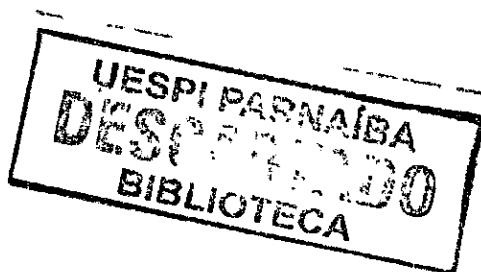
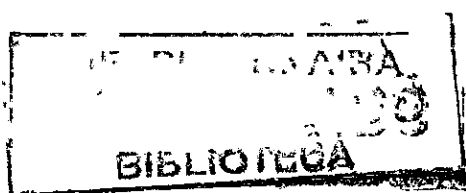


UNIVERSIDADE ESTADUAL DO PIAUÍ
CAMPUS PROFESSOR ALEXANDRE ALVES DE OLIVEIRA
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO



MARCELO DE SOUSA BARROS

MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA
GERENCIAMENTO DE PESQUISAS DE OPINIÃO PÚBLICA UTILIZANDO O
FRAMEWORK WEB2PY



Biblioteca UESPI - PHB
Registro Nº M596
CDD 658.403.8011
CUTTER B277m
V _____ EX. 01
Data 09 / 09 / 11
Visto #susa

PARNAÍBA - PIAUÍ

2011

MARCELO DE SOUSA BARROS

**MODELAGEM E DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA
GERENCIAMENTO DE PESQUISAS DE OPINIÃO PÚBLICA UTILIZANDO O
FRAMEWORK WEB2PY**

Monografia apresentada ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Piauí – UESPI, Campus Prof. Alexandre Alves de Oliveira, como parte das exigências para obtenção do título de Bacharel em Ciência da Computação.

Orientador: M.Sc. José Flávio
Gomes Barros

PARNAÍBA - PIAUÍ

2011



Ata de Apresentação de Trabalho de Conclusão de Curso

Aos quinze dias do mês de agosto de dois mil e onze, às 14h00, no Laboratório de Informática do Campus Prof. Alexandre Alves Oliveira (Parnaíba) – UESPI, na presença da banca examinadora, presidida pelo professor José Flávio Gomes Barros e composta pelos seguintes membros: Mayllon Veras da Silva e Jaclason Machado Veras, o aluno **Marcelo de Sousa Barros** apresentou o Trabalho de Conclusão de Curso de Graduação em Ciência da Computação como elemento curricular indispensável à colação de grau, tendo como título: **Modelagem e Desenvolvimento de uma Aplicação Web para Gerenciamento de Pesquisas de Opinião Pública Utilizando o Framework WEB2PY**. A banca examinadora reunida em sessão reservada deliberou e decidiu pelo resultado de **aprovado** ora formalmente divulgado ao aluno e aos demais participantes. Nada mais havendo a tratar, eu professor José Flávio Gomes Barros na qualidade de presidente da banca lavrei a presente ata que será assinada por mim, pelos demais membros e pelo aluno apresentador do trabalho. Parnaíba (PI), 15 de agosto de 2011.

OBS.:	<i>Nada a observar.</i>

Banca Examinadora

Prof. M.Sc. José Flávio Gomes Barros
Orientador, UESPI

Prof. Esp. Mayllon Veras da Silva
Avaliador, UESPI

Prof. M.Sc. Jaclason Machado Veras
Avaliador, UFPI

Aluno

Marcelo de Sousa Barros

***Em memória da minha inesquecível
Avó, Maria do Carmo de Sousa.***

AGRADECIMENTOS

Em primeiro lugar a **DEUS**, por todas as bênçãos que me tem concedido e por ter estado sempre ao meu lado, nos momentos de cansaço e de desânimo diante das dificuldades, nas lutas, nas vitórias e que mesmo nas derrotas me estendeu as mãos.

À minha família pelo apoio nos momentos difíceis vividos aqui nesta etapa da VIDA.

Aos meus Mestres da Universidade Estadual do Piauí - UESPI.

Ao Prof. M.Sc. José Flávio Gomes Barros, MEU ORIENTADOR E AMIGO, por ter confiado no meu trabalho a ser DESENVOLVIDO junto a UESPI.

Ao Prof. M.Sc. Jaclason Machado Veras pelo apoio e incentivos.

A todos os meus atuais e ex-colegas de trabalho.

Enfim, a todas as pessoas, que direta ou indiretamente, contribuíram para a concretização deste sonho.

RESUMO

O desenvolvimento de software é de fundamental importância para várias áreas, pois, proporciona suprir as necessidades exigidas pelo mundo globalizado. Atualmente o uso de frameworks no desenvolvimento de software é indispensável, pois, traz mais produtividade, consistência, gerando assim um produto com uma maior qualidade. Este trabalho apresenta o Web2py, um framework para desenvolvimento ágil de software web, escrito e programável em Python, baseado no Django e Ruby on Rails, que será utilizado no desenvolvimento de uma aplicação web para gerenciamento de pesquisas de opinião pública, denominada SYSPOP.

Palavras-Chave: Desenvolvimento na Web. Desenvolvimento ágil. Python. Framework Web2py. Pesquisa de Opinião Pública. SYSPOP.

ABSTRACT

Software development is of fundamental importance for many areas, therefore, provides the needs required by a globalized world. Currently the use of frameworks in software development is essential, therefore, carry greater productivity, consistency, thus creating a product with a higher quality. Web2py This paper presents a framework for agile software development web, written and programmable in Python, based on Django and Ruby on Rails, which will be used to develop a web application for managing public opinion polls, called SYSPOP.

Keywords: Agile Web Development. Python. Web2py Framework. Public Opinion Research. SYSPOP.

LISTA DE FIGURAS

Figura 1 – O Interpretador Python, no modo interativo.....	28
Figura 2 - Interface para gerenciamento do servidor.....	33
Figura 3 - Página da aplicação Welcome.....	34
Figura 4 - Página de login da aplicação Admin.....	34
Figura 5 - Página da Site.....	35
Figura 6 - Tela de edição de aplicações.....	37
Figura 7 - Diretório de uma aplicação.....	37
Figura 8 - Diagrama de Caso de Uso "Geral".....	41
Figura 9 - Diagrama de Classes.....	46
Figura 10 - Tela de login do SYSPOP.....	47
Figura 11 - Tela de Painel de controle do SYSPOP.....	47
Figura 12 - Tela de cadastro de usuários do SYSPOP.....	48
Figura 13 - Tela de cadastro de entrevistas do SYSPOP.....	48
Figura 14 - Tela de cadastro de entrevistados do SYSPOP.....	49
Figura 15 - Tela de cadastro de pesquisas do SYSPOP.....	49

SUMÁRIO

LISTA DE FIGURAS.....	7
1 INTRODUÇÃO	9
1.1 Justificativa e relevância	10
1.2 Objetivos do trabalho	10
1.3 Estruturação.....	11
2 PESQUISA DE OPINIÃO PÚBLICA	13
2.1 Fatos Históricos	13
2.2 Conceitos	15
2.3 Problemas.....	16
2.4 Soluções	17
3 A INTERNET E A WEB.....	19
3.1 Internet.....	19
3.2 A World Wide Web.....	20
4 A LINGUAGEM PYTHON	24
4.1 Visão geral.....	24
4.2 Compilação e interpretação	27
5 O FRAMEWORK WEB2PY	29
5.1 Visão geral.....	29
5.2 Principais características	31
5.3 O funcionamento do Framework Web2py.....	33
5.4 Exemplos de códigos.....	38
6 ANÁLISE E IMPLEMENTAÇÃO DO SISTEMA SYSPOP.....	39
6.1 Considerações iniciais	39
6.2 A modelagem.....	40
6.3 Diagrama de Caso de Uso do SYSPOP	40
6.4 Diagrama de classes	45
6.5 Interfaces do ambiente	46
7 CONCLUSÃO	50
7.1 Trabalhos futuros	50
8 REFERÊNCIAS	51
APÊNDICE	54

1 INTRODUÇÃO

As informações obtidas através de perguntas feitas às pessoas sobre o que elas fazem, fizeram ou pensam, são necessárias e úteis quando necessitamos decidir ou escolher sobre algo nas situações que ocorrem corriqueiramente no dia a dia, seja na vida particular ou na vida profissional das pessoas. Como exemplos temos: a decisão pela criação ou não de um determinado produto; a escolha sobre qual produto utilizar, a contratação de pessoas, a contratação de serviços de uma determinada empresa, dentre outras. Nessas situações são levados em consideração alguns dados, cuja obtenção é feita através da opinião de outras pessoas, de experiências vividas anteriormente, estudos feitos acerca do assunto em questão de forma geral.

Uma das formas mais utilizadas na pesquisa de opinião é o questionário, tornando-se um método muito eficiente para a coleta de dados. Esse tipo de pesquisa é usado, por exemplo, por empresas que desejam saber a satisfação de seus clientes em relação a algum determinado produto ou serviço, ou então para estudar a aceitação de determinado produto ou serviço pelo mercado ao qual se pretende atender, dentre muitos outros usos.

Diante da necessidade da busca de alguma vantagem competitiva, considerando o ambiente globalizado com alta competitividade em que vivemos, torna-se uma questão de sobrevivência nos tempos atuais a informação. Para que se possa tomar qualquer tipo de decisão é de fundamental importância termos informações acerca do que deve ser decidido.

Assim, com o crescimento exponencial do volume de informações disponíveis, torna-se imprescindível sua organização através do uso de Sistemas de Informação, e os mesmos, devido ao avanço e a popularização da Internet, vem se tornando instrumentos importantes e eficientes para a aplicação deste tipo de pesquisa.

Os Sistemas de Informação possuem diversas definições, mas torna-se necessário compreendermos do que realmente se trata, devido à sua importância nos dias atuais. Tais sistemas podem ser compreendidos como um conjunto de

componentes organizado e inter-relacionados, que permitem coletar, manipular e disseminar informações para auxiliar a tomada de decisão.

Portanto, esta pesquisa tem como objetivo geral a modelagem e o desenvolvimento de uma aplicação que auxilie na elaboração de pesquisas de opinião pública, bem como no processamento dos resultados obtidos.

1.1 Justificativa e relevância

As pesquisas de opinião fazem parte do dia a dia da população e têm uma grande visibilidade na mídia, sendo muito comum abrirmos jornais e revistas e encontrarmos textos citando o que pensam as pessoas a respeito dos mais diversos assuntos. Geralmente, as pesquisas de opinião são feitas por amostragem, ou seja, para saber a opinião dos moradores de uma cidade sobre um determinado assunto, não se entrevistam todos eles e sim apenas um grupo selecionado.

As pesquisas também são vistas como contendo propriedades pedagógicas e terapêuticas pelo fato de induzir um processo de autoconhecimento e um aprendizado informado sobre noções e conceitos populares vitais (ECHEGARAY, 2001).

Atualmente a busca por informação é crescente, sendo esta um bem imensurável e um diferencial para a tomada de decisões e devido a sua importância faz-se necessário que estejam disponíveis em tempo hábil. As pesquisas de opinião geram informações preciosas e dessa forma faz-se necessário a diminuição do tempo gasto na elaboração e processamento de resultados das pesquisas. Diante disto este trabalho se propõe a desenvolver uma aplicação web que permita mais velocidade na sua elaboração e processamento das informações obtidas, podendo assim acessar o sistema de qualquer lugar que tenha um computador com acesso a Internet.

1.2 Objetivos do trabalho

O objetivo geral deste trabalho é modelar e desenvolver uma ferramenta que auxilie no gerenciamento de pesquisas de opinião através de questionários com

questões fechadas, permitindo rapidez e facilidade na sua criação e na obtenção de resultados.

Para alcançar esse objetivo geral, os seguintes objetivos específicos deverão ser atingidos:

- Entender o funcionamento das pesquisas de opinião;
- Estudar a linguagem de programação Python;
- Compreender o funcionamento do framework Web2py;
- Contribuir com uma modelagem e implementação de uma aplicação web que auxilie no processamento dos resultados das pesquisas;

1.3 Estruturação

A estrutura dessa monografia compreende 07 (sete) capítulos, sendo eles, além desta introdução, os seguintes:

No Capítulo 2 traz uma revisão sobre Pesquisas de Opinião Pública, onde será destacada sua importância, e também será feita uma abordagem sobre seu surgimento, conceitos, problemas encontrados na sua aplicação e as soluções encontradas para estes.

Já o capítulo 3, faz uma introdução sobre a Internet, englobando os motivos que levaram à sua criação, sua definição e utilização. Também será feita uma rápida abordagem sobre a tecnologia Web, destacando seu surgimento, sua definição, estrutura e utilização.

O capítulo 4 faz uma introdução sobre a linguagem de programação Python, que foi utilizada neste trabalho, abordando seu surgimento, sua definição, vantagens e utilização.

Já no capítulo 5, apresenta o Framework Web2py abordando suas principais características e funcionalidades, assunto fundamental no contexto deste trabalho.

O capítulo 6 por sua vez, descreve a modelagem da aplicação SYSPOP e a implementação de um protótipo mostrando suas principais interfaces.

E, finalmente, o capítulo 7, expõe as conclusões acerca do trabalho, com ênfase nos resultados alcançados e nas perspectivas futuras.

2 PESQUISA DE OPINIÃO PÚBLICA

Este capítulo trata das pesquisas de opinião públicas, iniciando com uma abordagem sobre seu histórico, em seguida traz alguns conceitos, depois descreve alguns problemas encontrados na sua aplicação e para finalizar traz algumas soluções para esses problemas.

2.1 Fatos Históricos

A origem da expressão "opinião pública" está envolta em mistério. Na literatura da Grécia e Roma antigas, bem como ao longo da Idade Média, os filósofos tinham inteira consciência da importância da opinião das massas. Mas somente no século XVIII, entretanto, que se submeteu a expressão opinião pública a uma análise e tratamento sistemáticos. Durante os séculos XVII e XVIII escritores como Voltaire, Hobbes, Locke e Hume pagaram o seu tributo à força da opinião pública. Hobbes falou no mundo como sendo governado pela opinião. Já Lock considerou a opinião como uma das três categorias do direito e Hume deu expressão à clássica afirmação de que "é somente na opinião que o governo se fundamenta e esta máxima estende-se aos governos mais despóticos e militaristas, tanto quanto aos mais livres e populares". Blaise Pascal referiu-se à opinião pública como "Rainha do Mundo", ao que Voltaire replicou: "se a opinião é Rainha do Mundo, os filósofos governam a Rainha" (CHILDS, 1967).

Rousseau, escrevendo no século XVIII, fez uma das mais claras análises do conceito de opinião pública em sua época, aplicando sua teoria da infalibilidade popular ao estado, proclamando que "o desejo mais generalizado é também o mais justo". Este ponto de vista tem sido freqüentemente repetido desde então. Rousseau acreditava que mesmo o despotismo se apóia na opinião pública porque dizia ele, "o governo despótico é servil, mesmo quando fundamentado na opinião; pois você depende do preconceito daqueles a quem você governa pelo preconceito." Rousseau parece ter sido o primeiro a usar a expressão "l'opinion publique", e são valiosas suas considerações sobre as relações entre a opinião e o direito. Afirmou que "quem quer que se dedique à tarefa de legislar para um povo deve saber como manejar as opiniões e através delas governar as paixões dos homens" (CHILDS, 1967).

Um dos primeiros a discutir a importância da opinião pública como fator político e demonstrar que sentia os problemas de Relações Públicas inerentes à função pública, foi Jacques Necker, ministro da fazenda da França, cuja sua experiência levou-o a salientar as correlações existentes entre opinião pública e crédito público. Descobriu também que os salões da França desempenhavam um papel muito importante na formação da opinião pública de sua época, e que as opiniões da burguesia tinham realmente influência decisiva. Necker foi o autor da única análise pormenorizada do conceito de opinião pública na época da Revolução Francesa, embora fazendo inúmeras referências a assuntos tratados nos artigos efêmeros da época (CHILDS, 1967).

O primeiro exemplo conhecido de pesquisa de opinião foi uma enquete conduzida pelo *The Harrisburg Pennsylvanian* em 1824, a qual mostrava Andrew Jackson a frente de John Quincy Adams por 335 a 169 votos, na disputa pela presidência dos Estados Unidos. Esse tipo de enquete – tendenciosa e sem consistência científica – gradualmente tornou-se mais popular, mas permaneceu como um fenômeno local, geralmente restrito a uma única cidade. Em 1916, o *Literary Digest* deu início a uma pesquisa nacional (em parte como um exercício de alavancagem de vendas) e previu corretamente a eleição de Woodrow Wilson como presidente. Postando milhões de postais e contando os que retornaram, o *Digest* previu os vencedores das quatro eleições presidenciais seguintes.

Em 1936, todavia, o *Digest* teve uma surpresa desagradável, onde seus 2,3 milhões de "eleitores" constituíam uma vasta amostra, contudo, eram geralmente pessoas mais abonadas que tendiam a ser simpatizantes do Partido Republicano, cujo *Literary Digest* nada fez para corrigir esse viés. Na semana anterior à eleição, informaram que Alf Landon era muitíssimo mais popular do que Franklin D. Roosevelt. Na mesma época, George Gallup efetuou uma pesquisa muito menor, porém com base científica maior, na qual pesquisou uma amostra demograficamente representativa. Gallup previu corretamente a esmagadora vitória de Roosevelt. A *Literary Digest* rapidamente fechou as portas, enquanto a indústria das pesquisas de opinião levantou vôo.

Gallup inaugurou uma filial no Reino Unido, onde previu corretamente a vitória do Partido Trabalhista na eleição geral de 1945, em contraste com a opinião

de praticamente todos os outros analistas políticos, que esperavam uma vitória fácil do Partido Conservador, liderado por Winston Churchill.

Na década de 1950, as pesquisas de opinião haviam se espalhado pela maioria das democracias. Atualmente, atingiram praticamente todos os países, embora em sociedades mais autocráticas elas tendam a evitar questões políticas sensíveis.

2.2 Conceitos

Pesquisa de opinião pública é o que geralmente se atribui à opinião geral de uma sociedade. Uma pesquisa de opinião, sondagem de opinião ou estudo de opinião é um levantamento estatístico de uma amostra particular da vontade pública. Pesquisas de opinião geralmente são feitas para representar os pensamentos de uma população fazendo-se uma série de perguntas a um pequeno número de pessoas e então, extrapolando as respostas para um grupo maior dentro do intervalo de confiança.

A expressão *opinião pública* tem sido empregada em muitos sentidos diferentes, tantos, que sua utilização já foi posta em dúvida. Foram (e ainda são) feitas muitas tentativas de definição que todos pudessem aceitar e acaba por acrescentar mais uma à já longa lista. Virginia Sedmann, por exemplo, depois de cuidadosa análise de várias definições, concluiu que "opinião pública" é, para nós, "uma força ativa ou latente, derivada de um agregado de pensamentos, sentimentos e impressões pessoais, ponderados pelos vários graus de influência ou agressividade das opiniões individuais dentro do todo" (CHILDS, 1967).

As pesquisas definem interações entre elites políticas e cidadania e têm um papel central nos processos de democratização. Opinião pública é o conjunto de crenças e ideologias de determinado público sobre algum tema que recebe estímulos de comunicação, principalmente os midiáticos.

Algumas empresas que executam pesquisas de opinião pública no Brasil diferenciam o conceito de vontade popular (fruto da soma de opiniões individuais) do conceito de opinião pública, que é a construção coletiva de uma crença. Os níveis de informação e de mobilidade da população são determinantes para a formação da

opinião pública que se altera constantemente pela força e direção das ações midiáticas e pelo nível de informação da população.

A importância do estudo sobre a Opinião Pública se estabelece em cinco aspectos fundamentais: qual a opinião formada; como se formou; causas da formação; a estrutura social do público e o impacto do tema na formação da opinião; e a disposição do público para agir.

A pesquisa de opinião tem como objetivo investigar e retratar as atitudes e opiniões dos indivíduos, possibilitando à sociedade conhecer com precisão as tendências e posições dos diferentes segmentos sociais (ISMA, 2007).

Um órgão de pesquisa de opinião pública é o IBOPE, que utiliza a pesquisa de opinião pública como uma ferramenta eficaz para detectar com precisão posições e tendências dos diversos segmentos sociais. Baseada em dados científicos, a pesquisa de opinião é um excelente instrumento para identificar problemas e buscar soluções. Apontando, por exemplo, dados sobre o perfil e a imagem de marcas, empresários, políticos, entidades ou instituições (IBOPE, 2004).

Diante do contexto apresentado, a pesquisa de opinião pública traz outra característica que é avaliar a repercussão de novos acontecimentos ou ações governamentais na opinião pública ou como reagiriam setores da população diante da aprovação de projetos de lei que afetariam a vida da comunidade.

2.3 Problemas

Os resultados de uma pesquisa devem ser confiáveis e para isso deve-se evitar erros que viciem a pesquisa e comprometam negativamente todo um trabalho, e a coleta de dados deve ser feita de forma precisa. Segundo Néto (2004), situações como: o pesquisador evitar casas em que há cachorros ou em lugares de difícil acesso, incluir amigos na amostra, ou dela excluir pessoas com quem antipatiza, dentre outras, podem viciar a investigação.

Quando se faz cruzamento de variáveis, as perguntas abertas também se mostram problemáticas – em se tratando de pesquisas quantitativas. Um leque muito grande de variações de respostas pode gerar cruzamentos sem significado

algum, face à excessiva pulverização de variáveis. Estes cruzamentos servem para verificar o quanto uma variável influencia em outra, para verificar a consistência de respostas (se respondeu “a” na pergunta 2, deveria responder “c” na pergunta 8, por exemplo), dentre outras (NÉTO, 2004).

2.4 Soluções

Para que uma pesquisa possa ter resultados de credibilidade, é necessário que a coleta dos dados seja feita com a maior precisão possível. Assim, deve-se saber o que vai ser pesquisado, tendo-se em vista o problema de pesquisa, as respostas prováveis desse problema – as hipóteses – que serão testadas mediante a investigação científica, o lugar em que se fará a pesquisa – o universo ou a amostra, parte do universo de pesquisa.

Deve-se evitar – no caso das entrevistas – a inquirição de pessoas determinadas, ou aquelas mais fáceis de serem localizadas. Para evitar erros que viciam uma pesquisa e podem colocar por terra todo um trabalho é que existem os métodos. “Método” é o caminho para chegar a um objetivo. Um caminho que é feito de regras, as quais lhe dão cientificidade. É algumas destas regras se traduzem em etapas que se deve seguir, para chegar ao objetivo. Este objetivo é a prova científica, uma verdade relativa, sob eterna verificação e, portanto, sempre e sempre sujeito à refutação (NÉTO, 2004).

As técnicas estatísticas levam a selecionar um grupo representativo de todos os habitantes do município. A opinião deste grupo tende a refletir o que toda a comunidade pensa.

As técnicas utilizadas pelo IBOPE observam a coleta de dados feita por meio de amostras representativas da população estudada. Estas são definidas a partir de dados censitários levantados pelo Instituto Brasileiro de Geografia e Estatística (IBGE) e de um Levantamento Socioeconômico (LSE) realizado pelo próprio IBOPE. Por meio do LSE, o IBOPE tem acesso a vários tipos de informação sobre uma dada população: infra-estrutura urbana, condição socioeconômica e demográfica dos domicílios, posse de bens como televisor, computador, automóvel, TV por Assinatura e antena parabólica, acesso à Internet, entre outras. A partir

desses dados, as amostras são determinadas de modo a representar o universo a ser pesquisado. Existem diversas formas de classificação de pesquisas de acordo com o propósito do estudo em questão..Cada grupo exige a aplicação de um método de pesquisa diferente.

3 A INTERNET E A WEB

Este capítulo faz uma abordagem sobre a Rede mundial de computadores e a Web, abordando como surgiram, seus criadores, descrevendo também como funcionam.

3.1 Internet

A Internet surgiu como resposta à preocupação do governo americano, durante a guerra fria, de como deveria ser a comunicação militar caso ocorresse uma guerra nuclear (RUTHFIELD, 2001). Numa situação como essa, as tecnologias tradicionais não funcionariam, pois um sistema centralizado poderia ser facilmente destruído (RUTHFIELD, 2001). A Internet ou rede mundial de computadores foi criada na década de 60, pela Agência de Projetos de Pesquisas Avançadas do Departamento de Defesa dos Estados Unidos da América – ARPA (Department of Defense's Advanced Research Projects Agency), mais tarde chamada de DARPA, com o propósito ser uma rede capaz de suportar a um ataque nuclear, funcionando mesmo que outras máquinas da rede fossem destruídas, permitindo a comunicação entre as bases militares e os departamentos de pesquisa do governo americano.

Em 1972, um setor do departamento de defesa americano fez a primeira demonstração pública da ARPANet, uma rede de computadores que foi a precursora da Internet (LEINER, 1997) e, em 1983, a tecnologia da ARPANet foi substituída por uma tecnologia chamada Transmission Control Protocol/Internet Protocol (TCP/IP), a qual era mais adequada para redes com grandes quantidades de servidores (RUTHFIELD, 2001). Muitos consideram essa data como sendo o início oficial da Internet (RUTHFIELD, 2001).

O funcionamento da Internet baseia-se em protocolos, ou seja, conjuntos de padrões de regras e procedimentos para o controle da comunicação entre computadores (O'BRIEN, 2003). A rede mundial só é possível em função da aceitação de adoção de um padrão comum de protocolos: o TCP/IP (Transport Control Protocol – Protocolo de Controle de Transmissão – e o Internet Protocol – Protocolo de Internet) (VIDGEN et al., 2002; O'BRIEN, 2003).

Esta rede posteriormente começou a expandir-se para um espaço global, envolvendo universidades, agências governamentais, centros de pesquisa, dentre outros, passando a ser um canal não só de troca de informações militares, mas um canal de comunicação, colaboração, divulgação do conhecimento e de descobertas científicas, permitindo acesso rápido a informação. Vale lembrar que não há um dono da Internet, ela existe como resultado da cooperação entre milhares de pessoas espalhadas pelo mundo, trabalhando nos mais variados ambientes (DYSON, 1998).

Portanto, a Internet é um vasto conjunto de redes diferentes que utilizam certos protocolos comuns e fornecem determinados serviços comuns (TANENBAUM, 2003). A Internet pode ser considerada uma infraestrutura genérica de comunicação sobre a qual novas aplicações podem ser concebidas (LEINER, 1997).

Até o início da década de 1990, a Internet era um verdadeiro reduto de pesquisadores ligados às universidades, ao governo e à indústria. De acordo com Tanenbaum (2003) uma nova aplicação, a WWW (World Wide Web), mudou essa realidade e atraiu para a rede milhares de novos usuários, sem a menor pretensão acadêmica e segundo Kurose e Ross (2006) o surgimento da World Wide Web foi o principal evento da década de 1990, que levou a Internet para os lares e as empresas de milhões de pessoas no mundo inteiro, servindo de como plataforma para a habilitação e a disponibilização de centenas de novas aplicações.

3.2 A World Wide Web

A Web (também conhecida como WWW) teve início em 1989 no CERN, o centro europeu para pesquisa nuclear (TANENBAUM, 2003). A proposta inicial para uma teia de documentos vinculados veio de um físico do CERN, Tim Berners-Lee, em março de 1989 (TANENBAUM, 2003).

A World Wide Web, WWW ou simplesmente Web, foi desenvolvida para ser "um pool do conhecimento humano, que permitisse colaboradores em locais distantes compartilhar idéias e todos os aspectos de um projeto comum" (BERNERS-LEE et. al., 1994). Ela deveria permitir que documentos desenvolvidos

separadamente pudessem ser “ligados” facilmente e visualizados em um mesmo ambiente sem que isso exigisse grandes mudanças nem que possíveis mudanças precisassem ser feitas de forma centralizada (BERNERS-LEE et. al., 1994). A World Wide Web é uma estrutura arquitetônica que permite o acesso a documentos vinculados espalhados por milhões de máquinas na Internet (TANENBAUM, 2003). A Web foi criada com o propósito de permitir, mesmo que geograficamente distantes, a colaboração de grupos de cientistas de vários países, através da troca de informações na forma de relatórios, plantas, desenhos, fotos e outros documentos, baseando-se em mecanismos de armazenamento, recuperação e visualização de documentos eletrônicos ou páginas web.

Conforme Kurose (2006), a Web é uma aplicação cliente-servidor que permite aos usuários obter documentos de servidores Web por demanda, consistindo em um padrão para formato de documentos (HTML¹), browsers Web (por exemplo, o Microsoft Internet Explorer), servidores Web (por exemplo, servidores Apache) e um protocolo de camada de aplicação (HTTP²), que define o formato e a seqüência das mensagens que são passadas entre o browser e o servidor Web, sendo apenas um pedaço da aplicação Web.

Devemos observar que a Web é o lado gráfico da Internet, portanto são coisas distintas. A Web enriqueceu a Internet, deixando o conteúdo desta mais atraente, permitindo a incorporação de imagens e sons, além de prover um novo sistema de localização de arquivos, criando um ambiente no qual cada informação tinha um endereço único e poderia ser encontrada por qualquer usuário da rede. Para que uma página apontasse para outra é necessário um mecanismo de nomenclatura e localização de páginas. Para Tanenbaum (2003), três perguntas tinham de ser respondidas antes de uma página selecionada poder ser exibida:

- Qual é o nome da página?
- Onde a página está localizada?
- Como a página pode ser acessada?

¹ HTML - Linguagem de Marcação de Hipertexto (HyperText Markup Language)

² HTTP - Protocolo de Transferência de Hipertexto (Hypertext Transfer Protocol) – protocolo de camada de aplicação da Web, está no coração da Web e é definido no [RFC 1945] e no [RFC 2616] (KUROSE, 2006).

Como solução é atribuído um URL (Uniform Resource Locator) a cada página, sendo o nome universal da página. Segundo Tanenbaum (2003) as URLs são compostas por três partes: o protocolo (também chamado esquema), o nome DNS³ da máquina em que a página está e um nome local que indica a página específica (normalmente, um nome de arquivo na máquina onde ele reside). Como exemplo, o URL para a página oficial do Framework Web2py que contém o livro sobre o framework: <http://www.web2py.com/book/default/chapter/02>.

No exemplo acima temos o protocolo (HTTP), o nome DNS do host (www.web2py.com) e o nome do arquivo ([book/default/chapter/02](http://www.web2py.com/book/default/chapter/02)), sendo este último o caminho relativo ao diretório Web padrão em [web2py.com](http://www.web2py.com).

A base da Web é a transferência de páginas da Web do servidor para o cliente. Em sua forma mais simples, as páginas da Web são estáticas, isto é, são apenas arquivos que ficam armazenados em algum servidor esperando o momento de serem recuperados. Nesse contexto, até mesmo um vídeo é uma página da Web estática, porque é simplesmente um arquivo (TANENBAUM, 2003).

Segundo Junior (2003), podemos definir a tecnologia Web como sendo um sistema de padrões que inclui:

- Padrão de endereçamento: todos os recursos da Web têm um endereço único e podem ser localizados de qualquer lugar, independente da plataforma onde o recurso resida. Cada endereço é chamado de URL (Uniform Resource Locator).

- Padrão de comunicação: a tecnologia Web utiliza um protocolo de comunicação, ou seja, uma linguagem que permite a solicitação e obtenção de recursos da Web. Este protocolo, chamado HTTP (Hypertext Transfer Protocol), permite a busca de recursos em diversos formatos e não somente de hipertexto como o nome sugere.

- Padrão de estruturação das informações: o padrão inicial da tecnologia Web para apresentação das informações estava baseado em uma linguagem de marcação chamada HTML (Hypertext Markup Language). Esta Linguagem define principalmente elementos para visualização de informações. Entretanto, uma

³ DNS – Sistema de nomes de domínios (Domain Name System)

extensão da tecnologia Web foi a definição da metalinguagem chamada XML⁴ a qual permite definir de forma extensível como uma informação pode ser estruturada.

O crescimento da Internet causou um impacto significativo nos negócios, comércio, indústria, bancos, finanças, educação e setores de entretenimento e na vida particular das pessoas (MURUGESAN et al., 1999). Para Ginige (2000) um fato que ilustra esse impacto é o tempo de apenas quatro anos que levou para atingir 50 milhões de pessoas, sendo que o telefone, o rádio e televisão levaram 75, 35 e 15 anos, respectivamente, para atingir o mesmo número de pessoas. Segundo Tanenbaum (2003) sua enorme popularidade se deve à sua interface gráfica colorida, de fácil utilização para principiantes. Seu crescente uso fez surgir uma nova geração de sistemas de informação, que combinam navegação por meio de um espaço heterogêneo utilizando operações de consulta que incidem diretamente na informação apresentada. Do ponto de vista do projeto de aplicações Web, introduziu-se a hipermídia aos sistemas de informação (SCHWABE e ROSSI, 1998).

No mês de outubro de 1994, foi criado o World Wide Web Consortium (W3C), um consórcio internacional formado por empresas, instituições, pesquisadores, desenvolvedores e público em geral, com a finalidade de desenvolver a web a seu potencial máximo, criando normas e especificações aplicáveis aos diversos segmentos e setores da web, desde tecnologias e software até fabricantes e fornecedores (SILVA, 2011).

A história do desenvolvimento de aplicações na Web é relativamente curta. No princípio, havia apenas hipermídia, sistemas compostos por documentos conectados entre si por elos pré-definidos, com pouca ênfase no planejamento ou projeto e poucos foram propriamente testados. Atualmente, existem sistemas na Web que são aplicações de grande porte, que envolvem interação sofisticada de usuários a base de dados e, ainda, algumas dessas aplicações podem ser consideradas de missão crítica (GRIFFITHS et al., 2002).

⁴ XML - Linguagem de Marcação de Dados (eXtensible Markup Language)

4 A LINGUAGEM PYTHON

Este capítulo faz uma abordagem sobre a linguagem de programação Python, abordando seu surgimento, características, vantagens e utilização.

4.1 Visão geral

Python é uma linguagem de programação de altíssimo nível (VHLL – Very High Level Language), multiplataforma, interpretada, com tipagem forte e dinâmica, podendo ser utilizada como linguagem de script, permitindo automatizar tarefas e adicionar novas funcionalidades. Também possui estruturas de alto nível, como listas, dicionários, tuplas, strings, arquivos, classes, entre outros e vários módulos, que são conjuntos de funções pré-compiladas e portáveis, que implementam desde a manipulação de arquivos, chamadas do sistema, sockets, e até mesmo acesso a bibliotecas de construção de interfaces gráficas, como Tk. Encontram-se disponíveis para a maioria das plataformas no site oficial da linguagem: <http://www.python.org/> o interpretador Python e sua biblioteca padrão.

Python é uma linguagem orientada a objetos utilizada em uma variedade de domínios, tanto para programas independentes como para aplicações de script. Ela é gratuita, portátil, poderosa e fácil de usar (LUTZ, 2007).

Python foi desenvolvida em 1990 pelo holandês Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), para ser de uso geral, com objetivo de abranger vários domínios de aplicação. Seu nome foi inspirado no título do programa humorístico de TV "Monty Python's Flying Circus".

Além de ser uma linguagem de programação bem projetada, o Python também é útil para executar tarefas do mundo real - os tipos de coisas que os desenvolvedores fazem todo dia. Ele é comumente usado em diversos domínios, como ferramenta para escrever outros componentes e implementar programas independentes. Na verdade, como linguagem de propósito geral, as funções do Python são praticamente ilimitadas (LUTZ, 2007).

Python possui licença compatível com GPL (General Public License), mas com menos restrições, sendo um software de código aberto o que permite sua

utilização em produtos proprietários. A Python Software Foundation (PSF) é uma organização sem fins lucrativos, presidida por Guido Van Rossum, que se dedica a Linguagem Python e mantém a especificação da mesma. A implementação oficial do Python é mantida pela PSF e escrita em C, e por isso, é também conhecida como CPython (BORGES, 2010).

Python é um exemplo de FLOSS (Free/Libre and Open Source Software). Traduzindo, você pode distribuir livremente cópias deste software, ler seu código-fonte, modificá-lo, usar trechos em novos programas livres e tudo o que você quiser fazer. FLOSS é baseado no conceito de uma comunidade que compartilha conhecimento (FERRARI, 2010).

Python não exige altos investimentos em treinamentos, pois possui uma sintaxe coerente e simples, permitindo que seu aprendizado seja fácil. Também possui uma comunidade Python do Brasil que é uma das mais ativas, onde podemos obter ajuda de alta qualidade.

Python tem diversas vantagens em sua utilização, dentre as quais podemos citar:

- **Qualidade de software:** devido à sintaxe simples, permite um código mais legível de modo que programas escritos possam ser facilmente entendidos, modificados e reutilizados;

- **Produtividade do desenvolvedor:** com uma quantidade de código pequena, comparada a outras linguagens, resulta em menos digitação, depuração e manutenção.

- **Portabilidade:** devido a sua implementação padrão ser escrita em ANSI C portátil, permitindo sua compilação e execução em quase todas as plataformas disponíveis e o conjunto de módulos de bibliotecas padrão serem desenvolvidos de forma a serem independentes, ao máximo, de plataforma, programas desenvolvidos em uma plataforma específica podem ser executados em outras diferentes, desde que não se utilizem de recursos dependentes de plataforma;

- **Integração de componentes:** permite a comunicação com outros componentes de outras linguagens ou ferramentas, usando uma variedade de mecanismos de integração;

- **Gerenciamento de memória automática:** a responsabilidade por alocar e desalocar memória para os objetos é realizada por um mecanismo, denominado Garbage Collector, de forma automática;

- **Bibliotecas:** sua distribuição padrão inclui uma grande quantidade de bibliotecas que dão suporte a uma imensa variedade aplicações. Também podem ser instalados, separadamente, módulos adicionais que venham a acrescentar funcionalidades aos módulos da biblioteca padrão, ou que implementem novas classes e procedimentos.

Podemos utilizar Python em diversos tipos de aplicação, tais como: programação de sistemas, programação de GUI⁵, scripts de internet, integração de componentes, programação de banco de dados, programação numérica, jogos, processamento de imagens, programação de IA⁶ e diversas outras. Permite que a produtividade do desenvolvedor seja muitas vezes maior, se comparada com a utilização de linguagens mais tradicionais.

Ela é deliberadamente otimizada para velocidade de desenvolvimento – sua sintaxe simples, de tipagem dinâmica, ausência de etapas de compilação e seu conjunto de ferramentas incorporado, permitem que os programadores desenvolvam programas em uma fração do tempo de desenvolvimento necessário para algumas outras ferramentas. O resultado é que o Python aumenta a produtividade do desenvolvedor muitas vezes além do que se consegue com as linguagens tradicionais. (LUTZ, 2007).

Por permitir o desenvolvimento de aplicações usadas em Mainframes e até em celulares, a Linguagem Python é utilizada por importantes empresas de alta tecnologia como Google, Yahoo, Microsoft, Nokia, Disney, Seagate, IBM, entre outras.

⁵ GUI - Interface Gráfica com o Usuário

⁶ IA - Inteligência Artificial

O Python oferece toda a simplicidade e facilidade de uso de uma linguagem de script, junto com ferramentas de engenharia de software mais avançadas, normalmente encontradas nas linguagens compiladas. Ao contrário de algumas linguagens de script, essa combinação torna o Python útil para projetos de desenvolvimento de grande escala (LUTZ, 2007).

4.2 Compilação e interpretação

Vários componentes são gerados quando o pacote Python é instalado no computador, mas no mínimo um interpretador e a biblioteca de suporte. “Após serem escritos, as instruções dos programas são lidas e executadas pelo interpretador Python, que é uma camada de software lógico entre o código e a máquina” (LUTZ, 2007). Os arquivos que contêm as instruções em linguagem Python, por convenção, devem ser salvos com a extensão.py.

O Python executa algumas etapas ao executar um script, sendo a primeira a compilação do código-fonte para um formato conhecido como código de byte ou bytecode, que é uma representação de nível mais baixo e independente da plataforma do seu código-fonte, isso porque o código de byte pode ser executado muito mais rapidamente do que as instruções originais do código-fonte. O código de byte é uma representação específica do Python (LUTZ, 2007).

O código fonte é traduzido pelo Python para bytecode, que é um formato binário com instruções para o interpretador. O bytecode é multiplataforma e pode ser distribuído e executado sem fonte original (BORGES, 2010). Os bytecodes são armazenados em arquivos com extensão “.pyc”, no computador (bytecode normal) ou “.pyo” (bytecode otimizado), fazendo com que o tempo de carga na execução seja reduzido, pois não haverá necessidade de uma nova compilação do programa numa próxima execução. Automaticamente, o interpretador fará uma nova compilação se os arquivos fontes forem alterados, gerando novos bytecodes. Todo esse processo faz com que programas escritos em Python possam ser mais lentos, ao serem executados, do que se fossem desenvolvidos em linguagem compiladas como C ou C++.

Quando um programa ou um módulo é evocado, o interpretador realiza a análise do código, converte para símbolos, compila (se não houver bytecode

atualizado em disco) e executa na máquina virtual Python (BORGES, 2010). Segundo Lutz (2003), a Máquina Virtual Python (PVM) é o mecanismo de tempo de execução do Python; ela está sempre presente como parte do sistema Python e é o componente que realmente executa seus scripts.

Podemos utilizar o interpretador Python de forma interativa ou através de programas com interfaces mais sofisticadas, como as IDEs (*Vide* Figura 1).

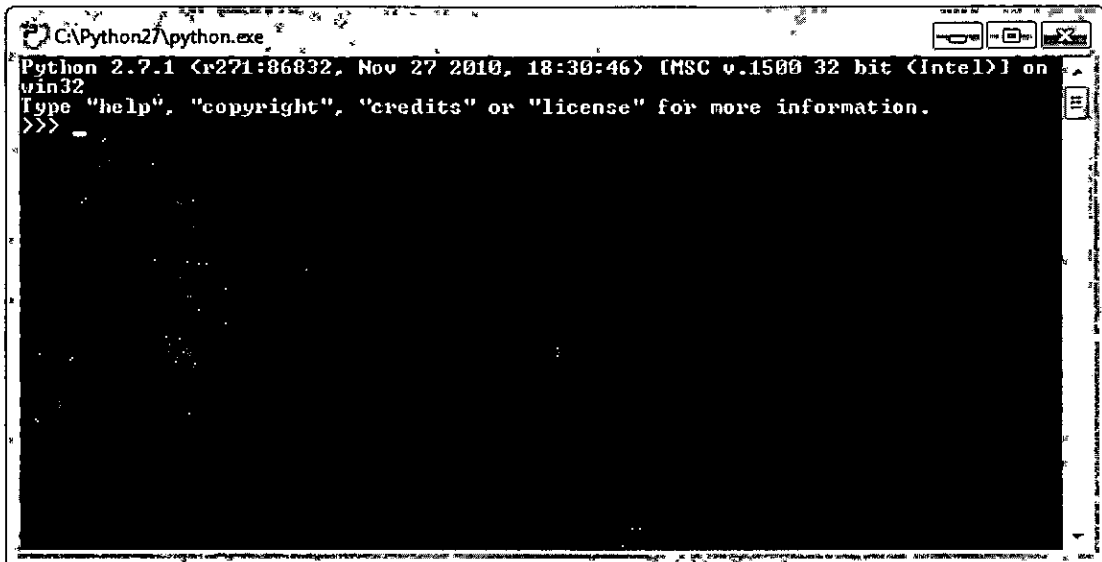


Figura 1 – O Interpretador Python, no modo Interativo.

O modo interativo é uma característica diferencial da linguagem, pois é possível testar e modificar trechos de código antes da inclusão do código em programas, fazer extração e conversão de dados ou mesmo analisar o estado dos objetos que estão em memória, entre outras possibilidades (BORGES, 2010).

5 O FRAMEWORK WEB2PY

Este capítulo tem por objetivo dar uma visão geral do framework Web2py, abordando algumas de suas principais características e funcionalidades, de acordo com as informações contidas na documentação oficial do framework, escrito por seu criador, e artigos sobre o mesmo.

5.1 Visão geral

Web2py é um framework open source disponibilizado sob a licença GPL2 , para desenvolvimento ágil de aplicações web seguras, inspirado no Ruby on Rails e no Django, tendo como principal característica a facilidade de utilização, foi criado pelo italiano Massimo Di Pierro, inicialmente para ser uma ferramenta de ensino, tendo sua primeira versão estável foi lançada em outubro de 2007. Segundo Pierro (2010) o Web2Py é um framework livre e de código aberto para desenvolvimento ágil de aplicações web seguras baseadas em banco de dados e é escrito e programável em Python. Web2py é um framework full-stack, o que significa que web2py contém todos os elementos necessários para construir aplicações web totalmente funcionais.

O Web2py Enterprise Framework é um web framework desenvolvido em Python, que já contém todos os componentes necessários para o desenvolvimento de aplicações web (ROCHA, 2010). De acordo com Pierro (2010) o Web2py é projetado para guiar o desenvolvedor a seguir as boas práticas de engenharia de software, tais como usar o padrão projeto MVC (Model View Controller), e assim permite organizar o código, incentivando a reutilização de código. Também fornece, ao desenvolvedor, várias bibliotecas para utilizar na implementação e teste de cada uma das camadas da aplicação.

Web2Py difere de outros frameworks web, uma vez que não requer instalação ou configuração, ele roda em qualquer arquitetura que suporte Python (Windows, Windows CE, Mac OS X, IOS e Unix/Linux), e para o desenvolvimento, implantação e as fases de manutenção para as aplicações podem ser feitas localmente através de uma interface web local ou remota (PIERRO, 2010).

O Web2py é um framework inovador para desenvolvimento de aplicações web escrito em Python, cujo principal foco é a agilidade no desenvolvimento de aplicações baseadas em banco de dados, seguras, escaláveis e portáteis. Por ser escrito totalmente em Python, ele é multiplataforma: roda em GNU/Linux, Mac OS, Windows, JVM (Jython), Google App Engine (nativamente), entre outros (JUSTEN, 2010).

Segundo Pierro (2010) o Web2py tem como objetivo dar condições para que o desenvolvedor preocupe-se apenas na aplicação a ser desenvolvida, pois não necessita ser instalado e não exige configurações complicadas, bastando baixar o framework, descompactar e começar a desenvolver. Também permite se conectar e lidar de forma transparente com 10 sistemas de banco de dados (SQLite, PostgreSQL, MySQL, MSSQL, FireBird, Oracle, IBM DB2, Informix, Ingres e Google App Engine (GAE), mas já vem com um servidor web próprio (Rocket) e com os pacotes necessários para utilização do SQLite e dispõe de subsistemas de criação automática de formulários com validação automática, autenticação e autorização, gerador de códigos AJAX para melhor interação do usuário com a aplicação, upload seguro de arquivos, integração com vários padrões Web, dentre outros. Ele também permite compatibilidade entre as versões do framework sem que seja necessário nenhuma alteração de código.

O Web2py foi projetado para ser seguro, considerando questões de segurança de aplicações web, com preocupação de tratar automaticamente de problemas que permitam vulnerabilidades através da aplicação de práticas bem estabelecidas, como a validação de formulários evitando a injeção de códigos maliciosos efetuando o correto escape da saída HTML para prevenir ameaças como o cross-site scripting e renomeando os arquivos de upload utilizando hash seguro.

A camada de abstração de banco de dados (DAL) cria de forma dinâmica todo o script SQL de acordo com o banco em uso, não necessitando conhecimento em SQL para esse determinado banco de dados. Para que seja possível a conexão com cada tipo de banco de dados, é necessário ter instalado o driver correspondente. Para Rocha (2010a) ela é uma API que mapeia objetos Python em objetos de banco de dados como queries, tabelas e registros. A DAL gera códigos SQL dinamicamente, em tempo real, utilizando sempre o dialeto SQL referente ao

banco de dados em uso. Dessa forma, você não precisa escrever código SQL ou aprender comandos SQL de um banco de dados específico, e sua aplicação será portátil para diferentes bancos de dados.

O web2py permite conexão com mais de um banco de dados simultaneamente em uma mesma aplicação, gerando uma interface administrativa que permite administrar os dados, dispensando a utilização de ferramentas para administrá-los.

Segundo Pierro (2010), o Web2py é composto pelos seguintes componentes: bibliotecas, servidor web, aplicação admin, aplicação examples e aplicação welcome. O primeiro são as bibliotecas que podem ser acessadas durante o desenvolvimento e fornecem as funcionalidades do Web2py. O segundo é o servidor web denominado Rocket, que vem embutido no framework e pode ser utilizado no desenvolvimento de aplicações, mas também podem ser utilizados outros como o Apache, o Lighttpd, o Cherokee ou qualquer outro que suporte WSGI, CGI/FastCGI ou atue como proxy.. O terceiro é uma aplicação utilizada para administrar as aplicações instaladas. O quarto contém exemplos de código; e o quinto é uma aplicação modelo.

Segundo Pierro (2010), o Web2py pode ser executado com CPython ou Jython e funciona com a maioria dos seus recursos na plataforma gratuita de hospedagem de aplicações do Google, o GAE (Google App Engine). Ele também afirma que aplicações desenvolvidas com o Web2py possam ser licenciadas como open source ou software comercial, não estando sujeitas ao tipo de licença sob a qual é distribuído (GPL2). Também possibilita que as aplicações sejam compiladas em byte-code para que sejam distribuídas com código fechado.

5.2 Principais características

De acordo com Pierro (2010), o Web2py é um framework que torna o desenvolvimento web mais fácil, rápido e seguro, que possui características muito atraentes, sendo algumas delas:

- É multiplataforma, podendo rodar em Linux, Mac, Windows, Windows Mobile, e também no Google App Engine e Amazon, sua única dependência é o Python.
- Não necessita de instalação, é só baixar o arquivo descompacta e usar, possibilitando desenvolver utilizando qualquer dispositivo USB.
- Não há arquivos de configuração.
- Possui o um servidor web integrado, mas também suporta Apache e outros.
- Inclui um sistema de banco de dados relacionais, o SQLite.
- Inclui sistemas de tickets para auxiliar no Debug das aplicações.
- Dispõe de uma sofisticada camada de abstração de banco de dados (DAL – Database Abstraction Layer), automatizando códigos SQL em tempo de execução para vários gerenciadores de banco de dados (SQLite , PostgreSQL , MySQL , MSSQL , FireBird , Oracle, Google Big Table).
- Gera automaticamente uma interface de administração de banco de dados para as aplicações.
- Previne as vulnerabilidades mais comuns, como Cross Site Scripting, Injeção de Falhas e execução de arquivos suspeitos.
- Incentiva as boas práticas da engenharia de software, como o uso do padrão de projeto MVC (Model-View-Controller),
- Inclui bibliotecas de Autenticação (Login, Logout, registros, alteração de senhas, dentre outras)
- Bibliotecas de Autorização (Controle de acessos).
- Bibliotecas para manipulação de HTML , ATOM , RSS , RTF , JSON , AJAX , XML-RPC , CSV , REST e WIKI.
- Trabalha com Flash/Flex via PyAMF e com os Toolkits Web do Google através do Pyjamas.
- Inclui JQuery para efeitos e AJAX.
- Possui uma poderosa linguagem de templates baseada no Python que permite renderizar objetos para HTML ou qualquer outro protocolo baseado em textos.
- Manipula streaming automaticamente de upload e download de grandes mídias.

5.3 O funcionamento do Framework Web2py

Para o desenvolvimento de aplicações necessitamos inicialmente da versão do Framework Web2py apropriada ao sistema operacional em que será utilizado que pode ser obtido no site oficial <http://www.web2py.com>. Pode ser que seja necessário instalar a biblioteca python-tk no Ubuntu utilizando o seguinte comando: `sudo apt-get install python-tk`. Depois de baixada a versão necessária descompacte o arquivo e dentro do diretório web2py execute o comando "python web2py.py" através do terminal se for plataforma Linux, se for Windows basta executar o arquivo "web2py.exe". Em seguida será exibida a seguinte tela:

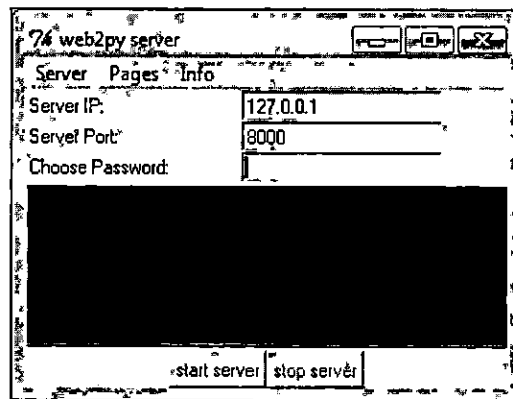


Figura 2 - Interface para gerenciamento do servidor.

Nesta tela deverá ser informada uma senha para que se tenha acesso a aplicação de administração, senha do administrador, sendo necessária informar sempre uma nova senha ao iniciar o Web2py, caso contrário a aplicação de administração será desabilitada, tudo como medida de segurança. Depois de ser definida a senha de administração seu navegador será aberto automaticamente e direcionado para <http://127.0.0.1:8080/>, que é a página da aplicação Welcome (Figura 3), que é a aplicação que serve como modelo básico para outras a serem desenvolvidas. Podemos observar também a aplicação Admin (Interface administrativa) que é utilizada para administrar as aplicações instaladas, e também a aplicação Examples (Alguns exemplos), com a documentação interativa online e uma réplica do site oficial do Web2py.

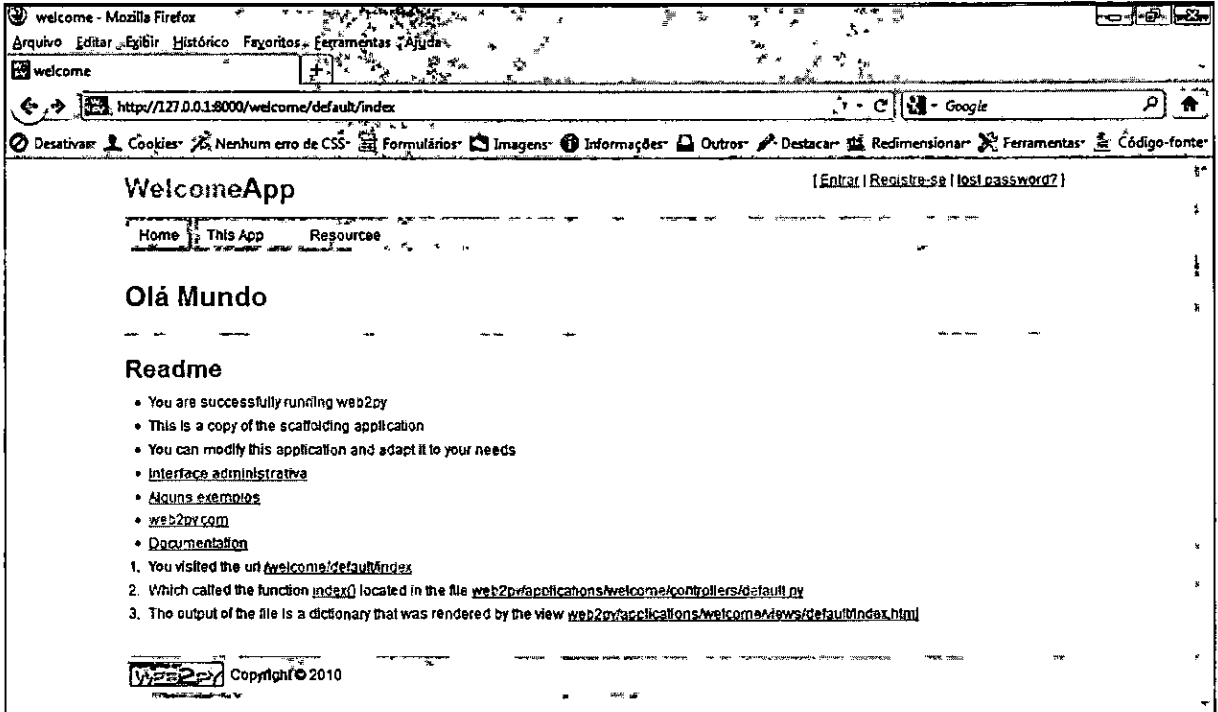


Figura 3 - Página da aplicação Welcome.

Clicando em Interface administrativa (aplicação Admin) será feito um redirecionamento para uma página de login (Figura 4), nela deverá ser informada a mesma senha informada na inicialização, em seguida será redirecionada para a página site (Figura 5), onde são listados todos os aplicativos instalados.

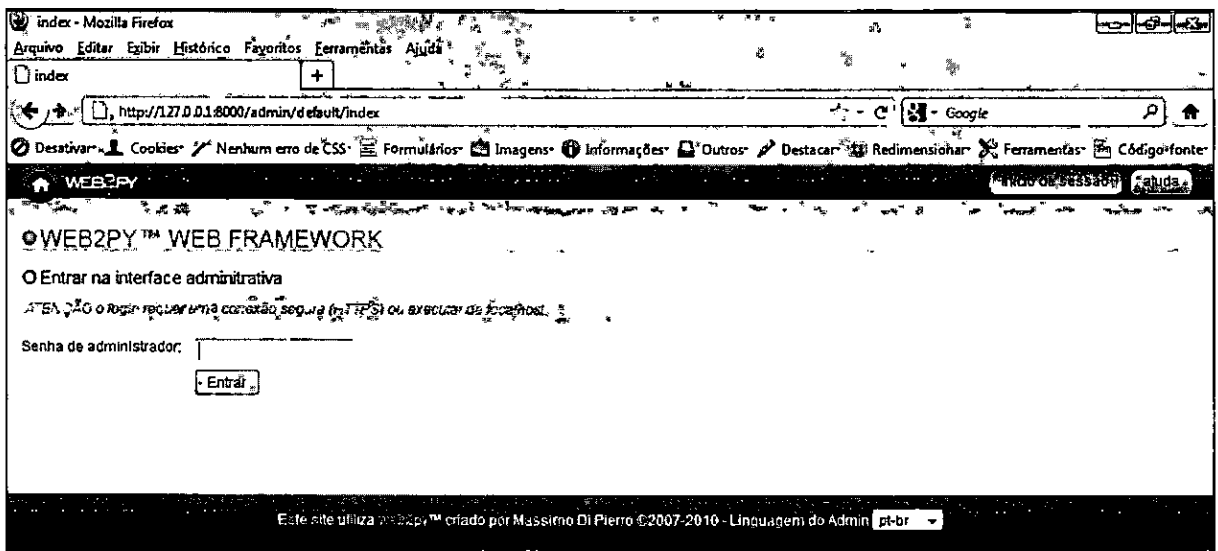


Figura 4 - Página de login da aplicação Admin.

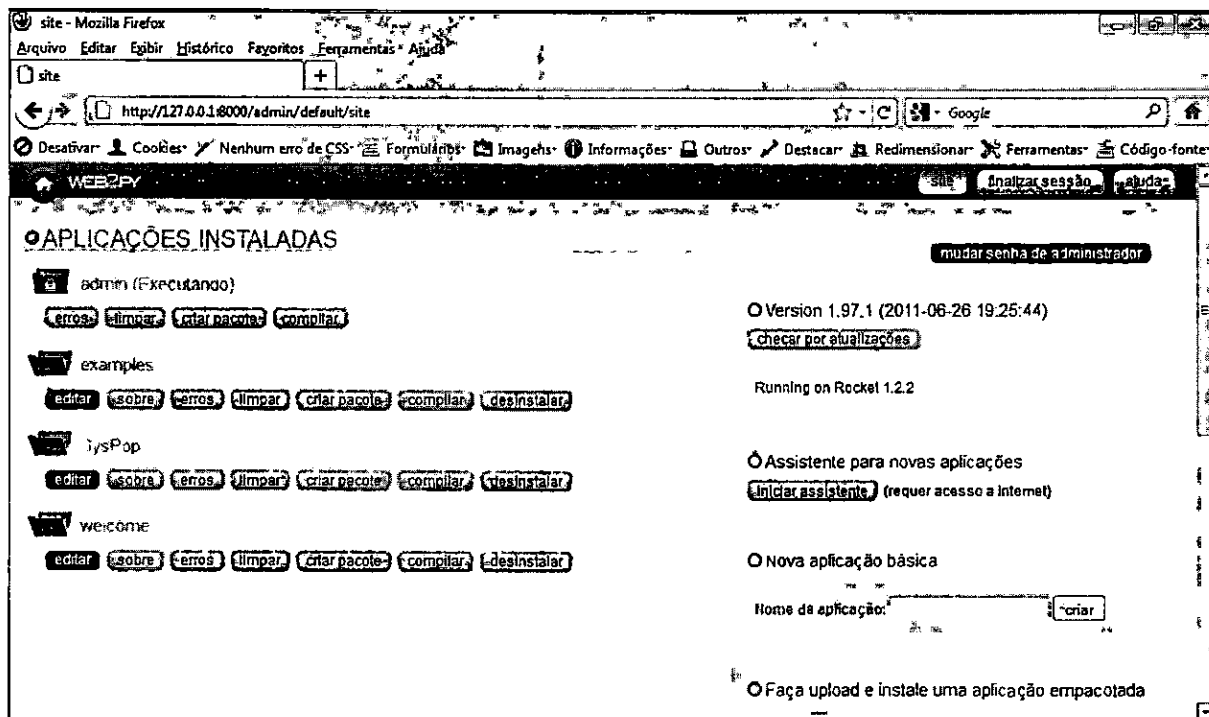


Figura 5 - Página da Site.

A interface administrativa é uma aplicação com poderes para administrar os arquivos e bancos de dados das aplicações instaladas através das seguintes opções:

- **Criar:** cria uma nova aplicação com base na estrutura da aplicação Welcome.
- **Instalar:** instalar uma nova aplicação. Também podemos instalar aplicações informando o caminho para um pacote w2p ou através da sua URL. Em <http://www.web2py.com/appliances> encontram-se disponíveis várias aplicações w2p totalmente funcionais.
- **Desinstalar:** remove uma instalação.
- **Criar pacote:** permite empacotar uma aplicação instalada, sendo gerado um arquivo com formato w2p para que possa ser distribuído.
- **Compilar:** compila uma aplicação para a distribuição binária sem o código fonte.
- **Erros:** exhibe os tickets, de erros gerados pela aplicação
- **Limpar:** limpa todos os arquivos temporários, sessões, tickets de erro e arquivos de cache de uma aplicação.
- **Editar:** permite abrir a aplicação em modo de edição.

Quando uma aplicação é criada é exibida uma tela exibindo toda a estrutura da aplicação, que consiste em arquivos distribuídos em categorias que correspondem a pastas no diretório de aplicações (web2py\applications\): (Vide figura 6)

- **modelos:** é o local onde definimos e gerenciamos os modelos de dados da aplicação, efetuamos conexões com banco de dados e definimos a modelagem das tabelas, constantes, variáveis e configurações de acesso global.

- **controladores:** são as ações da aplicação. Aqui definimos as regras de negócio e as validações de tempo de execução; o controlador é quem recebe a entrada de dados, invoca os objetos do modelo de dados, efetua as validações e envia como resposta uma visão.

- **visões:** são as visões que servirão para apresentar os dados dos modelos, invocados e tratados pelos controladores. São criadas a partir de templates que podem responder conteúdo no formato HTML, RSS, XML e JSON, entre outros.

- **linguagens:** local onde definimos arquivos de linguagem que permitem a internacionalização das aplicações.

- **estáticos:** neste diretório, inserimos arquivos que não necessitam de processamento, como estruturas estáticas de layout, imagens, arquivos de estilo CSS e JavaScript.

- **módulos:** aqui colocaremos bibliotecas e módulos Python que não fazem parte do web2py, mas que podem ser importados e utilizados nas aplicações.

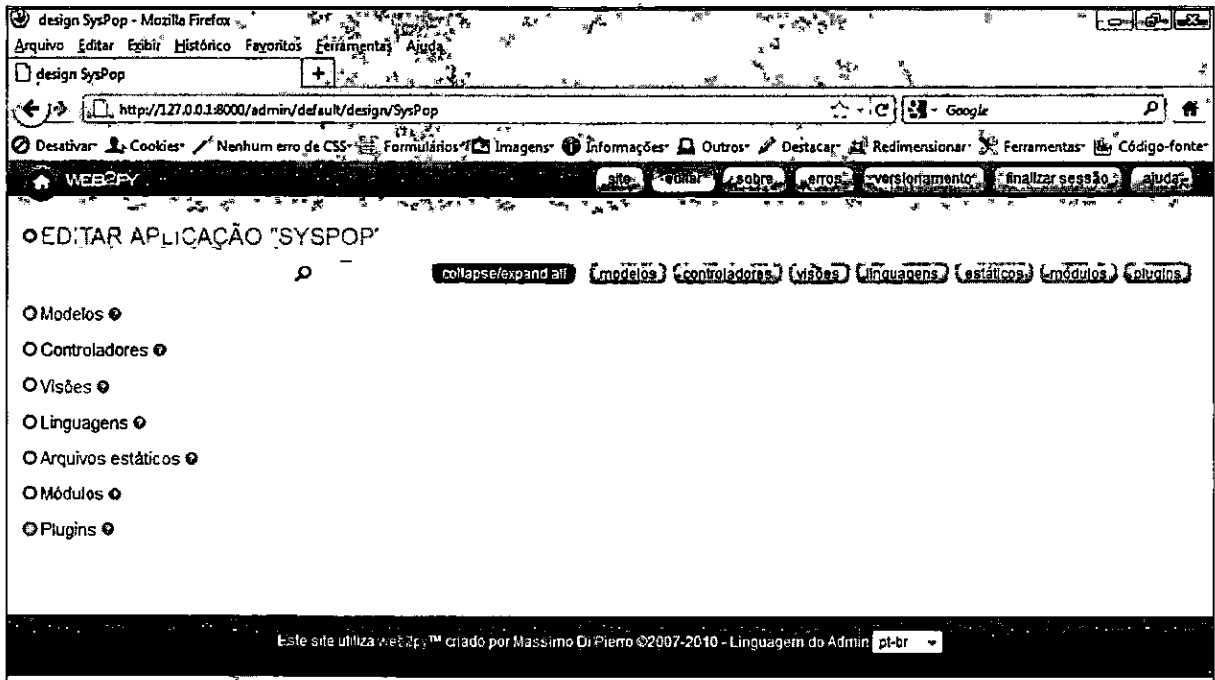


Figura 6 - Tela de edição de aplicações.

As aplicações ainda possuem outros tipos de arquivo (arquivos de sessão, tickets de erro, dentre outros), mas estes arquivos não são listados na página de edição, pois não são criados ou gerenciados pelo desenvolvedor: eles são criados e modificados pela própria aplicação.

As pastas e os arquivos exibidos na interface administrativa correspondem, respectivamente, às pastas e arquivos localizados no diretório da aplicação (`web2py\applications\`), conforme demonstrado na figura 7, a seguir:

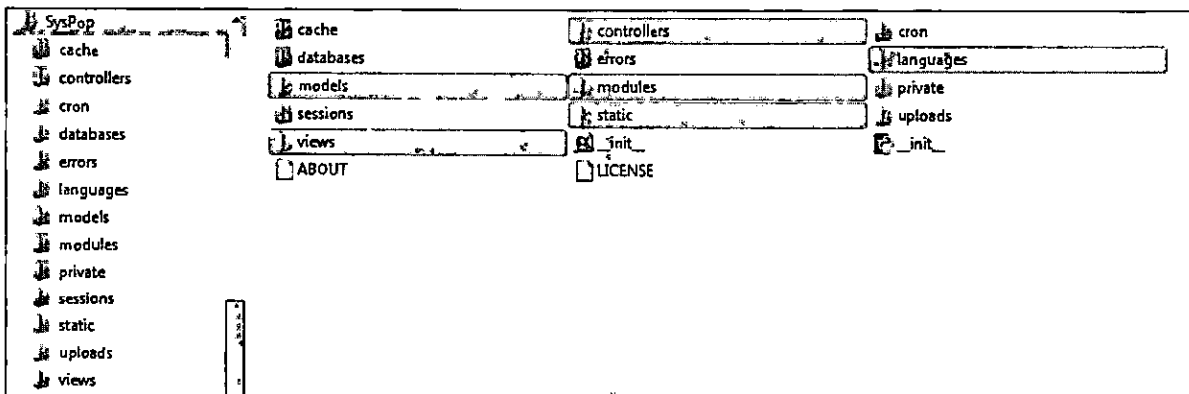


Figura 7 - Diretório de uma aplicação.

Qualquer operação realizada na interface administrativa pode ser realizada através de comandos de terminal, e os arquivos podem ser editados em qualquer editor de textos ou IDE de sua preferência.

5.4 Exemplos de códigos

Serão exibidos a seguir exemplos de códigos que demonstram o poder e simplicidade do Web2py:

- **Exemplo 1**, definição da tabela pessoa com os atributos nome e sobrenome:

```
db.define_table('pessoa', Field('nome'), Field('sobrenome'))
```

- **Exemplo 2**, conexão com o banco de dados:

```
db = DAL('meu_banco://minha_base_de_dados.db')
```

- **Exemplo 3**, validação de email:

```
db.define_table('minha_tabela', SQLField('email'))  
db.mihha_tabela.email.requires=IS_EMAIL()
```

6 ANÁLISE E IMPLEMENTAÇÃO DO SISTEMA SYSPOP

Este capítulo trata do desenvolvimento da aplicação SYSPOP que o objetivo principal deste trabalho. Inicialmente são feitas algumas considerações, em seguida é feita uma descrição da modelagem do sistema através dos diagramas de casos de uso, de classes e de seqüência.

6.1 Considerações iniciais

O contexto geral deste trabalho é o desenvolvimento de uma aplicação web, denominada SYSPOP, para gerenciar pesquisas ou entrevistas com questões fechadas (objetivas), de forma a facilitar o tratamento e análise da informação, permitindo economizar tempo. De acordo com Tanaka e Melo (2001), este tipo de questão tem sua utilização em situações que se conhece a abrangência e os tópicos mais mencionados como resposta, permitindo a um informante optar por uma das respostas oferecidas, facilitando sua resposta sobre determinado assunto. Quando não se conhece variabilidade das respostas de um determinado item, faz-se necessário o uso de questões abertas que dão uma maior liberdade ao respondente.

No modelo proposto para o SYSPOP, o usuário será inicialmente levado a uma tela de identificação (login), onde serão informados os dados de sua conta de usuário necessários para seu acesso as funcionalidades do sistema, de acordo com seus privilégios. Após a identificação do usuário poderão ser exibidas, dentre outras, opções de manipulação de usuários da aplicação, das pesquisas e de relatórios.

O SYSPOP deve atender alguns requisitos não funcionais: ser desenvolvido para ser utilizado em ambiente web e oferecer segurança através do controle dos usuários por meio de autenticação baseada em login e senha. O SYSPOP deve ser capaz de suprir os seguintes requisitos funcionais:

- **Cadastro de usuário:** deve permitir o cadastro de usuário mediante o fornecimento de informações que permitam sua posterior identificação.
- **Cadastro de tipos de usuários:** deve permitir o cadastro de tipos de usuário, definindo para cada tipo privilégios para interagir com a aplicação.

- **Cadastro de pesquisas:** deve permitir o cadastro de pesquisas pelo usuário do sistema, permitindo a utilização de diferentes pesquisas.
- **Cadastro de questão:** deve permitir o cadastro de questões, permitindo que as mesmas sejam associadas a uma pesquisa.
- **Cadastro de alternativa de questão:** deve permitir o cadastro de alternativas (opções), sendo estas associadas a uma questão.
- **Cadastro de entrevistados:** deve permitir o cadastro das pessoas a serem entrevistadas.
- **Gerar relatórios:** deve permitir que o administrador da aplicação gere relatórios sobre as pesquisas, pesquisadores, entrevistados e usuários.

No desenvolvimento da aplicação web SYSPOP foram utilizados: o Framework Web2py, a linguagem de programação Python e a base de dados em SQLite.

6.2 A modelagem

A modelagem de um sistema permite o seu entendimento, sendo assim, utilizamos a UML (Unified Modeling Language), que é uma linguagem (no sentido de permitir a comunicação) para visualização, especificação, construção e documentação de um sistema de informações para a modelarmos o SYSPOP.

A seguir serão apresentados os diagramas de caso de uso, de classe e de sequência.

6.3 Diagrama de Caso de Uso do SYSPOP

Um caso de uso descreve os procedimentos (bem como suas variações conforme certas condições) de um sistema, subsistema, ou mesmo de apenas um procedimento, sem especificar como eles são implementados (LARMAN, 2000) (BOOCH; RUMBAUGH; JACOBSON, 1999).

No diagrama de caso de uso geral podemos ter uma visão geral das ações dos usuários (administrador e pesquisador) dentro aplicação, demonstrando que os mesmos necessitam de autenticação para que possam interagir com a aplicação. Vejamos o diagrama na figura 8, a seguir:

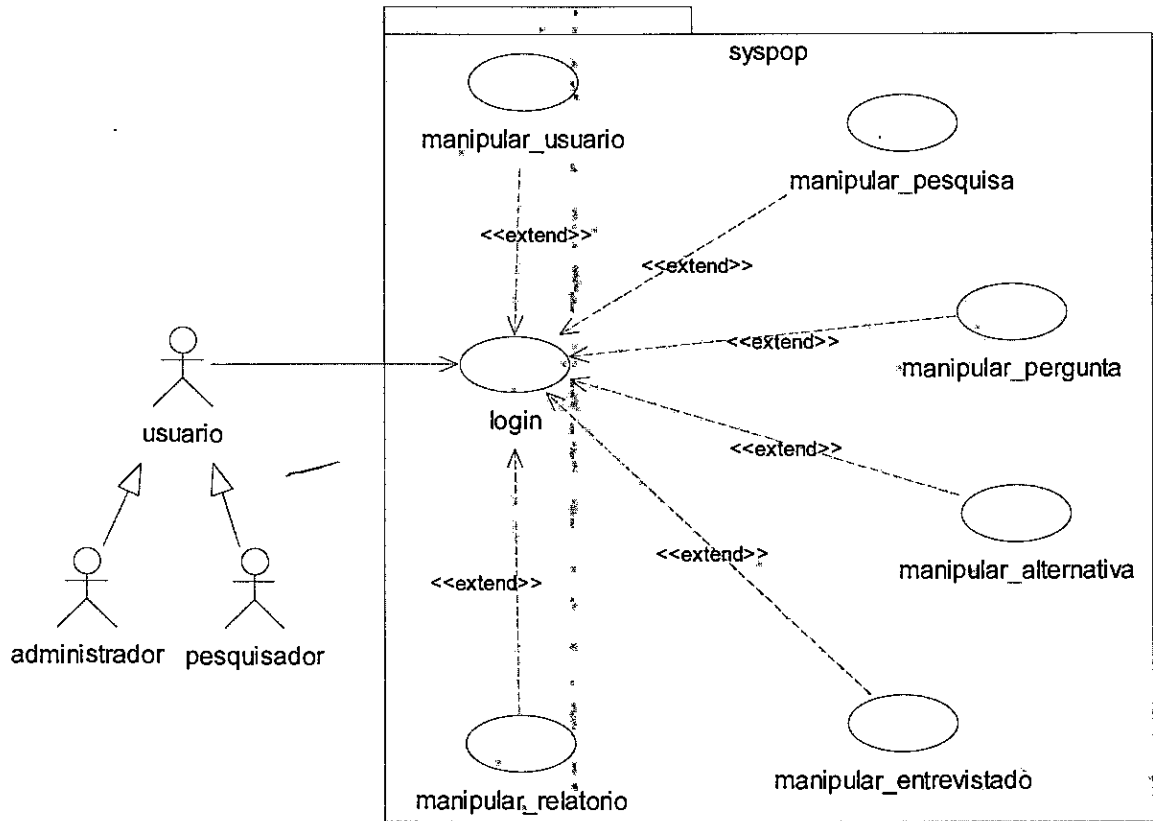


Figura 8 - Diagrama de Caso de Uso "Geral".

A seguir, estão expostos os atores e casos de usos que fazem parte do diagrama de Caso de Uso "Geral" da figura 8.

Documentação do Caso de Uso Manipular Usuários.

Nome do Caso de Uso	Manipular Usuários
Ator Principal	Administrador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação dos usuários da aplicação.
Pré-Condições	O usuário, necessita ter privilégios de administrador da aplicação.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	
	2) Autenticar e avaliar privilégios do usuário
3) Realizar ações (criar, editar, pesquisar e excluir) sobre os usuários	
	4) Analisar dados para efetuar atualização do

	sistema
	5) Gravar os dados atualizados
Restrições / Validações	<p>1) Para que sejam realizadas as ações referentes a manipulação dos usuários deve ser observado a existência ou não dos mesmos na base de dados</p> <p>2) Apenas usuário com privilégios de administrador da aplicação poderá realizar ações sobre os usuários.</p> <p>3) Antes de serem gravadas na base de dados as informações devem ser validadas.</p>

Documentação do Caso de Uso Manipular Pesquisas.

Nome do Caso de Uso	Manipular Pesquisas
Ator Principal	Administrador
Ator Secundário	Pesquisador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação das pesquisas.
Pré-Condições	O usuário necessita ter privilégios especiais para efetuar algumas ações.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	
	2) Autenticar e avaliar privilégios do usuário
3) Realizar ações (criar, editar, pesquisar e excluir) sobre as pesquisas	
	4) Analisar dados para efetuar atualização do sistema
	5) Gravar os dados atualizados
Restrições / Validações	<p>1) Para que sejam realizadas as ações referentes a manipulação das pesquisas, devem ser observados a existência ou não das mesmas.</p> <p>2) As ações de criar e excluir pesquisas serão executadas apenas por usuários com privilégios de administrador.</p> <p>3) As informações devem ser validadas antes de serem gravadas na base de dados.</p>

Documentação do Caso de Uso Manipular Pergunta.

Nome do Caso de Uso	Manipular Pergunta
Ator Principal	Administrador
Ator Secundário	Pesquisador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação das perguntas.
Pré-Condições	O usuário necessita ter privilégios especiais para efetuar algumas ações.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	
	2) Autenticar e avaliar privilégios do usuário
3) Realizar ações (criar, editar, pesquisar e excluir) sobre as perguntas	
	4) Analisar dados para efetuar atualização do sistema
	5) Gravar os dados atualizados
Restrições / Validações	<p>1) Para que sejam realizadas as ações referentes a manipulação das perguntas, devem ser observados a existência ou não das mesmas.</p> <p>2) As ações de criar e excluir perguntas serão executadas apenas por usuários com privilégios de administrador.</p> <p>3) As informações devem ser validadas antes de serem gravadas na base de dados.</p>

Documentação do Caso de Uso Manipular Alternativa.

Nome do Caso de Uso	Manipular Alternativa
Ator Principal	Administrador
Ator Secundário	Pesquisador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação das alternativas.
Pré-Condições	O usuário necessita ter privilégios especiais para efetuar algumas ações.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	

	2) Autenticar e avaliar privilégios do usuário
3) Realizar ações (criar, editar, pesquisar e excluir) sobre as alternativas.	
	4) Analisar dados para efetuar atualização do sistema
	5) Gravar os dados atualizados
Restrições / Validações	<p>1) Para que sejam realizadas as ações referentes a manipulação das alternativas, devem ser observados a existência ou não das mesmas.</p> <p>2) As ações de criar e excluir alternativas serão executadas apenas por usuários com privilégios de administrador.</p> <p>3) As informações devem ser validadas antes de serem gravadas na base de dados.</p>

Documentação do Caso de Uso Manipular Entrevistados.

Nome do Caso de Uso	Manipular Entrevistados
Ator Principal	Administrador
Ator Secundário	Pesquisador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação dos entrevistados.
Pré-Condições	O usuário necessita ter privilégios especiais para efetuar algumas ações.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	
	2) Avaliar privilégios do usuário
3) Realizar ações (criar, editar, pesquisar e excluir) sobre os entrevistados	
	4) Analisar dados para efetuar atualização do sistema
	5) Grava os dados atualizados
Restrições / Validações	<p>1) Para que sejam realizadas as ações referentes a manipulação dos entrevistados, devem ser observados a existência ou não dos mesmos.</p> <p>2) Apenas usuários com privilégios de administrador podem excluir os entrevistados cadastrados.</p>

	3) As informações devem ser validadas antes de serem gravadas na base de dados.
--	---

Documentação do Caso de Uso Manipular Relatórios.

Nome do Caso de Uso	Manipular Relatórios.
Ator Principal	Administrador
Ator Secundário	Pesquisador
Resumo	Este Caso de Uso informa como será executado o processo de manipulação de relatórios.
Pré-Condições	O usuário necessita ter privilégios especiais para efetuar algumas ações.
Ações do Ator	Ações do Sistema
1) Solicitar acesso ao sistema	
	2) Avaliar privilégios do usuário
3) Realizar ações (gerar, pesquisar e excluir) sobre os relatórios.	
	4) Analisar dados para efetuar atualização do sistema
	5) Grava os dados atualizados
Restrições / Validações	1) Para que sejam realizadas as ações referentes a manipulação das pesquisas, devem ser observados a existência ou não das mesmas. 2) Somente usuários com privilégios de administrador podem gerar ou excluir relatórios.

6.4 Diagrama de classes

O diagrama de classe mostra graficamente como as classes e interfaces se relacionam, especificando também as propriedades/constantes de cada classe/interface e quais são os seus métodos oferecidos/a implementar respectivamente (LARMAN, 2000; BOOCH; RUMBAUGH; JACOBSON, 1999; FURLAN, 1998).

A figura 9 apresenta o diagrama de classes do SYSPOP:

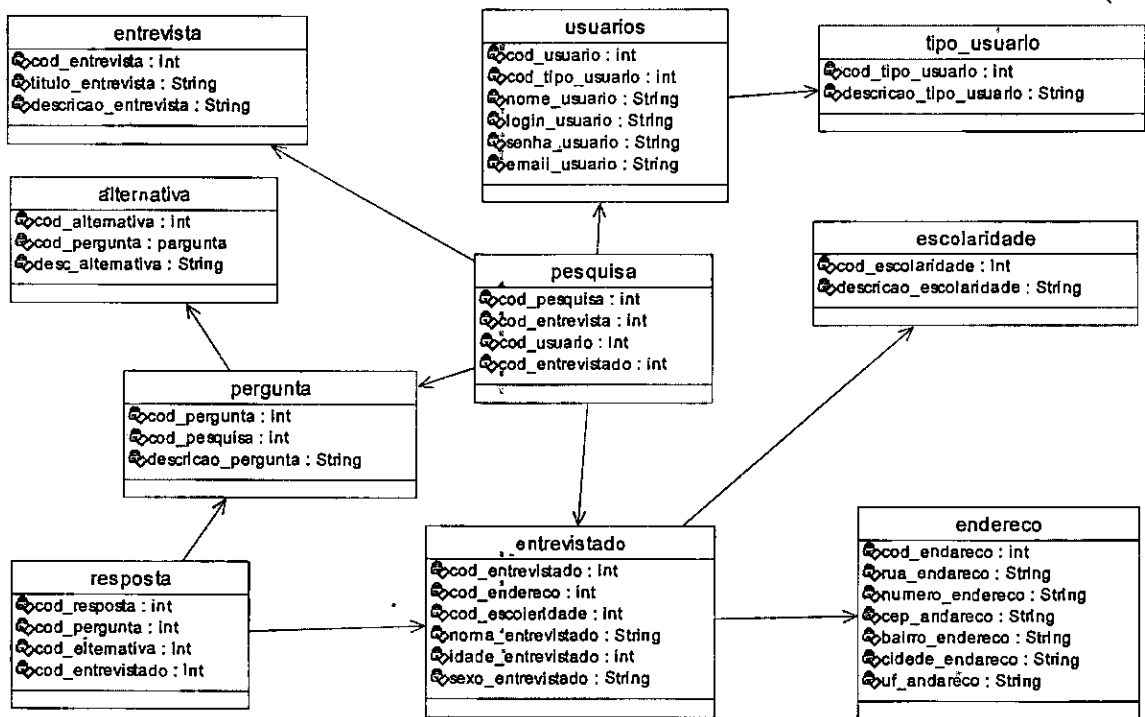


Figura 9 - Diagrama de Classes.

6.5 Interfaces do ambiente

A aplicação SYSPOP tem como objetivo final auxiliar no processamento dos dados de uma pesquisa de opinião de questionário fechado, dando mais confiabilidade e velocidade. Neste protótipo (v.1-2011) desenvolvido é possível fazer o cadastro dos usuários do sistema e definir os privilégios dos mesmos, realizar o cadastro dos eventuais entrevistados, cadastrar as entrevistas e também as pesquisas.

Para que um usuário tenha acesso aos recursos da aplicação será necessário que este seja autenticado, através de seu nome de usuário e sua senha. (Vide figura 11)

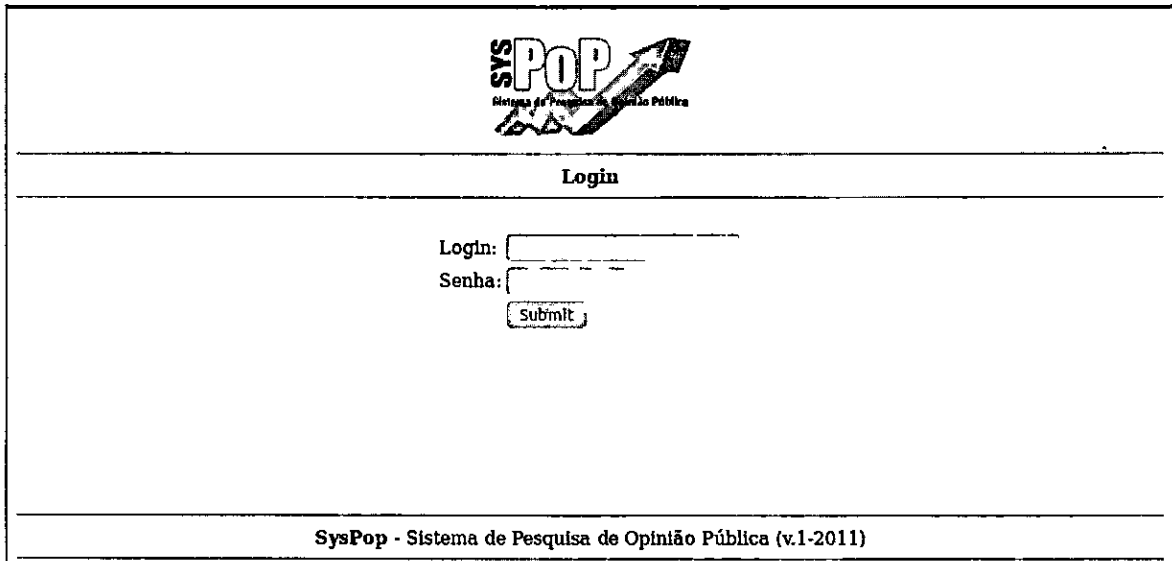


Figura 10 - Tela de login do SYSPOP

Em seguida, com o usuário autenticado, será apresentada uma tela denominada Painel de controle (Vide figura 12) onde o mesmo terá acesso as funcionalidades da aplicação de acordo com seus privilégios.

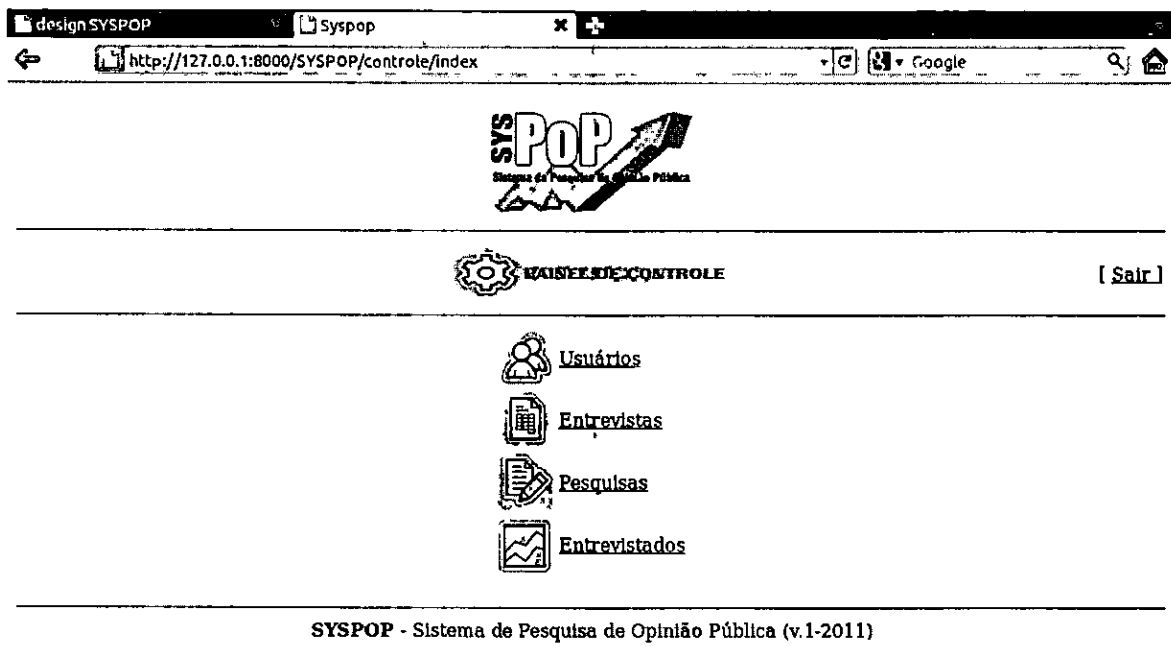


Figura 11 - Tela de Painel de controle do SYSPOP

A página de cadastro de usuários é apresentada na figura 13. Nesta página serão informados dados básicos para que seja necessário realizar o cadastro do usuário da aplicação, bem como definir o seu tipo (perfil), no qual estão definidos seus privilégios.

design SYSPOP SYSPOP Syspop

http://127.0.0.1:8000/SYSPOP/controle/criar_usuario

SYSPOP
Sistema de Pesquisa de Opinião Pública

Cadastrar usuário [Sair]

First name:

Last name:

Username:

E-mail:

Password:

Cadastrar

Voltar para [Painel de controle] [Usuários cadastrados]

SYSPOP - Sistema de Pesquisa de Opinião Pública (v.1-2011)

Figura 12 - Tela de cadastro de usuários do SYSPOP

Já na página de cadastro de entrevistas serão informados o título, descrição, sendo esta o tipo das pesquisas a serem realizadas.

design SYSPOP SYSPOP Syspop

http://127.0.0.1:8000/SYSPOP/controle/criar_entrevista

SYSPOP
Sistema de Pesquisa de Opinião Pública

Cadastrar entrevista [Sair]

Titulo:

Descricao:

Cadastrar

Voltar para [Painel de controle] [Entrevistas cadastradas]

SYSPOP - Sistema de Pesquisa de Opinião Pública (v.1-2011)

Figura 13 - Tela de cadastro de entrevistas do SYSPOP

Já na página de cadastro de entrevistados serão informados os dados das pessoas entrevistadas nas pesquisas. Serão cadastrados dados como: nome, sexo, idade, grau de escolaridade, rua/avenida, número da residência, bairro, CEP, cidade, uf. Vejamos na figura 15, a seguir:

design SYSPOP SYSPOP Syspop

http://127.0.0.1:8000/SYSPOP/controle/criar_entrevistado

Google

SYSPOP
Sistema de Pesquisa de Opinião Pública

Cadastrar entrevistado [Sair]

[Nova escolaridade] [Novo endereco]

Nome:

Idade:

Sexo:

Desc Escolaridade:

Desc Endereco:

Voltar para [[Painel de controle](#)] [[Entrevistados cadastrados](#)]

SYSPOP - Sistema de Pesquisa de Opinião Pública (v.1-2011)

Figura 14 - Tela de cadastro de entrevistados do SYSPOP

Na página de cadastro de pesquisas (Vide figura 16) serão informados o título da entrevista, o nome do entrevistado e o nome do entrevistador. Dados necessários para se definir os participantes da pesquisa.

design SYSPOP SYSPOP Syspop

http://127.0.0.1:8000/SYSPOP/controle/criar_pesquisa

Google

SYSPOP
Sistema de Pesquisa de Opinião Pública

Cadastrar pesquisa [Sair]

Cod Entrevista:

Cod Usuario:

Cod Entrevistado:

Voltar para [[Painel de controle](#)] [[Pesquisas cadastradas](#)]

SYSPOP - Sistema de Pesquisa de Opinião Pública (v.1-2011)

Figura 15 - Tela de cadastro de pesquisas do SYSPOP

7 CONCLUSÃO

O presente trabalho apresenta em seu contexto geral a modelagem de uma aplicação web para auxiliar o gerenciamento de pesquisas de opinião pública na forma de questionário fechado, o SYSPOP; e também procuramos demonstrar as principais funcionalidades do framework Web2py na implementação de um protótipo (v.1-2011) da aplicação. Neste protótipo foram implementadas as funções de cadastro de usuários, entrevistas, entrevistados e pesquisas.

7.1 Trabalhos futuros

Para as próximas versões devem ser implementadas melhorias como autenticação dos usuários do sistema, geração de diversos tipos de relatórios, realização de pesquisas online, dentre outras.

8 REFERÊNCIAS

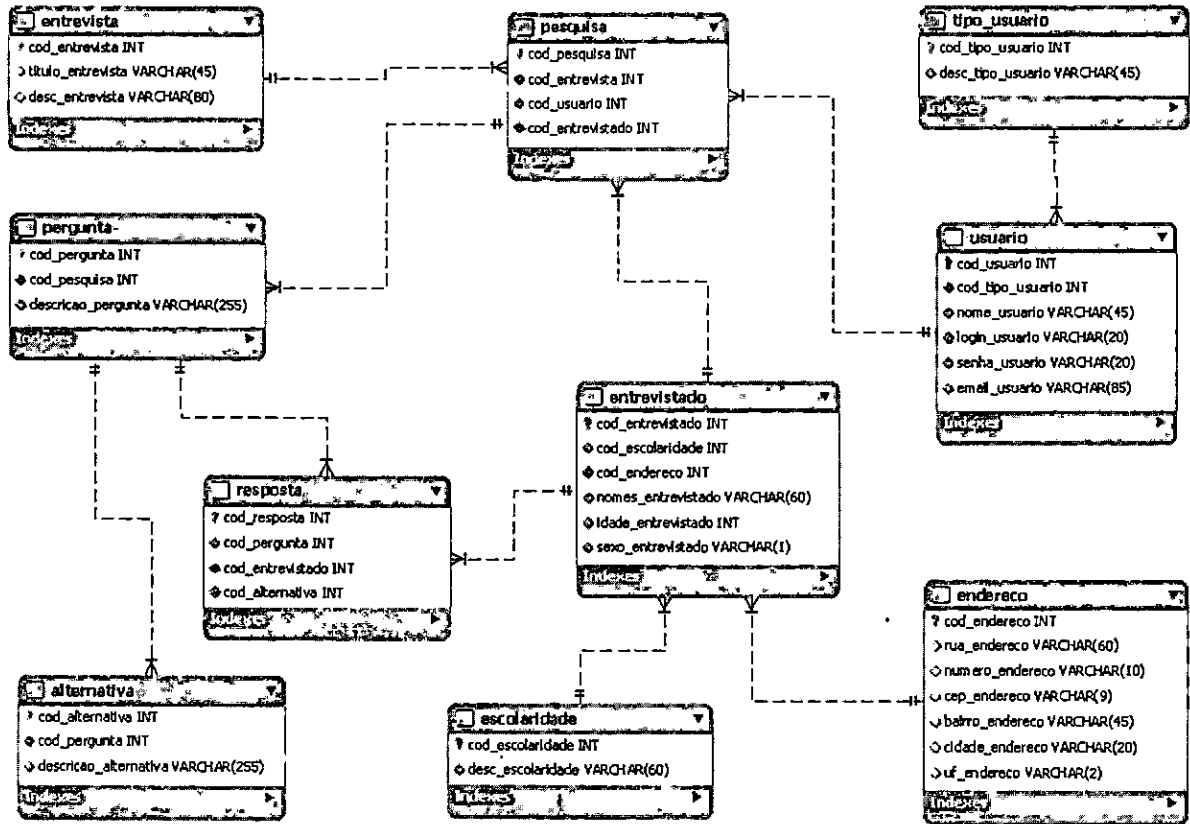
- [1] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. Reading: Addison Wesley, 1999.
- [2] BORGES, Luiz Eduardo. **Python para Desenvolvedores**. 2 ed. Rio de Janeiro: Edição do Autor, 2010.
- [3] CHILDS, Harwood L. Relações públicas, propaganda e opinião pública: **Que é opinião pública?** 2. ed. Rio de Janeiro: FGV, 1967. p. 44-61.
- [4] DYSON, Peter. **Dominando Internet Information Server: Internet/Intranet - A Bíblia**. Tradução Lavio Pareschi. São Paulo: Makron Books, 1998.
- [5] ECHEGARAY, Fabián. Opinião Pública: **O papel das pesquisas de opinião pública na consolidação da democracia: a experiência latino-americana**. vol. 7 no.1 Campinas, 2001.
- [6] FERRARI, Marlon. **A importância da Linguagem Python no cenário empresarial**. Revista Espírito Livre. Edição 017. Agosto, 2010. Disponível em <http://revista.espiritolivre.org>. Acessado em julho de 2011.
- [7] FRANCE, R.; KOBRYN, C. UML for Software Engineer, In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 23., 2001, Toronto. **Proceedings ...** Toronto: IEEE Computer Society Press, 2001. p. 705-706.
- [8] FURLAN, J. D. **Modelagem de Objetos Através da UML**. São Paulo: Makron Books, 1998.
- [9] IBOPE. **Pesquisa de opinião pública do IBOPE**. 2004. Disponível em: http://www.ibope.com.br/calandraWeb/BDarquivos/sobre_pesquisas/pesquisa_opinioao. Acessado em fevereiro de 2011.

- [10] ISMA. **O que é pesquisa de opinião pública?** Mato Grosso, 27 Junho 2007. Disponível em: <http://www.ismamt.blogspot.com/2008/04/o-que-pesquisa-de-opinião-pública>. Acessado em fevereiro de 2011.
- [11] JUNIOR, Luiz Antonio Zanetti. **Sistemas de informação baseados na tecnologia Web: Um estudo sobre seu desenvolvimento**. Dissertação (Mestrado em Administração) - Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo, São Paulo, 2003.
- [12] JUSTEN, Álvaro. **Web2py**. Revista Tldigital. Edição de Julho de 2010.
- [13] KUROSE, James F. ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top-down**. Tradução Arlete Simille Marques; revisão técnica Wagner Luiz Zucchi. 3 ed. São Paulo: Pearson Addison Wesley, 2006.
- [14] LARMAN, G. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos**. Porto Alegre: Bookman, 2000.
- [15] LUTZ Mark; ASCHER David. **Aprendendo Python**. Tradução João Tortello. 2.ed. Porto Alegre: Bookman, 2007. 568 p.
- [16] MATHEUS, Carlos. **Opinião pública e amostragem**. São Paulo, 29 de novembro, 2010. Disponível em: <http://carlosmatheus.org/opiniao-publica-e-amostragem/>. Acessado em fevereiro de 2011.
- [17] MURAGESAN, S.; DESHPANDE, Y.; HANSEN, S.; GINIGE, A. **Web engeneering: A new discipline for development of web-based systems**. In: Proceedings of the First ICSE Workshop on Web Engineering, Los Angeles, CA, USA: ACM Press, 1999, p. 794-795.
- [18] NÉTO, João Marques Brandão. **Como se faz pesquisa de opinião pública**. Pernambuco: Revista Eletrônica PRPE, Fevereiro, 2004.
- [19] O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da Internet**. São Paulo: Saraiva, 2003.

- [20] PIERRO, Massimo Di. **The official Web2py book**. 3 ed. Disponível em: <http://www.web2py.com/book/>. Acessado em julho de 2011.
- [21] PYTHON. Disponível em: <http://www.python.org>. Acessado em julho de 2011.
- [22] ROCHA, Bruno Cezar. **Desenvolvimento Web com Python**. Revista Espírito Livre. Edição 017. Agosto, 2010. Disponível em <http://revista.espiritolivres.org>. Acessado em julho de 2011.
- [23] ROCHA, Bruno Cezar. **Trabalhando com banco de dados no Web2py**. Revista Espírito Livre. Edição 019. Outubro, 2010a. Disponível em <http://revista.espiritolivres.org>. Acessado em julho de 2011.
- [24] RUTHFIELD, S. **The Internet's History and Development: From Wartime Tool to the Fish-Cam**. ACM Crossroads, New York, Jan.2001. Disponível em: <http://www.acm.org/crossroads/xrds2-1/inet-history.html>. Acessado em julho de 2011.
- [25] SILVA, Maurício Samy. **HTML 5: A linguagem de marcação que revolucionou a web**. São Paulo: Novatec Editora, 2011.
- [26] TANAKA, Oswaldo Y.; MELO, Cristina. **Avaliação de Programas de Saúde do Adolescente - um modo de fazer**. São Paulo: Edusp, 2001.
- [27] TANENBAUM, Andrew S. **Redes de computadores**. Tradução Vandenberg D. de Souza. 4.ed. Rio de Janeiro: Elsevier, 2003.
- [28] VIDGEN, R. et al. **Developing web information systems: from strategy to implementation**. London: Butterworth-Heinemann, 2002. 274p.

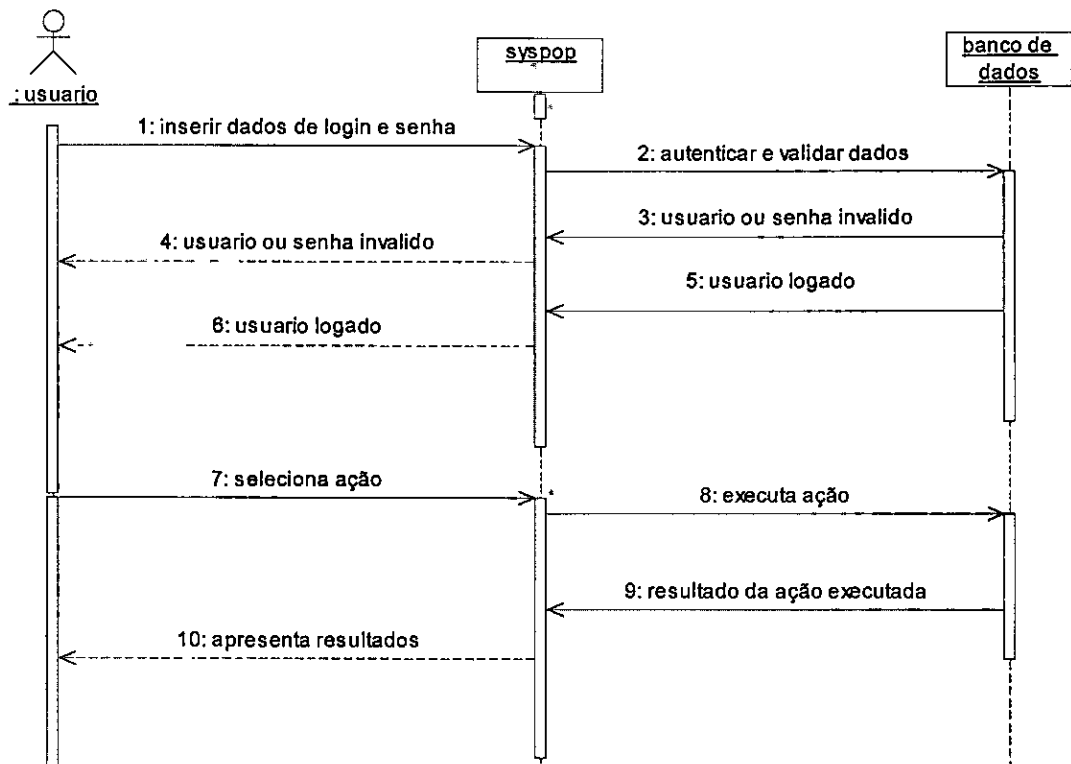
APÉNDICE

MODELAGEM DO BANCO DE DADOS DO SYSPOP



Retirado do WorkBench 5.1 OSS

DIAGRAMA DE SEQUÊNCIA GERAL DO SYSPOP



PARTE DOS CÓDIGOS REFERENTES A CRIAÇÃO DAS CLASSES DO BANCO DE DADOS

```
# coding: utf8

base = DAL('sqlite://base.sqlite')

from gluon.tools import Auth, Crud
auth = Auth(globals(),base)
auth.define_tables(username=True)
crud = Crud(base)

escolaridade = base.define_table('escolaridade',
    Field('descricao', 'string', length='60',
        label='Descrição'),
    format='%(descricao)s'
    )

escolaridade.descricao.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
endereco = base.define_table('endereco',
    Field('rua', 'string', length='60'),
    Field('numero', 'string', length='10',
        label='Número'),
    Field('cep', 'string', length='9'),
    Field('bairro', 'string', length='45'),
    Field('cidade', 'string', length='20'),
    Field('uf', 'string', length='2'),
    format='Rua: %(rua)s - N°: %(numero)s - Bairro:
    %(bairro)s - Cidade: %(cidade)s - UF: %(uf)s'
    )

endereco.rua.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
endereco.numero.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
endereco.cep.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
endereco.bairro.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
endereco.cidade.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
endereco.uf.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```



```
entrevistado = base.define_table('entrevistado',
    Field('nome', 'string', length='60'),
    Field('idade', 'integer'),
    Field('sexo', 'string', length='1'),
    Field('desc_escolaridade', escolaridade),
    Field('desc_endereco', endereco),
    format='% (nome) s'
)
```



```
entrevistado.nome.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
entrevistado.idade.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
entrevistado.sexo.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
entrevistado.desc_escolaridade.requires=IS_IN_DB(base,
'escolaridade.id', '%(descricao)s')
```

```
entrevistado.desc_endereco.requires=IS_IN_DB(base, 'endereco.id',
'Rua: %(rua)s - Nº: %(numero)s - Bairro: %(bairro)s - Cidade:
%(cidade)s - UF: %(uf)s')
```



```
entrevista = base.define_table('entrevista',
    Field('titulo', 'string', length='45'),
    Field('descricao', 'string', length='80'),
    format='% (titulo) s'
)
```



```
entrevista.titulo.requires=IS_NOT_EMPTY(error_message='Campo
obrigatório.')
```

```
entrevista.descricao.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
pesquisa = base.define_table('pesquisa',  
    Field('cod_entrevista', entrevista,  
        label='Entrevista'),  
    Field('cod_usuario', base.auth_user),  
    Field('cod_entrevistado', entrevistado),  
    )
```

```
pesquisa.cod_entrevista.requires=IS_IN_DB(base, 'entrevista.id',  
    '%(descricao)s', error_message='Não encontrado no banco.')
```

```
pesquisa.cod_usuario.requires=IS_IN_DB(base, 'auth_user.id',  
    'auth_user.first_name', error_message='Não encontrado no banco.')
```

```
pesquisa.cod_entrevistado.requires=IS_IN_DB(base, 'entrevistado.id',  
    '%(nome)s', error_message='Não encontrado no banco.')
```

```
pergunta = base.define_table('pergunta',  
    Field('descricao', 'string', length='250'),  
    Field('cod_pesquisa', pesquisa),  
    format='% (descricao)s'  
    )
```

```
pergunta.descricao.requires=IS_NOT_EMPTY(error_message='Campo obrigatório.')
```

```
pergunta.cod_pesquisa.requires=IS_IN_DB(base, 'pesquisa.id',  
    error_message='Não encontrado no banco.')
```

```
alternativa = base.define_table('alternativa',  
    Field('descricao', 'string', length='250'),  
    Field('cod_pergunta', pergunta),  
    format='% (descricao)s'  
    )
```

```
alternativa.descricao.requires=IS_NOT_EMPTY(error_message='Campo  
obrigatório.')
```

```
alternativa.cod_pergunta.requires=IS_IN_DB(base, 'pergunta.id',  
'%(descricao)s',error_message='Não encontrado no banco.')
```

PARTE DOS CÓDIGOS REFERENTES AO CONTROLLER DA APLICAÇÃO

```
# coding: utf8

@auth.requires_login()
def index():
    return dict()

#####

## crud de usuario

@auth.requires_membership('Administrador')
def criar_usuario():
    cod = base.auth_user.id
    msg = crud.create(base.auth_user,
next=URL('criar_membro_grupo'))
    return dict(msg=msg)

@auth.requires_membership('Administrador')
def lista_usuario():
    msg = base().select(base.auth_user.ALL)
    membros = base().select(base.auth_membership.ALL)
    grupos = base().select(base.auth_group.ALL)
    return dict(msg=msg, membros=membros, grupos=grupos)

@auth.requires_membership('Administrador')
def editar_usuario():
    msg = crud.update(base.auth_user, request.args(0),
next=URL('lista_usuario'))
    return dict(msg=msg)

#####

## crud de grupos de usuarios

@auth.requires_membership('Administrador')
def criar_grupo_usuario():
```

```

    msg = crud.create(base.auth_group, next=URL('lista_usuario'))
    return dict(msg=msg)

@auth.requires_membership('Administrador')
def lista_grupo_usuario():
    msg = base().select(base.auth_group.ALL)
    return dict(msg=msg)

@auth.requires_membership('Administrador')
def editar_grupo_usuario():
    msg = crud.update(base.auth_group, request.args(0),
next=URL('lista_grupo_usuario'))
    return dict(msg=msg)
#####

## crud de membro de grupos de usuarios

@auth.requires_membership('Administrador')
def criar_membro_grupo():
    msg = crud.create(base.auth_membership,
next=URL('lista_usuario'))
    return dict(msg=msg)

@auth.requires_membership('Administrador')
def lista_membro_grupo():
    msg = base().select(base.auth_membership.ALL)
    return dict(msg=msg)

@auth.requires_membership('Administrador')
def editar_membro_grupo():
    msg = crud.update(base.auth_membership, request.args(0),
next=URL('lista_usuario'))
    return dict(msg=msg)
#####

## crud de escolaridade (entrevistado)

```

```

@auth.requires_login()
def criar_escolaridade():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.escolaridade,
next=URL('criar_entrevistado'))
    return dict(msg=msg)

@auth.requires_login()
def lista_escolaridade():
    msg = base().select(base.escolaridade.ALL)
    return dict(msg=msg)

@auth.requires_login()
def editar_escolaridade():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.escolaridade, request.args(0),
next=URL('lista_escolaridade'))
    return dict(msg=msg)
#####

## crud de endereço (entrevistado)

@auth.requires_login()
def criar_endereco():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.endereco, next=URL('criar_entrevistado'))
    return dict(msg=msg)

@auth.requires_login()
def lista_endereco():
    msg = base().select(base.endereco.ALL)
    return dict(msg=msg)

@auth.requires_login()
def editar_endereco():
    crud.messages.delete_label = 'Marque para excluir'

```



```

        crud.messages.submit_button = 'Atualizar'
        msg = crud.update(base.endereco, request.args(0),
next=URL('lista_endereco'))
        return dict(msg=msg)
#####

## crud de entrevistado

@auth.requires_login()
def criar_entrevistado():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.entrevistado,
next=URL('lista_entrevistado'))
    return dict(msg=msg, escolaridade=escolaridade,
endereco=endereco)

@auth.requires_login()
def lista_entrevistado():
    msg = base().select(base.entrevistado.ALL)
    endereco = base().select(base.endereco.ALL)
    escolaridade = base().select(base.escolaridade.ALL)
    return dict(msg=msg, enderecos=endereco,
escolaridades=escolaridade)

@auth.requires_login()
def editar_entrevistado():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.entrevistado, request.args(0),
next=URL('lista_entrevistado'))
    return dict(msg=msg)
#####

## crud de entrevista

@auth.requires_login()
def criar_entrevista():

```

```

crud.messages.submit_button = 'Cadastrar'
msg = crud.create(base.entrevista, next=URL('lista_entrevista'))
return dict(msg=msg)

@auth.requires_login()
def lista_entrevista():
    msg = base().select(base.entrevista.ALL)
    return dict(msg=msg)

@auth.requires_login()
def editar_entrevista():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.entrevista, request.args(0),
next=URL('lista_entrevista'))
    return dict(msg=msg)
#####

## crud de pesquisa

@auth.requires_login()
def criar_pesquisa():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.pesquisa, next=URL('lista_pesquisa'))
    return dict(msg=msg)

@auth.requires_login()
def lista_pesquisa():
    msg = base().select(base.pesquisa.ALL)
    entrevistados = base().select(base.entrevistado.ALL)
    usuarios = base().select(base.auth_user.ALL)
    entrevistas = base().select(base.entrevista.ALL)
    return dict(msg=msg, entrevistados=entrevistados,
usuarios=usuarios, entrevistas=entrevistas)

@auth.requires_login()
def editar_pesquisa():

```

```

crud.messages.delete_label = 'Marque para excluir'
crud.messages.submit_button = 'Atualizar'
msg = crud.update(base.pesquisa, request.args(0),
next=URL('lista_pesquisa'))
return dict(msg=msg)

@auth.requires_login()
def pesquisa():
    pesquisas = base(base.pesquisa.id==request.args(0)).select()
    usuarios = base(base.auth_user.id==request.args(2)).select()
    entrevistados =
base(base.entrevistado.id==request.args(3)).select()
    entrevistas = base(base.entrevista.id==request.args(1)).select()
    respostas = base().select(base.resposta.ALL)
    alternativas = base().select(base.alternativa.ALL)
    perguntas = base().select(base.pergunta.ALL)
    return dict(pesquisas=pesquisas, usuarios=usuarios,
entrevistados=entrevistados, entrevistas=entrevistas,
respostas=respostas, altenativas=alternativas, perguntas=perguntas)
#####

## crud de pergunta

@auth.requires_login()
def criar_pergunta():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.pergunta, next=URL('criar_alternativa'))
    return dict(msg=msg)

@auth.requires_login()
def lista_pergunta():
    resposta = base().select(base.resposta.ALL)
    alternativas = base().select(base.alternativa.ALL)
    msg = base().select(base.pergunta.ALL)
    return dict(msg=msg, alternativas=alternativas,
resposta=resposta)

```

```

@auth.requires_login()
def editar_pergunta():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.pergunta, request.args(0),
next=URL('lista_pergunta'))
    return dict(msg=msg)
#####

## crud de alternativa de pergunta

@auth.requires_login()
def criar_alternativa():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.alternativa, next=URL('lista_pergunta'))
    return dict(msg=msg)

@auth.requires_login()
def lista_alternativa():
    msg = base().select(base.alternativa.ALL)
    return dict(msg=msg)

@auth.requires_login()
def editar_alternativa():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.alternativa, request.args(0),
next=URL('lista_pergunta'))
    return dict(msg=msg)
#####

## crud de resposta de pergunta

@auth.requires_login()
def criar_resposta():
    crud.messages.submit_button = 'Cadastrar'
    msg = crud.create(base.resposta, next=URL('lista_resposta'))

```

```
return dict(msg=msg)

@auth.requires_login()
def lista_resposta():
    msg = base().select(base.resposta.ALL)
    return dict(msg=msg)

@auth.requires_login()
def editar_resposta():
    crud.messages.delete_label = 'Marque para excluir'
    crud.messages.submit_button = 'Atualizar'
    msg = crud.update(base.resposta, request.args(0),
next=URL('lista_resposta'))
    return dict(msg=msg)
#####
```