

SISTEMA CLOUD COMPUTING PARA AUTOMAÇÃO DE PROCEDIMENTOS LABORATORIAIS*

Artur Augusto Martins¹

Resumo

Este trabalho tem como objetivo apresentar o projeto de uma plataforma de cloud computing onde profissionais de diversos segmentos da metrologia, calibração, instrumentação e controle poderão cooperar entre si no desenvolvimento de metodologias de medição, análise numérica e automação. A plataforma permite o modelamento matemático e o estudo analítico e/ou numérico de processos utilizando recursos disponíveis de cloud computing e a apresentação dos resultados e geração de relatórios através de uma interface web. Através do desenvolvimento cooperativo de novos procedimentos para análise de dados, modelos de instrumentos e melhorias operacionais do sistema pretende-se que o mesmo aproxime-se cada vez mais de um LIMS totalmente funcional e que tenha maior aderência à norma ISO/IEC 17025.

Palavras-chave: LIMS; Computação em nuvens; Automação; Modelagem matemática.

CLOUD COMPUTING SYSTEM FOR LABORATORY PROCEDURES AUTOMATION

Abstract

This work aims to present the design of a cloud computing platform where professionals from various segments of metrology, calibration, instrumentation and control may cooperate in the development of measurement methodologies, numerical analysis and automation. The platform allows creation of mathematical models for analytic and/or numerical evaluation of processes using cloud computing resources for further reporting and presentation of results through a web interface. By means of the cooperative development of new procedures for data analysis, instruments models and general system improvements is intended that the software be a fully functional LIMS and achieve greater adherence to ISO / IEC 17025.

Keywords: LIMS; Cloud computing; Automation; Mathematical modelling.

¹ Eng^o Eletricista, Pesquisador Assistente do laboratório de Metrologia elétrica do Instituto de Pesquisas Tecnológicas do estado de São Paulo, Brasil.

1 INTRODUÇÃO

Sistemas para gestão de informações laboratoriais, ou mais conhecidos pela sigla LIMS, já foram desenvolvidos e estudados por diversas empresas, das quais podemos citar em especial o Autolab [1] que inspirou em muitos aspectos este trabalho. Estes sistemas são amplamente utilizados como ferramenta para aumentar a qualidade e produtividade dos processos de um laboratório, sendo softwares geralmente desenvolvidos para plataforma Windows que costumam ser implantados no ambiente do laboratório ou empresa em que é utilizado. Na última década, a internet evoluiu de forma que diversos aplicativos passaram a funcionar utilizando parcialmente ou totalmente uma infraestrutura remota ao cliente, como citado em [2], seguindo o modelo SaaS (Software as a Service) e facilitando o acesso e interoperabilidade de tecnologias. O presente projeto propõe um sistema LIMS que utiliza e interage com serviços nas nuvens e incentiva o desenvolvimento colaborativo de novos procedimentos para medição e cálculo numérico.

2 MATERIAIS E MÉTODOS

O software OpenU (nome dado para este projeto) está sendo desenvolvido utilizando o framework Ruby on Rails com o banco de dados PostgreSQL e interage com outras tecnologias de computação nas nuvens. Diversas bibliotecas OpenSource em JavaScript, como JQuery, jsVISA, Github.js entre outras, disponibilizam funções para tornar exequível a interações entre o usuário, sistema LIMS e serviços nas nuvens. Uma API para computação científica fornecida pelo servidor OpenCPU [3], que possibilita a execução de funções matemáticas e geração de gráficos utilizando a linguagem estatística R [4] é o principal aplicativo que interage com o OpenU.

2.1 Fluxo de Informações

O sistema apresentado disponibiliza para seus usuários uma sequência de funcionamento. Diante de um item a ser testado, que pode ser um instrumento ou amostra, uma calibração ou ensaio pode ser realizado utilizando o OpenU, se seguirmos alguns passos. Estes passos se iniciam no modelamento das características do item, passando pela definição de procedimentos de análise, interação com a instrumentação utilizada, modelamento matemático e processamento numérico até a apresentação dos resultados (Figura 1).

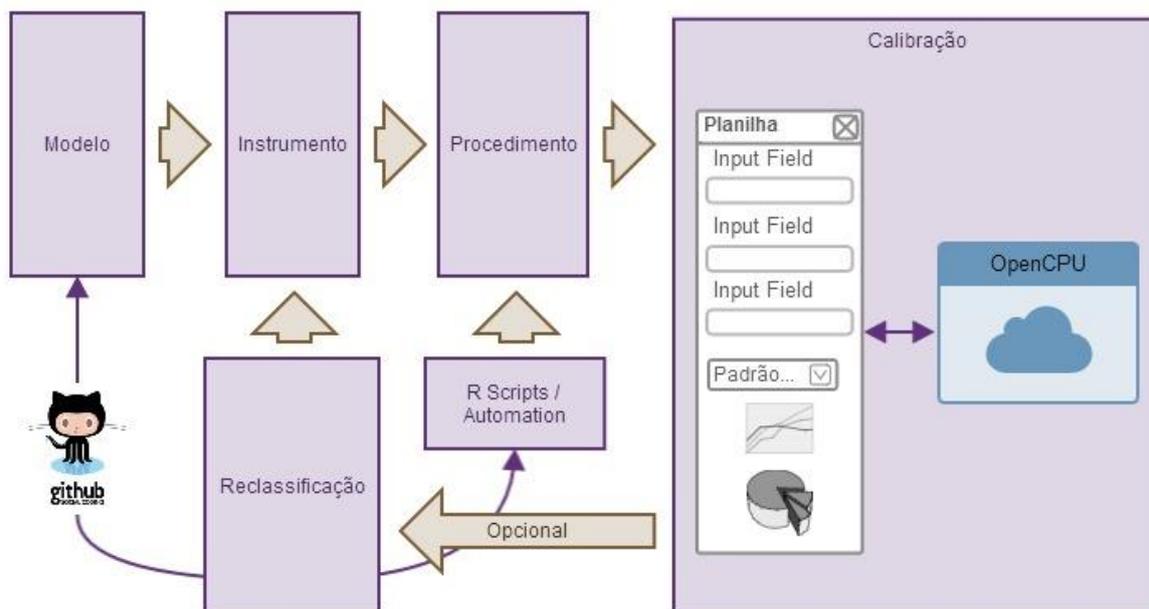


Figura 1. Fluxo para realização de testes

Cada etapa deste fluxo fornece formulários que devem ser preenchidos para que as etapas seguintes possam ser executadas. Algumas delas disponibilizam sub-etapas que atendem casos mais específicos vivenciados em um laboratório de calibração. A tabela a seguir (Tabela 1) explica a função de cada etapa e as informações que nelas são registradas.

Tabela 1. Atribuições das etapas e informações registradas

<i>Etapa</i>	<i>O que é definido</i>	<i>Informações geradas</i>
<i>Modelo</i>	<i>Características físicas e informações que não se alteram com o tempo de um dado modelo de instrumento ou amostra</i>	<i>faixas, fabricante, erro máximo admissível, etc.</i>
<i>Instrumento</i>	<i>Detalhes mais específicos para o exemplar de um instrumento</i>	<i>Porta de comunicação, número de série, histórico de calibração e reclassificação.</i>
<i>Procedimento</i>	<i>Planilha de entrada para ensaio ou calibração e modelo matemático</i>	<i>Variáveis de entrada, número de leituras, formatação da planilha, método para entrada das medições e cálculo.</i>
<i>Reclassificação</i>	<i>Alterações em um modelo</i>	<i>Incerteza de calibração, correções para leituras.</i>
<i>R Scripts / Automação</i>	<i>Funções para automatizar procedimentos e extrair resultados dos dados de entrada</i>	<i>Código fonte em JavaScript para automação e R para análise numérica.</i>
<i>Calibração</i>	<i>Procedimentos para a realização dos testes</i>	<i>Leituras realizadas, resultado das medições.</i>

2.2 Automação

O sistema oferece uma interface para automação de instrumentação através de uma camada de abstração do protocolo VISA (Virtual Instrument Software Architecture), possibilitando a aquisição de dados por meio de qualquer navegador web moderno. A solução utilizada para a automação de instrumentos e aquisição de dados baseia-se em um servidor WebSockets que fornece uma API para que o navegador WEB, através de uma biblioteca feita em JavaScript chamada jsVISA [5] comunique-se com os instrumentos. Esta biblioteca comunica-se com o protocolo VISA do computador no qual o servidor WebSockets é executado, utilizando o interpretador JavaScript nativo do navegador para realizar a automação. O servidor tem como tarefa receber os comandos que são identificados com uma chave única e previamente vinculadas a uma função de retorno no navegador, escreve-os no instrumento e enviar novamente as informações para o navegador com a chave original. O navegador recebe as informações e executa a função de retorno (Figura 2). Esta função tem como objetivo principal registrar os valores na planilha de calibração.

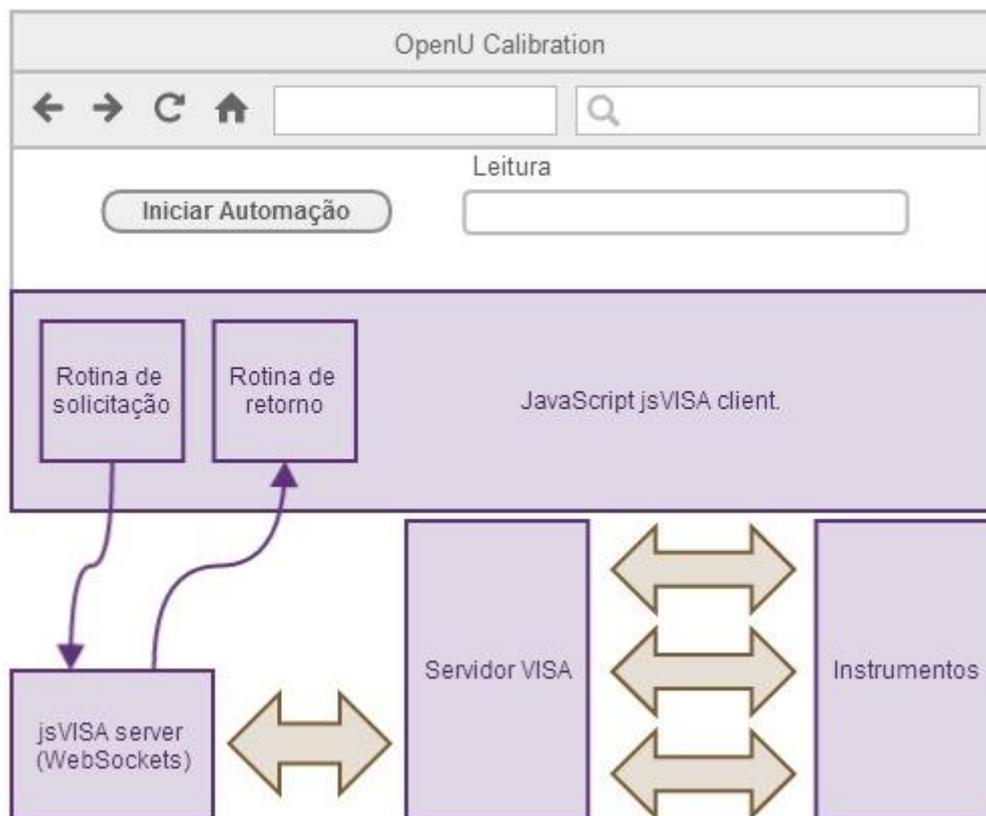


Figura 1. Interação entre o navegador e o servidor jsVISA para automação.

2.3 Procedimentos e Cálculo numérico

Procedimentos descrevem o passo a passo para a realização de uma calibração ou ensaio. Neles são definidas as variáveis de entrada, quantidade de leituras e o modelo matemático que calcula o desvio das medidas. No momento em que se define o procedimento, também é associado o método para aquisição e

processamento dos dados. Pode-se programar e reaproveitar diversos algoritmos e vincula-los a diferentes procedimentos.

Para análise numérica, um script que pode ser padrão ou definido por usuário é gravado no servidor OpenCPU através de um cliente http embutido na aplicação. Após a compilação de todas as informações envolvidas no teste, os dados da planilha de calibração em questão são encapsulados em um objeto JSON, enviados ao OpenCPU que em seguida retorna o resultado para o navegador (Figura 3).



Figura 2. Requisições entre OpenU e OpenCPU para processamento numérico

2.4 Colaboração

O sistema de versionamento do OpenU utiliza o site GitHub [6] como repositório. O GitHub faz uso do SCM (source code management) Git, que é utilizado por muitos projetos open source e ajuda, especialmente quando se trabalha em equipe, a controlar as modificações e a autoria das alterações no código e das melhorias implementadas ao sistema.

Os modelos, scripts R e scripts de automação são as informações que podem ser compartilhadas através do site. Após o envio dos dados para o GitHub, estas passam a ser acessíveis para desenvolvimento colaborativo.

3 RESULTADOS E DISCUSSÃO

O software foi testado para algumas aplicações no laboratório de metrologia elétrica do IPT. Um procedimento automatizado para as calibrações de resistor padrão e o desenvolvimento de um padrão para medição de distorção harmônica foram algumas das aplicações alcançadas com o novo sistema.

Como validação do script para cálculo de incertezas e da integração dos dados entre os módulos do OpenU, foi solucionado o problema apresentado na página 21 do ISO GUM [7], onde se deseja obter as estimativas de incerteza para uma medida de potência. O problema já havia sido solucionado passo a passo em [8], e os resultados obtidos utilizando o software são totalmente compatíveis, como mostra a imagem obtida do módulo de calibração do OpenU (Figura 4).

Calcular U: 0.037, k: 2.05, Veff: 53

Mostrar 10 registros

Var.	Leitura	Descrição	Incerteza	Div. Distribuição	Coef.	Contribuição
alpha	0.002	alpha (meas)	0	1	48.32347	0
alpha	-	alpha (Ema)	0.0003	1.732050808	48.32347	0.00007
R	5.07	R (meas)	0	1	1.92531	0
R	-	R (Ema)	0.01	1.732050808	1.92531	0.00012
t	25	t (meas)	0	1	0.01933	0
t	-	t (Ema)	0.1	1.732050808	0.01933	0
V	7.07	V (meas)	0	1	-2.76134	0
V	-	V (Ema)	0.005	1.732050808	-2.76134	0.00006
VI	9.7616666666667	VI (meas)	0.00788106027835803	1	1	0.00006

Mostrando de 1 até 9 de 9 registros

Figura 3. Apresentação dos resultados.

As componentes que na coluna descrição aparece entre parênteses a sigla “EMA” tem como origem o modelo do instrumento. As que contêm a palavra “Leitura” são originadas da variabilidade das leituras e calculadas pelo script R, assim como os coeficientes de sensibilidade.

3.1 Aquisição de Dados e Processamento Numérico

O método para aquisição de dados do OpenU foi idealizada com a preocupação de alcançar maior fluidez e compatibilidade com a WEB. Nas aplicações de computação nas nuvens, as rotinas acontecem de forma assíncrona, diferente do que estamos acostumados com sistemas supervisórios tradicionais.

Para exemplificar esta diferença, podemos citar uma rotina de automação, onde utilizando um multímetro para medir a tensão de um termopar desejamos obter o correspondente em temperatura. Em aplicações desktop tradicionais, enquanto o primeiro passo ainda é executado, o segundo fica esperando e o aplicativo fica "bloqueado". Em ambiente Windows, se este tempo de espera for muito grande, a famosa mensagem “o programa não está respondendo” é exibida, que além de gerar transtorno ao usuário, também pode ocasionar entre outros problemas, perda de dados.

Em aplicações web, o tempo necessário para termos o valor de retorno de uma RPC depende do tempo de envio da requisição para o serviço remoto, processamento e retorno dos dados. Não temos como monitorar o que acontece no servidor remoto, portanto para sanar este problema de sincronismo utilizam-se funções de retorno, que já estão pré-definidas e tratam o resultado assim que o servidor sinaliza que a resposta está disponível para o navegador. E da mesma forma funcionam as requisições do OpenU ao servidor WebSocket para automação e ao serviço de processamento nas nuvens OpenCPU.

Como exemplo do emprego da metodologia supramencionada, podemos citar o desenvolvimento de um analisador de espectro. Para obter informações de amplitude e fase das harmônicas de um sinal, um processo que consiste na digitalização de formas de onda utilizando um multímetro de rápida amostragem, a aplicação da transformada rápida de Fourier e a realização de uma série de ajustes lineares foi implementado com sucesso (Figura 5).

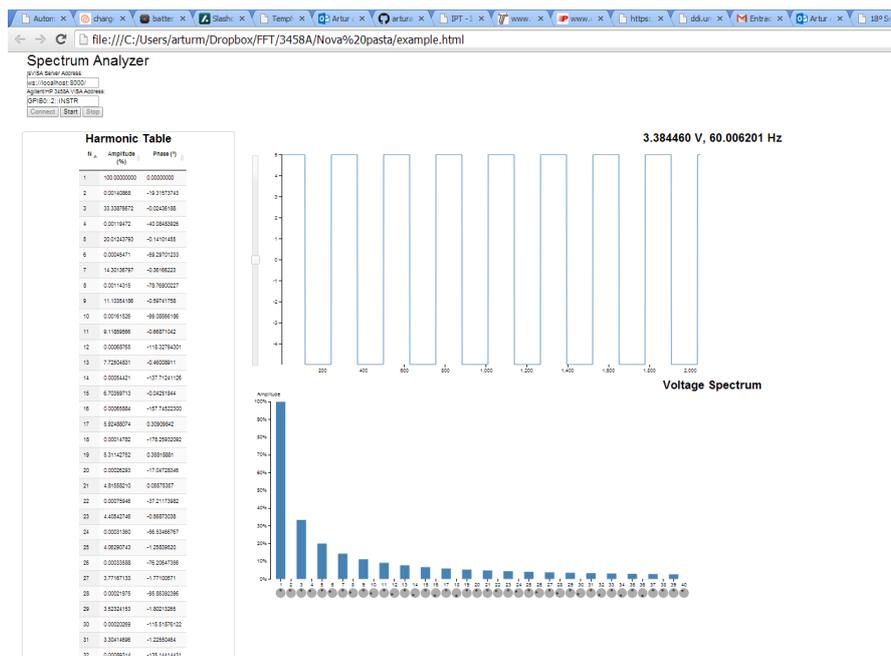


Figura 4. Interface do analisador de espectro.

Este analisador de espectro pode ser utilizado de qualquer navegador WEB moderno, dependendo apenas da instrumentação e do servidor jsVISA.

4 CONCLUSÃO

O desenvolvimento do OpenU resultou em um LIMS funcional, mas ainda em estágio inicial de testes. O modelo de desenvolvimento colaborativo através da WEB é novo para este tipo de software, mas diante de inúmeros projetos de sucesso que podem ser listados no GitHub, estima-se que este é um caminho válido e prospero.

REFERÊNCIAS

- 1 Stempniak CR, Marcon, F, LIMS para atendimento da norma ISSO/IEC 17025, Revista Analytica nº 30, 2007.
- 2 Armbrust M, Fox A, Griffith R, et al. A View of Cloud Computing; 2010.
- 3 Ooms JCL, Open Computing: Challenges and Opportunities, 2011.
- 4 The R Project for Statistical Computing. [acesso em 23/05/2014]. Disponível em: <http://www.r-project.org/>.
- 5 Biblioteca JavaScript jsVISA. [acesso em 23/05/2014]. Disponível em: <https://github.com/arturaugusto/jsVISA>.
- 6 GitHub, acesso em 23/05/2014, disponível em: <https://github.com>.
- 7 Evaluation of measurement data. Guide to the expression of uncertainty in measurement. [acesso em 23/05/2014]. Disponível em: <http://www.bipm.org/en/publications/guides/gum.html>.
- 8 Yehia R. Guia para a estimativa da incerteza de medição. Instituto de Pesquisas Tecnológicas; 2008.