7] BORGES, L. P.; DORES, R. C. Automação predial sem fio utilizando bacnet/zigbee com foco em economia de energia. 76f. Trabalho de conclusão de curso - Curso de Graduação em Engenharia de Controle e Automação – UNB, Brasília, 2010.
- [8] INAMASU, Ricardo Y.; BERNARDI, Alberto C. de Campos. Agricultura de precisão: Resultados de um novo olhar. Brasília: Embrapa, 2014. p. 21-33.
- [9] KUROSE, James F. Redes de computadores e a internet uma abordagem top-down. 6.ed. São Paulo: Pearson Education do Brasil, 2013. 656p.
- [10] CONTE, T; TRAVASSOS, G.H; MENDES E. Revisão Sistemática sobre Processos de Desenvolvimento para Aplicações Web. Relatório Técnico ESE/PESC - COPPE/UFRJ. 2005.
- [11] CONALLEN, Jim. Modeling Web Application Architectures with UML. Communications of ACM, v. 42, n. 10, Oct. 1999.
- [12] BOAR, B.H. Application Prototyping: A Requirements Strategy for the 80's. WileyInterscience, New York, 1984.
- [13] SCHWINGER, Wieland; KOCH, Nora. Modeling web applications. Web Engineering, p. 39-64, 2006.
- [14] RAMOS, Allan. MVC - Afinal, é o quê? Web Site, 2015. Disponível em: <<https://tableless.com.br/mvc-afinal-e-o-que/>>. Acessado em: 3 jun. 2017.
- [15] SCHMITZ, Daniel P. Dominando Adobe Flex 4. Bauru: Editora Viena, 2010. 231p.
- [16] PARK, J. PHP a Bíblia. 2.ed. Rio de Janeiro: Campus, 2002. 697p.
- [17] FURTADO, Gustavo. Você precisa saber o que é SQL! Web Site, 2013. Disponível em: <http://www.dicasdeprogramacao.com.br/o-que-e-sql/>. Acessado em: 03 jun. 2017.
- [18] PINTO, Pedro. Quais os 5 melhores editores de texto em 2016? Website, 2017. Disponível em: <https://pplware.sapo.pt/software/quais-os-5-melhores-editores-texto-2016/>. Acessado em: 3 jun. 2017.
- [19] TECMUNDO. O que é tcp/ip?. Disponível em: <<https://www.tecmundo.com.br/o-que-e/780-o-que-e-tcp-ip-.htm>>. Acesso em: 13 nov. 2017.
- [20] PAIVA, C. K. A. Palestrantes Especiais, 2010. Disponível em: <<http://palestrasespeciais.blogspot.com.br/2010/12/meteorologia>>. Acesso em: 9 maio, 2017.
- [21] OPENSOFTE. Web service: o que é, como funciona, para que serve?. Disponível em: <<https://www.opensoft.pt/web-service/>>. Acesso em: 14 nov. 2017.
- [22] JSON. Introdução ao json. Disponível em: <<https://www.json.org/json-pt.html>>. Acesso em: 14 nov. 2017.
- [23] MDN WEB DOCS. Html5. Disponível em: <<https://developer.mozilla.org/pt-br/docs/web/html/html5>>. Acesso em: 14 nov. 2017.
- [24] BRASIL ESCOLA. Tipos de gráficos. Disponível em: <<http://brasilecola.uol.com.br/matematica/graficos.htm>>. Acesso em: 30 nov. 2017.
- [25] THIAGO BELEM. Aprendendo urls amigáveis (friendly urls). Disponível em: <<http://blog.thiagobelem.net/aprendendo-urls-amigaveis>>. Acesso em: 30 nov. 2017.