

mente especializado e é difícil de ser ensinado, fazendo com que haja um grande ônus na manipulação e agregação desses dados. Na verdade, o desejado seria que os pesquisadores do LabInstru tivessem menos esforço nesta etapa, para que seu tempo e atenção fossem dispensados às questões de sua área particular de pesquisa. Considerando esta preocupação, surgiu a motivação para a elaboração do projeto e posterior implementação de um sistema de informação para este domínio, o qual será apresentado adiante.

3. *Framework Web2Py*

A utilização da internet como ambiente para o desenvolvimento de sistemas tem crescido continuamente devido à facilidade proporcionada por essa plataforma, possibilitando acesso remoto ao sistema, necessitando apenas de um navegador compatível e de acesso a internet [Osório 2015]. Assim, uma aplicação web vai além de uma simples página com conteúdo, é, na verdade, uma aplicação cliente-servidor na qual é possível: cadastrar, recuperar, atualizar e deletar informações em um banco de dados, interagir com esses dados por meio de consultas ou relatórios e solucionar problemas que envolvam automatização de processos através de programação. Tais aplicações são geralmente escritas usando linguagens de marcação e de programação.

Ao desenvolver uma aplicação web várias funções são similares e repetidas diversas vezes, como a construção de formulários, validação de dados, conexão com o banco, entre outras ações de gerenciamento. Neste cenário, *frameworks* podem auxiliar a obter software de qualidade ao passo que reduzem o esforço de desenvolvimento. Os principais benefícios decorrentes da utilização de *frameworks* advêm da modularidade, reusabilidade, extensibilidade e inversão de controle proporcionados [Fayad et al. 1999, Pinto 2000].

O *framework* Web2Py foi identificado como sendo o mais adequado para o desenvolvimento da plataforma proposta. Este *framework* é voltado para o desenvolvimento web, é livre e gratuito e foi implementado na linguagem de programação Python por uma equipe liderada por Massimo di Pierro em colaboração com mais de 70 desenvolvedores. Um dos principais objetivos do Web2Py é proporcionar agilidade no desenvolvimento de aplicações web seguras e baseadas em bancos de dados [di Pierro 2013].

O Web2py foi projetado levando em consideração que o desenvolvedor venha a adotar boas práticas da Engenharia de Software, tais como a adoção do padrão arquitetural *Model-View-Controller* (MVC). Este padrão tem o objetivo de separar os dados (*Model*) da interface com o usuário (*View*), a fim de que as alterações em uma camada não afetem a outra [Gordon et al. 2012]. Este padrão desacopla o acesso a dados e a lógica de negócio da apresentação de dados e interação com o usuário, por meio da introdução do componente intermediário, o *Controller* [Gamma et al. 2005]. O padrão MVC é utilizado, em geral, no desenvolvimento de sistemas com interface de usuário, focados na flexibilidade e reutilização de código.

O Web2py também é um *framework full-stack*, ou seja, a partir do momento que o pacote do Web2py é obtido, todas as ferramentas para o desenvolvimento de uma aplicação estão presentes nele, sem a necessidade de instalação do mesmo e de suas dependências, dispensando também configurações manuais. Em um *framework* PHP (*Personal Home Page*), por exemplo, são necessárias as instalações de um *web server*, como o Apache, de um SGBD (MySQL, por exemplo) e da própria linguagem de programação, além da configuração dos mesmos [Duarte 2016, Gordon et al. 2012]. Isto enfatiza a simplicidade e uma das vantagens do Web2Py em relação à outros *frameworks* existentes.

O Web2py possui uma camada de abstração de banco e dados que gera dinamicamente todo o código SQL. A classe DAL (*Database Abstraction Layer*) pode se conectar e lidar de forma transparente com diversos bancos de dados como: SQLite, MySQL, PostgreSQL, Oracle, entre outros. O Web2py gera uma interface administrativa básica para acesso aos dados, com a possibilidade de inserir, deletar, editar e consultar dados diretamente [di Pierro 2013].

4. Solução Proposta

A solução proposta para o problema identificado, intitulada *Plataforma de Estações LabInstru*, consiste em uma plataforma web, desenvolvida com o *framework* Web2py, para armazenamento, consulta e disponibilização dos dados oriundos das diferentes estações meteorológicas. Nesta plataforma, as seguintes funcionalidades estão disponíveis:

1. Cadastrar estação meteorológica;
2. Cadastrar dados meteorológicos;
3. Consultar disponibilidade de dados;
4. Consultar dados meteorológicos.

O cadastro de uma estação meteorológica consiste no fornecimento de um identificador único para a mesma, que será utilizado para rotular a respectiva estação. Em termos práticos, o identificador da estação costuma estar associado ao local em que a mesma está instalada como, por exemplo, EST, Policlínica, Embrapa, etc. A Figura 4 mostra a interface gráfica para cadastro da estação na plataforma proposta.

web2py™				
HOME	NOVA ▾	CONSULTA	VERIFICAÇÃO	CONTATO
<h2>Nova Estação</h2>				
Escolha a estação		<input type="text"/>		
		<input type="button" value="ENVIAR"/>		
<input type="button" value="VOLTAR"/>				
<h3>Estações Cadastradas</h3>				
Nº	Estação			
1	EST			
2	MUSA			
3	IFAM			
4	INPA			
14	POLICLINICA			
15	RIONEGRO			

Figura 4. Captura da tela de cadastro de novas estações.

O cadastro de dados meteorológicos está sujeito à escolha de uma estação correspondente. Neste caso, o usuário fornece também um arquivo de medições oriundo do servidor das estações meteorológicas. O cabeçalho do arquivo é examinado para a captura de quais dados meteorológicos encontram-se disponíveis e qual a ordem em que são apresentados. Posteriormente, este cabeçalho é ignorado e apenas os dados meteorológicos são importados na plataforma, atrelados à estação previamente selecionada. A Figura 5 mostra a interface gráfica para cadastro de dados meteorológicos.

Alguns cuidados foram considerados no cadastro de dados meteorológicos: apenas o administrador da plataforma pode fazê-lo, evitando que outros usuários forneçam informações incorretas ou corrompidas. Algumas medições podem ter valores faltantes, indicados por NaN (*Not a Number*) no arquivo fornecido. Estes valores devem ser apropriadamente persistidos na plataforma, não sendo substituídos por valores nulos que, por