ISSN: 1981-1896
REVISTA
AUGUSTUS

GESTÃO HÍDRICA USANDO A BIBLIOTECA GOOGLE CHARTS

Sávio Silva Rodrigues
Acadêmico em Agronomia pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
savio.rm123@hotmail.com – cs3desenvolvimento@gmail.com

Leonardo Nascimento Lima
Acadêmico em Agronomia pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
leo-nascimento.l@hotmail.com

Pedro Augusto Alves Amaral
Acadêmico em Engenharia Agrícola e Ambiental pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
pedroaugustoalves@hotmail.com

Josué Santos Duarte
Acadêmico em Engenharia Agrícola e Ambiental pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
josueduart16@gmail.com

Giovana Cristina Gomes Barbosa
Acadêmico em Engenharia Agrícola e Ambiental pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
ggiovanacristina@gmail.com

Jeniel Borges dos Santos
Acadêmico em Engenharia Agrícola e Ambiental pelo Instituto de Ciências Agrárias da UFMG, Montes Claros, MG, Brasil
jenielbotges@gmail.com

Edson de Oliveira Vieira
Doutor em Engenharia Agrícola pela Universidade Federal de Viçosa (UFV), Montes Claros, MG, Brasil
eovieira@ufmg.br

RESUMO

O gerenciamento hídrico compreende não só a captação de dados, mas também o tratamento, armazenamento e posterior apresentação ao usuário. Deste modo, o presente trabalho buscou desenvolver uma aplicação web usando a biblioteca Google Charts para que os dados sejam apresentados de forma gráfica. O trabalho desenvolvido no Laboratório de Hidráulica do Instituto de Ciências Agrárias da Universidade Federal de Minas Gerais, valeu-se da linguagem de programação JavaScript e da biblioteca jQuery para recuperar dados de

um reservatório, armazenados em um servidor web e plotá-los em uma página HTML. Deste modo, criou-se uma aplicação que apresenta ao usuário o estado de um reservatório em tempo real, pois, o gráfico de nível do mesmo é atualizado a cada minuto. Concluiu-se então que o uso da biblioteca Google Charts no gerenciamento hídrico é viável, apresentando a gestores de recursos hídricos uma nova ferramenta de apoio a tomada de decisões.

Palavras-chave: Controle-Reservatórios. Instrumentos de gestão hídrica. Sistema de informação de Recursos Hídricos. Agroinformática.

WATER MANAGEMENT USING THE LIBRARY GOOGLE CHARTS

ABSTRACT

Water management comprises not only the capture of data, but also the treatment, storage and subsequent presentation to the user. Thus, the present work sought to develop a Web application using the Google Charts library so that the data will be presented graphically. The work, developed at the Hydraulics Laboratory of the Agricultural Science Institute of the Federal University of Minas Gerais, used the JavaScript programming language and the jQuery library to retrieve data from a reservoir, stored on a Web server, and plot them in an HTML page. In this way, an application is created that presents the user with the state of a reservoir in real time, Therefore, the same level chart is updated every minute. It was then concluded that the use of the Google Charts Library in water management is feasible, introducing water resource managers a new tool to support decision-making.

Keywords: Reservoir-Control. Tools-Water Resources. Water Resources information system. Agro informatics.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

1 INTRODUÇÃO

O Sistema Nacional de Informações sobre Recursos Hídricos (SNIRH) é um dos instrumentos de gestão previsto na Política Nacional de Recursos Hídricos, instituída pela lei nº 9.433, de 08 de Janeiro de 1997, conhecida como Lei das Águas. Porém, no gerenciamento hídrico não basta coletar informações, etapas de tratamento, armazenamento e recuperação dos dados também devem ser realizadas. Não obstante, os usuários e gestores devem ter acesso a esses dados em tempo real. Todavia, por muitas vezes tais dados se apresentam na forma de extensas planilhas, o que, para muitos não têm significado imediato.

Deste modo, faz-se necessário que os dados, após coletados e processados, sejam mostrados de forma gráfica, pois, além de facilitar o entendimento, possibilita um melhor tempo de resposta perante a informação apresentada. Isso agiliza o processo de tomada de decisão por parte do usuário.

Nesse âmbito, a Google apresenta a biblioteca de desenvolvimento web “Google Charts”, que traz ferramentas para a apresentação das informações na forma de gráficos interativos atualizados em tempo real e compatíveis com a maioria dos *browsers*[1]. Outro ponto importante é que o uso da mesma é grátis, logo, um atrativo a mais à utilização dessa, principalmente em ambiente de desenvolvimento acadêmico. Também é importante ressaltar que, mesmo sendo uma biblioteca gratuita, é robusta e entrega um resultado de qualidade.

Sendo assim, o objetivo do presente trabalho foi criar, a partir de dados armazenados em servidor virtual, uma aplicação Web que mostre, em tempo real, a dinâmica do nível de um reservatório monitorado por uma estação em campo, usando para tal a biblioteca Google Charts, disponibilizando então, uma nova ferramenta de auxílio à gestão de reservatórios hídricos, o que trará ainda, maior assertividade na tomada de decisões.

[1] *Software* de acesso às páginas HTML.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

2 METODOLOGIA

O trabalho foi desenvolvido no Laboratório de Hidráulica do Instituto de Ciências Agrárias da Universidade Federal de Minas Gerais. Utilizou-se como ambiente de desenvolvimento o software Visual Studio Code (Versão 1.30.2) em que todos os *scripts* e páginas foram criadas e testadas no *browser* Google Chrome (Versão 76.0.3809.100 64 bits). A linguagem de marcação utilizada foi o HTML5 eas páginas foram estilizadas a partir de CSS puro, além do uso da biblioteca Bootstrap (Versão 4.3.1).

Inicialmente, se faz necessário que o *script* Js (JavaScript), responsável pela automação dos processos na página, receba um elemento HTML do tipo , pois, é na mesma que a biblioteca Google Charts plotará o gráfico. Após essa etapa, a aplicação cliente faz requisições periódicas ao servidor através da biblioteca, da linguagem JavaScript, jQuery (Versão 3.3.1) utilizando da função $.ajax().

O servidor retorna os dados no formato JSON e são passados à função "google.visualization.DataTable()". A resposta do servidor traz uma espécie de tabela com colunas de nível, volume, data e hora. Junto a essa tabela tem-se uma coluna de funções, isto é, junto aos dados supracitados tem-se metadados como rótulos de pontos ou cor de linha em pontos específicos do gráfico.

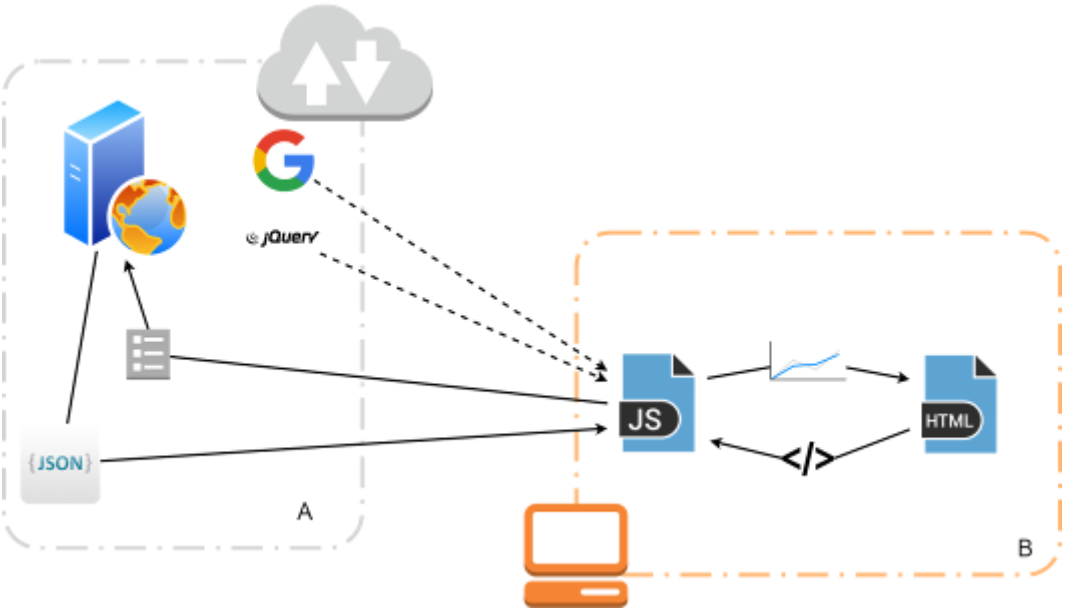
Os gráficos são personalizados, são inseridas informações de título, legenda, cor, etc. Em seguida são plotados a partir da função "class.draw(data,options)" em que:

- "class" é a instância da classe correspondente aos gráficos;
- "data" é a variável correspondente aos dados retornados pelo servidor e;
- "options" é a variável que traz as informações relacionadas ao conteúdo, como por exemplo, o título e legenda do gráfico.

A Figura 1 demonstra como é feita a comunicação entre as partes do sistema. É importante salientar que, as requisições da página HTML só se iniciam após a mesma ser carregada, pois, desse modo, não se compromete a velocidade na qual esta é apresentada ao usuário. Note que existe uma distinção das atividades realizadas em nível de servidor (Figura 1A) e em nível cliente (Figura 1B).

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

Figura 2: Esquema detalhando a comunicação cliente x servidor



Posto que os dados são atualizados em tempo real no servidor, a página utiliza-se de mecanismos para que os gráficos se mantenham atualizados. Todo o processo supracitado está englobado dentro de uma função que, a cada 1 minuto é chamada pela função setInterval().

Implementado o sistema, foram feitos testes unitários para validar individualmente todas as funções da aplicação. Após essa etapa foram feitos testes de integração da aplicação local com o servidor. Ao fim dos testes e feitas as devidas considerações consoante aos mesmos, o sistema foi hospedado virtualmente. Para tanto, utilizou-se um pacote gratuito da empresa AwardSpace.

3 RESULTADOS E DISCUSSÃO

Para compreender a implementação deste sistema deve-se entender o código fonte do mesmo. As figuras 3 a 8 demonstram como foi feita a implementação do esquema proposto na

Figura **2** culminando na página apresentada nas figuras 9 e 10.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

A Figura 2 apresenta o *script* de marcação da página em HTML5 em que ,a identificada por *id="drawChart"* será utilizada mais tarde para plotagem do gráfico de nível.

Figura 3: HTML5 – elemento de suporte ao gráfico.

A Figura 3 por sua vez, apresenta o *script* Js responsável pelas requisições das bibliotecas jQuery e Google Charts.

Figura 4: Js – requisição de bibliotecas.

Feitas as requisições de bibliotecas, em conformidade com o esquema proposto na Figura 1, a página faz a requisição do tipo *ajax* ao servidor web como mostra a Figura 4. É importante observar que o endereço *"main.php"* é a página no servidor responsável pela comunicação com o cliente, ou seja, é esta quem recebe as requisições, agrupa os dados e retorna ao requisitante. A Figura 5 exemplifica essa resposta feita em formato JSON, que foi escolhido por se tratar de um formato leve de comunicação.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

Figura 5: Js – jQuery – requisição Ajax.

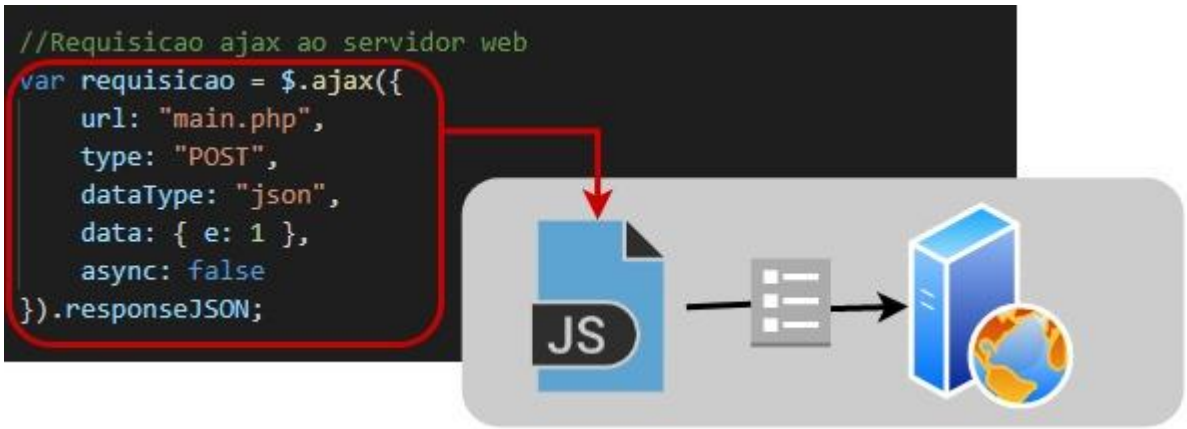
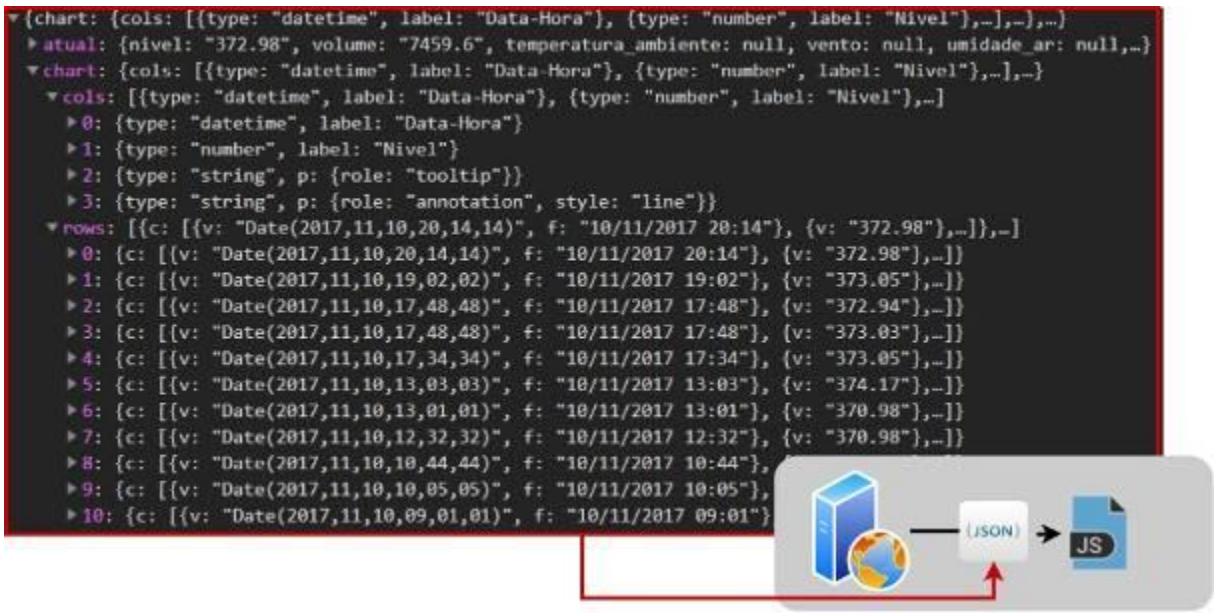


Figura 6: JSON – resposta.

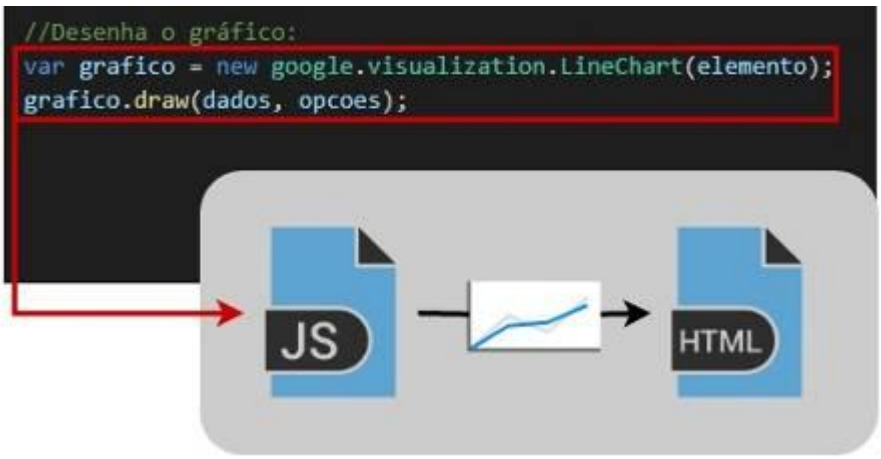


Na Figura 6 a variável “dados” recebe um objeto com a tabela criada pela biblioteca Google Charts a partir da resposta (Figura 5) do servidor. A partir dessa tabela, desenha-se o gráfico (Figura 7) no elemento (Figura 2) passado ao *script* inicialmente.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

Figura 7: Js - Google Charts recebe os dados.

Figura 8: Js - Google Charts desenha o gráfico.



Conforme citado, as figuras anteriores demonstraram a implementação da página bem como os mecanismos utilizados para o seu funcionamento, todavia, o interesse do leitor se volta às figuras 9 e 10, pois, essas representam o resultado final deste trabalho.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

Figura 9: HTML: página principal em dispositivo móbile.

Figura 10: HTML: página principal em desktop.

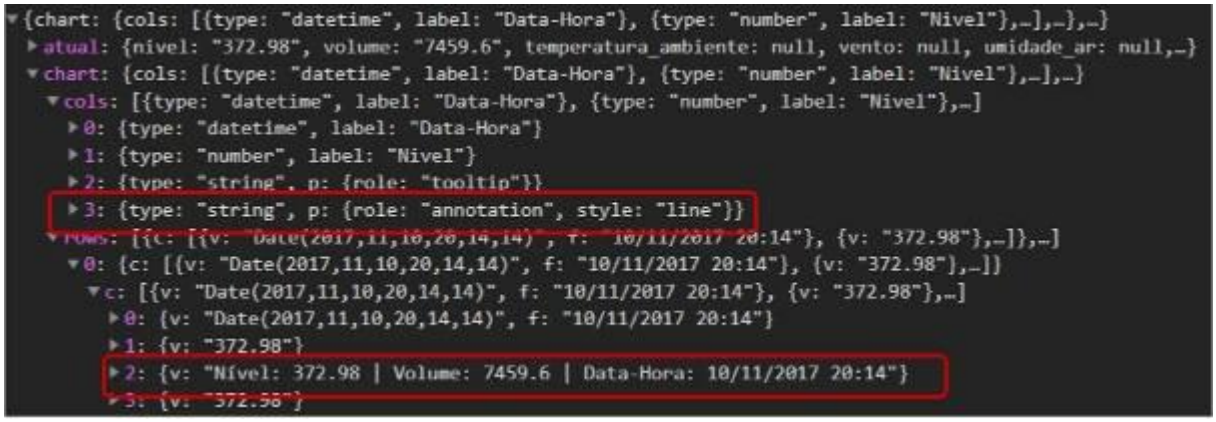
Vale ressaltar que a biblioteca utilizada traz a possibilidade de se aplicar rótulos aos pontos do gráfico, que neste caso mostra informações associadas ao ponto de destaque[2]: nível, volume, data e horário. Esse recurso pode ser utilizado para diversos fins, como por exemplo, destacar visualmente pontos extremos do reservatório.

[2] A caixa de texto com as informações são apresentadas ao usuário ao posicionar o cursor sobre o ponto.

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

As informações associadas aos pontos são definidas no servidor e passados ao cliente junto à resposta JSON. Note que na Figura 10 tem-se um exemplo de resposta, semelhante ao apresentado na Figura 5, no entanto, nessa é interessante observar o texto destacado em vermelho que corresponde ao rótulo personalizado.

Figura 11: JSON - Google Charts: colunas de função.



Também é importante observar uma nítida diferença entre o tamanho das imagens apresentadas, pois, estas correspondem a renderização da página aqui descrita em dispositivos de tamanhos distintos. A responsividade é um conceito que na Psicologia refere-se a uma atitude compreensiva. Este conceito é adotado no desenvolvimento web no sentido de otimizar as páginas para os diferentes tamanhos de telas, a saber, uma mesma página se adequa tanto a um dispositivo móvel (Figura 8), quanto em um computador (Figura 9) ou ainda em uma *smart* TV.

Nesse sistema a responsividade é importante, porque nem sempre o usuário tem acesso a um computador, desse modo, pode acessar o site do seu dispositivo móvel sem nenhuma dificuldade, o que não acontece em páginas não responsivas. Por outro lado, se o mesmo deseja fazer uma apresentação por exemplo, a página também se adequa a telas maiores como uma *smart* TV. Nesse sistema, utilizou-se o *framework* Bootrstrap para garantir a responsividade.

O estado do reservatório monitorado é dinâmico, portanto, o sistema também deve ser.
Sendo assim, a página é constantemente atualizada, desse modo, tem-se a garantia de que o

Sávio Silva Rodrigues
Leonardo Nascimento Lima
Pedro Augusto Alves Amaral
Josué Santos Duarte
Giovana Cristina Gomes Barbosa
Jeniel Borges dos Santos
Edson de Oliveira Vieira

estado do reservatório apresentado ao usuário seja o que mais se aproxime da realidade. A Figura 11 demonstra toda a comunicação feita pela página cliente com o servidor (Figura 1), desde as requisições de bibliotecas (Figura 3) até as requisições à página *"main.php"* (Figura 4) que como citado anteriormente, corresponde à página de comunicação do servidor. A Figura 11 evidencia que no período de tempo apresentado, o gráfico foi atualizado diversas vezes após a plotagem inicial.

Figura 12: Chrome DevTools: atividade de rede da página HTML

Name	Status	Type	Initiator
loader.js	200	script	(index)
jquery.js	200	script	(index)
bootstrap-responsive.min.css	200	stylesheet	(index)
main.css	200	stylesheet	(index)
jsapi.js	200	script	(index)
bootstrap.js	200	script	(index)
loader.js	200	script	loader.js:249
engines.json	200	xhr	mcafee_wa_co...
tooltip.css	200	stylesheet	loader.js:238
util.css	200	stylesheet	loader.js:238
jsapi_compiled_format_module.js	200	script	loader.js:225
jsapi_compiled_default_module.js	200	script	loader.js:225
jsapi_compiled_ui_module.js	200	script	loader.js:225
jsapi_compiled_corechart_module.js	200	script	loader.js:225
main.php	200	xhr	jquery.js:2
main.php	200	xhr	jquery.js:2
main.php	200	xhr	jquery.js:2
main.php	200	xhr	jquery.js:2

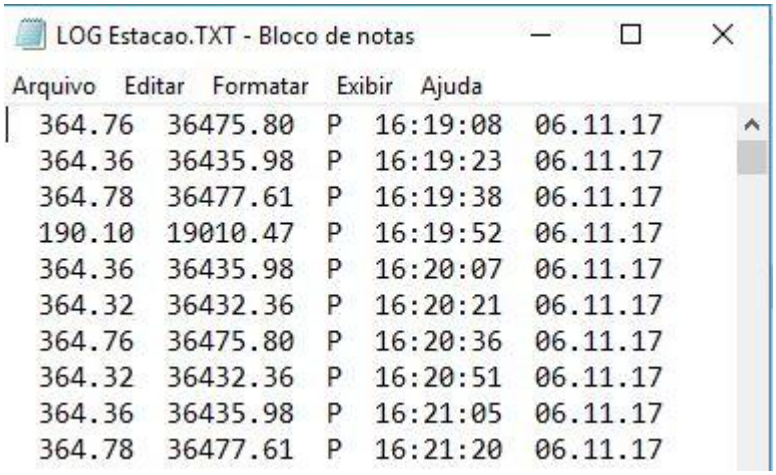
Outrora, o acesso aos dados do reservatório era feito a partir do arquivo de *log* da estação em campo (Figura 12), entretanto, algumas dificuldades ficaram evidentes:

- A estação na maioria das vezes fica geograficamente distante do gestor e,
- O gestor precisa:
  - acessar fisicamente o cartão de memória da estação,
  - fazer *backup* para um computador e,
  - plotar o gráfico em um editor de planilhas (exemplo: Microsoft Excel).

Dada a complexidade desse processo, bem como o fato desse não antecipar casos extremos, fica nítido que o objetivo implícito desse trabalho é simplificar a experiência do

usuário. Tal objetivo vai de encontro com a filosofia da linguagem de programação Python[3] que ressalta a necessidade de sistemas de informação simplificarem a experiência e o acesso do usuário (PETERS, 2004).

Figura 13: Arquivo de log da estação que monitora o reservatório.



Com o objetivo de disponibilizar essa aplicação globalmente, a mesma foi hospedada no servidor web AwardSpace sob o domínio gratuito <http://gerhisa.scienceontheweb.net/>.

CONCLUSÕES

Os dados levantados em campo são simultaneamente transformados em informações e apresentados em tempo real aos gestores. Estes por sua vez têm acesso à página independente da distância do reservatório pois a página está hospedada em um servidor web.

Esse sistema se mostra como uma ferramenta ímpar, pois se adequa às diversas realidades, sendo que algumas dessas cabem trabalhos futuros para sua otimização.

[3] Apesar de não ter sido utilizada nesse sistema como linguagem de programação, a filosofia pregada pela Python é um fator predominante nesse trbalho.

Alguns aspectos da gestão hídrica podem ser antecipados, como a variação brusca no nível do reservatório, ou ainda os valores críticos (volume ecológico e nível de vertente), pois diante dos mesmos, medidas precisam ser tomadas no menor tempo possível.

Dito isso, nota-se que esse trabalho se adequa tanto a realidades restritas — como tanques industriais — quanto cenários mais amplos — como a disponibilidade hídrica da barragem que fornece água potável para uma grande cidade. Logo, a partir da aplicação aqui descrita, tem-se então uma nova ferramenta de gestão de recursos hídricos, que servirá de suporte à tomada de decisões e ajudará antecipar-se a eventos extremos.

REFERÊNCIAS

AGÊNCIA NACIONAL DE ÁGUAS. **Sistema Nacional de Informações Sobre Recursos Hídricos.** Disponível em: <http://www.snirh.gov.br/>. Acesso em: 29 agosto de 2019.

BOOTRSTRAP. **Bootrstrap: documentation.** Disponível em: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>. Acesso em: 15 novembro de 2019.

BRASIL. Lei Nº 9.433, de 8 de janeiro de 1997. **Lex:** Política nacional de recursos hídricos. Disponível em: <http://www.planalto.gov.br/ccivil_03/LEIS/L9433.htm>. Acesso em: 29 agosto de 2019.

GOOGLE DEVELOPERS. **Google Charts: Interactive charts for browsers and mobile devices.** Disponível em: <https://developers.google.com/chart/>. Acesso em: 26 agosto de 2019.

PETERS, T. **PEP 20 - The Zen of Python.** 22 agosto 2004. Disponível em: <https://www.python.org/dev/peps/pep-0020/>. Acesso em: 29 agosto de 2019.

RODRIGUES, S. S.; LIMA, L. N.; VIEIRA, E. O. **GERHISA – Gerenciador de Recursos Hídricos.** Disponível em: <http://gerhisa.scienceontheweb.net/>. Acesso em: 14 novembro de 2019.

_____
Recebido em 25/11/2019.
Aceito em 11/12/2019.