

Como programar um wiki

Wiki Python

Os wikis vêm revolucionando a Internet. Saiba como programar um com apenas 150 linhas de código.

POR JOSÉ PEDRO ORANTES E JOSÉ MARÍA RUÍZ

Nesta edição vamos construir um wiki simples, baseado no site pioneiro do tipo, de Ward Cunningham. Seu wiki (ou mais corretamente, *WikiWiki*) é o mais antigo do mundo: Cunningham & Cunningham, Inc. [1] (por falar nisso, consulte o maior wiki do planeta para obter informações sobre ele em [2]).

O site é bastante simples. Seu sistema foi criado com o objetivo de que as pessoas pudessem acrescentar informação. Assim o site poderia crescer sozinho. Vamos reproduzir grande parte dele com um programa em *Python* de apenas 150 linhas, que deve ser baixado em [3]. Novamente, essa linguagem demonstra todo seu poder.

Nosso próprio WikiWiki

Criar um wiki no Python não é tão complicado como pode parecer. A receita é:

- 01 Muito CGI
- 02 Um pouco de expressões regulares
- 03 Um pouco de manipulação de diretórios
- 04 Um servidor Web ao gosto do leitor (mas que suporte CGI)

Com esses ingredientes vamos programar um wiki completo, que pode ser ampliado de acordo com a necessidade.

Funcionamento

Vamos ao primeiro passo para desenvolver qualquer programa: saber exatamente o que queremos. Seguindo as regras do c2.com, o wiki será composto por páginas interligadas. Os links serão palavras-chave com um formato especial: elas serão formadas por duas palavras juntas (sem espaços ou separadores), cada uma começando com uma letra maiúscula. Isso pode parecer confuso, como veremos em alguns exemplos. Primeiro, confira as palavras erradas, que não vão ser consideradas palavras-chave:

- ⇒ Olá
- ⇒ Uma Coisa

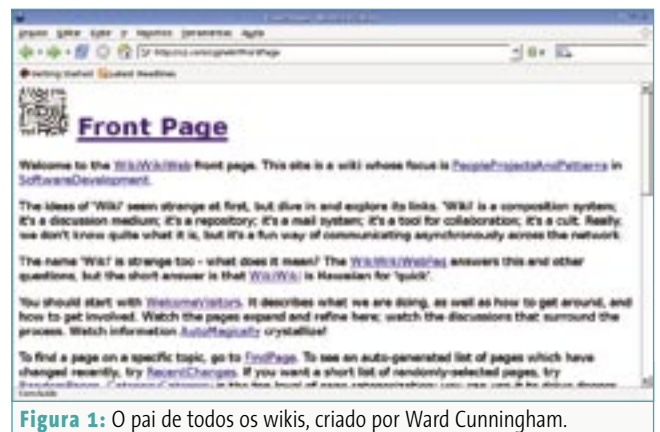


Figura 1: O pai de todos os wikis, criado por Ward Cunningham.

- ⇒ Outra-Coisa
- ⇒ softwareLivre
- ⇒ 3porquinhos

Nenhuma delas corresponde ao nosso padrão. A seguir, as que se enquadram:

- ⇒ OláMundo
- ⇒ UmDoisTrês
- ⇒ LinuxMagazine
- ⇒ FreeBSD

É por isso que o termo original é “WikiWiki”, em vez de simplesmente “wiki”.

Agora que já vimos as palavras-chave, vamos ver como elas são utilizadas. A verdade é que não há muito o que explicar, elas simplesmente devem ser digitadas e pronto! Por exemplo:

```
Este é um texto de um WikiWiki para a LinuxMagazine. Neste 2
exemplo, as palavras WikiWiki e LinuxMagazine devem se 2
converter automaticamente em palavras-chave.
```

Cada palavra-chave digitada pode virar uma página. Caso ela não contenha nenhum texto associado com ela, um ponto de interrogação surge ao lado da palavra. Basta clicar nele para acrescentar o texto. Em textos prontos, basta clicar no botão *Editar*.

O texto que vamos editar vai estar em formato *HTML*, assim podemos introduzir formatações como, por exemplo, listas ou negrito.

Uma vez editadas, as páginas poderão ser armazenadas e, a partir de então, a versão modificada já estará disponível. Na teoria, não deveria ser possível apagar páginas, já que o objetivo de um wiki é acumular informação. Como a única maneira de criar novas páginas é adicionar palavras-chave a páginas existentes, assim asseguramos que qualquer página será acessível a partir de outra. Tudo estará interligado, não haverá páginas soltas.

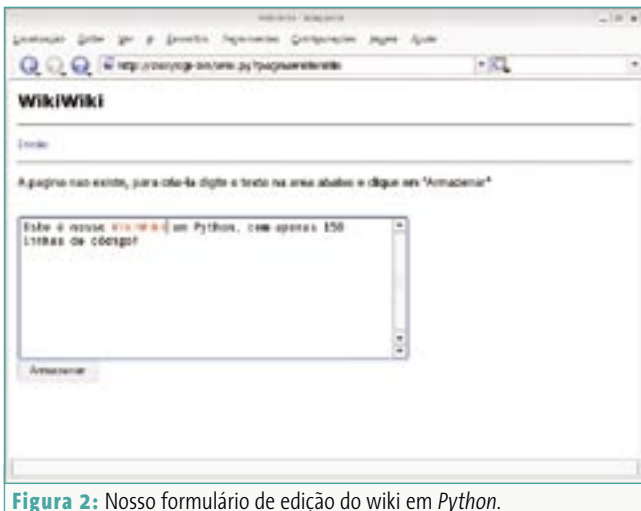


Figura 2: Nosso formulário de edição do wiki em Python.

CGI

CGI (*Common Gateway Interface*) é um protocolo que permite a um servidor web enviar e receber informações de um programa externo. Basicamente, o servidor web delega a geração das páginas a programas externos (que obtêm parâmetros para a criação de páginas a partir do servidor web).

Esses parâmetros podem ser enviados de duas maneiras distintas: *GET* e *POST*. Esses são os modos usados para a troca de informações entre o navegador e o servidor de web. O *GET* simplesmente passa os parâmetros através da URL (na barra de endereço do navegador) para o servidor.

Quando visitamos muitas das páginas da Internet, podemos ver que a URL tem um formato parecido com <http://www.algumaweb.org/algo.algo?variavel1=valor1&variavel2=valor>.

Nesse exemplo, o navegador está passando 2 variáveis ao servidor (*variavel1* e *variavel2*). A URL usa o símbolo “&” para separar variáveis entre si. Não é muito complicado.

Já o *POST* é diferente: o navegador e o servidor trocam informações usando o protocolo HTTP, de modo que não vemos as variáveis sendo passadas.

Para o nosso programa, o que importa não é usar *GET* ou *POST*, mas o conceito de CGI. Nosso programa recolherá variáveis e gerará páginas dependendo delas.

Para isso, usaremos o módulo *cgi* do Python:

```
import cgi
```

Esse módulo permite recolher as variáveis passadas pelo servidor web. Para isso, temos que usar a função `cgi.FieldStorage()`. Ela fornece um dicionário de objetos do tipo *FieldStorage*. Cada um contém a informação de uma das variáveis. De toda essa informação, só nos interessa o valor da variável, que podemos acessar com:

```
variables = cgi.FieldStorage()
valor = variables["nome"].value
```

Assim poderemos obter o valor da variável. Se for oferecido um dicionário, poderemos consultar se há alguma variável em particular usando o método `has_key()`: `variables.has_key("nome")`.

Formulários

A parte vital de um wiki é o poder de editar o conteúdo das páginas. Para isso, usaremos formulários em HTML para recolher as informações e enviá-las ao servidor web.

Os formulários são compostos de muitos controles, como, por exemplo, botões, campos de texto, rótulos e arquivos. Precisamos de um formulário com um campo extenso de texto e um botão para guardar o texto editado. Vamos ao básico.

Um formulário HTML fica entre duas tags: `<form>` e `</form>`. A primeira delas leva a uma série de dados que definem o comportamento do formulário. Queremos controlar o método de envio (POST ou GET) e o programa que receberá os dados (`wiki.py`).

```
<form method="POST"action="wiki.py">
...
</form>
```

Dentro do formulário são definidos os controles que serão mostrados. Cada um deles, por sua vez, define uma série de parâmetros. O primeiro deles será uma *textarea*.

```
...
<textarea cols="55" rows="10" name="texto"></textarea>
...
```

`cols` indica o número de colunas (de um tipo de largura) e `rows`, o número de linhas. O parâmetro `name` define o nome com que o conteúdo da *textarea* será enviado ao servidor web. Para simplificar, vamos usar o – pouco original – nome de “texto”.

O segundo tipo de controle de que necessitaremos é *input*:

```
...
<input type="submit" value="Editar"> </input>
<input type="hidden" name="editar" value="algo"> </input>
...
```

Aqui vemos os dois tipos de que necessitamos. O parâmetro `type` é usado para especificar o tipo, pois existem vários diferentes. No nosso caso, teremos:

`submit`, que gera um botão cujo título é recolhido do parâmetro `value`. Esse botão, ao ser pressionado, envia os dados ao servidor; e `hidden`, que é usado para passar ao servidor dados relacionados com controles. Em nosso caso, passamos a variável `editar=algo`.

Estrutura

Nosso wiki funcionará assim: ao ser acessado pela primeira vez, não havendo variáveis, será gerada uma página de apresentação. Essa página terá um botão que nos permitirá editá-la.

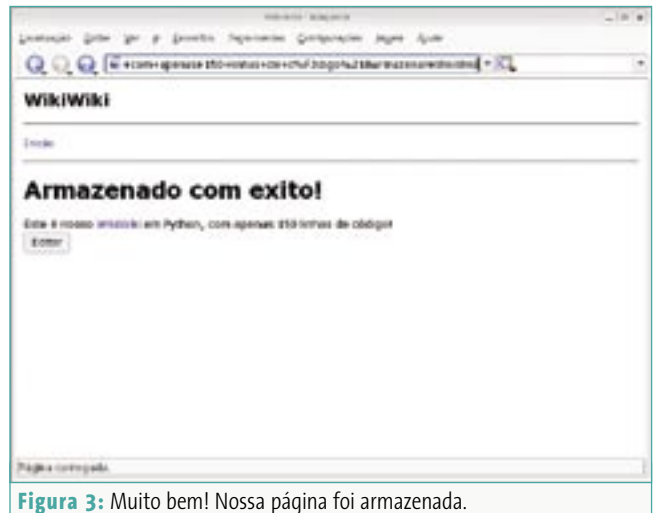


Figura 3: Muito bem! Nossa página foi armazenada.

Para isso, incluiremos um botão dentro do formulário HTML, que enviará uma variável `editar`, com a cadeia de caracteres “WikiWiki” como valor.

Ao ser iniciado novamente, teremos uma variável `editar` e, ao invés de gerar uma página, a aplicação buscará em seu diretório por um arquivo `WikiWiki.txt`. Se ele existir, o texto será carregado em uma *textarea*, que nos permitirá editá-lo. Ele virá acompanhado de um botão `armazenar`, que, ao ser pressionado, envia duas variáveis: `armazenar`, com o nome da página (WikiWiki, nesse caso); e `texto`, com o texto que editamos. Quando o `wiki.py` detectar as duas variáveis, irá guardar o conteúdo do texto em um arquivo com o nome que contenha a variável `armazenar`, concatenado com o `.txt`.

O texto, antes de ser armazenado, é analisado em uma busca por palavras-chave. No caso de elas existirem, será realizada a seguinte alteração, no caso de existir a página (ou seja, o arquivo `WikiWiki.txt`):

```
...WikiWiki... en ...<a href="wiki.py?pagina=WikiWiki"> ?
WikiWiki</a>...
```

ou realizará a seguinte alteração:

```
...WikiWiki... en ...WikiWiki <a href="wiki.py?pagina=WikiWiki"?</a>...
```

No caso de não existir uma página, um “?” deve ser colocado ao lado da palavra-chave para indicar que essa página deve ser criada. Para isso, teremos apenas que clicar sobre o “?”, digitar o texto e clicar em *Armazenar* (como antes).

Assim, teremos que controlar 4 situações:

➤ Se não houver variáveis, geraremos a página de boas-vindas.

- Se houver a variável `pagina`, carregaremos a correspondente.
- Se existir a variável `editar`, então carregaremos a página indicada em modo de edição.
- Se existirem as variáveis `armazenar` e `texto`, vamos armazenar o texto enviado.

Análise do texto

Para localizar as palavras-chave, usaremos expressões regulares (abordadas na **edição nº 17**). O importante agora é localizar a expressão regular adequada ao formato palavra-chave. Já vimos que elas são compostas de duas ou mais palavras, cada uma delas com a primeira letra em maiúscula.

Para indicar que queremos uma palavra com a primeira letra maiúscula e pelo menos duas palavras, usamos a seguinte expressão regular: `[A-Z][a-z]+`. Se quisermos indicar que são duas palavras, usaremos `[A-Z][a-z]+[A-Z][a-z]+`. Para a possibilidade de mais de duas palavras, basta colocar a segunda parte entre parênteses e um "+". Assim, chegamos à nossa expressão regular:

```
[A-Z][a-z]+([A-Z][a-z]+)+[A-Z]*[a-z]*
```

Agora que já podemos localizar as palavras-chave, vamos substituí-las no texto por links para suas respectivas páginas. Para isso, é preciso gerar uma lista a partir dos resultados da expressão regular. É aí que entra a função `findall()`, que fornece uma lista com todas as cadeias de caracteres que seguem o padrão de palavra-chave. Mas teremos um problema: algumas das cadeias de caracteres vão estar duplicadas.

Como vamos limpar os valores duplicados da lista? Isso pode ser feito de muitas maneiras (alguém na Internet bolou essa solução: `reduce(lambda 1, x: x not in 1 and 1.append(x)`

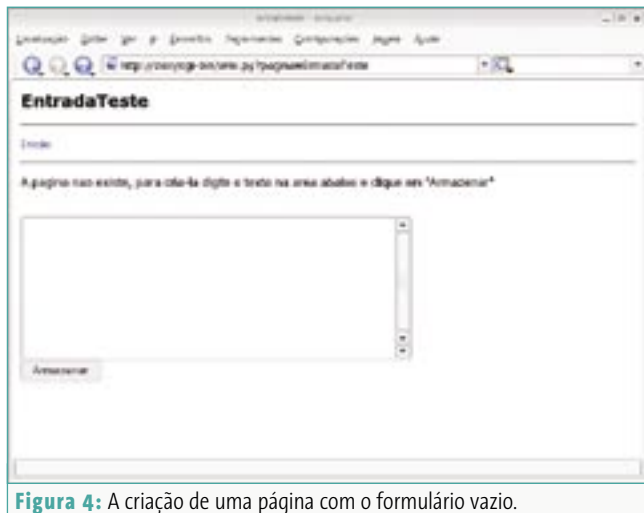


Figura 4: A criação de uma página com o formulário vazio.

or `1, a, []`), sendo `a` a cadeia de caracteres. Mas explicar essa simples linha levaria um artigo inteiro). Mais simples é empregar um pouco de matemática.

O Python permite a geração de um conjunto a partir de uma lista. Um conjunto pode conter diversos elementos, mas sua quantidade não importa, o importante é apenas que estejam presentes.

Vamos supor que temos a lista: `[1,1,1,3,4,5,5,6]` e que geraremos um conjunto usando a função `set([1,1,1,3,4,5,6])`. O resultado será o conjunto `{1,3,4,5,6}`. O Python permite interferir sobre um conjunto, extraindo um elemento diferente de cada vez. Isso nos permite realizar o seguinte truque:

```
01 >>> lista = [1,1,1,3,4,5,5,6]
02 >>> conj = set(lista)
03 >>> for elemento in conj:
04 ... print "Elemento: ", elemento
05 ...
06 Elemento: 1
07 Elemento: 3
08 Elemento: 4
09 Elemento: 5
10 Elemento: 6
11 >>>
```

É assim que converteremos nossa lista de palavras em um conjunto para coletarmos. Como já temos as palavras, teremos apenas que substituí-las pelos links correspondentes. Mas a página pode não existir. Então, teremos que testar essa condição com um `if`. Aí sim substituiremos a palavra-chave no conteúdo da página pelo link.

Usaremos a função `replace()` do `string.replace()`, que aceita 2 parâmetros. O primeiro é a cadeia a ser substituída e o segundo, a cadeia que será usada para substituição. Apenas uma execução de `replace()` já vai realizar todas as substituições de uma vez. Essa é a razão pela qual fizemos o truque do conjunto: se alguma cadeia de caracteres se repetisse na lista de palavras-chave, ocorreria o seguinte. Com a primeira, realizaríamos a alteração:

```
...WikiWiki... -> ...<a href="wiki.pl$pagina=WikiWiki">WikiWiki</a>...
```

Mas, ao encontrar outra vez a palavra-chave “WikiWiki” na lista, substituiríamos os “WikiWiki” recém-inseridos novamente pelo link:

```
...<a href="wiki.pl$pagina=WikiWiki">WikiWiki</a>... ->
...<a href="wiki.pl$pagina=...<a href="wiki.pl$pagina2
=WikiWiki">WikiWiki</a>...">...<a href="wiki.pl\2
pagina=WikiWiki">WikiWiki</a>...</a>..."
```

Esses métodos são usados na função `analise()` do arquivo `wiki.py`, vital para nosso programa.

Geração de HTML

Bem, agora nos encarregaremos de gerar o HTML. Graças ao CGI, a tarefa consiste apenas em imprimir o código HTML com `print`. O HTML devolvido pelo CGI tem uma peculiaridade: deve ser enviado com uma tag que indica o tipo de informação que ele contém. Além disso, essa tag tem que ir separada por dois *carriage returns* (enters) do resto da página. O código em questão é:

```
print "Content-Type:text/html"
print
print
```

Os prints sem parâmetros imprimem os enters. Para facilitar o design do código, dividimos a geração da página em 3 funções:

```
def cabeçalho(titulo)
def corpo(titulo, texto)
def editarCorpo(titulo,texto)
def rodape()
```

Toda página terá um cabeçalho, que imprime a parte não visível, como as tags `<title>`, e um rodapé, que fechará a página HTML (`</body> </html>`). No meio estará o corpo, que pode ser gerado por `corpo()` ou por `editarCorpo()`. Essa segunda função abre um formulário para a edição do conteúdo da página.

O texto da função `corpo()` é gerado normalmente usando uma variável `conteudo` do tipo `string`. Dependendo das circunstâncias, conterà uma informação ou outra.

As funções

```
def pagina(titulo, texto)
def editarPagina(titulo, texto)
```

se encarregam de gerar as páginas usando as 3 funções anteriores. Isso simplifica muito o código e nos permite gerar páginas de maneira simples.



Figura 5: A página de boas-vindas do nosso wiki.

Cada vez que temos que interagir com o sistema de arquivos, usamos:

```
def carregaPagina(titulo)
def armazenaPagina(titulo,texto)
```

A função `carregaPagina()` devolve o conteúdo da página que é passada como parâmetro.

Últimos detalhes

Temos que levar em consideração que o programa `wiki.py` deve ter permissões para escrita no diretório onde ele se encontra, visto que é aí que os arquivos são depositados. Também temos que lhe dar permissão de execução (basta o comando `chmod +x wiki.py`).

Resta apenas copiar o `wiki.py` para o diretório referente ao `localhost/cgi-bin`, para que ele seja executado quando digitamos no navegador: <http://localhost/cgi-bin/wiki.py>.

Aparecerá então uma página como a mostrada na figura 4. ■

INFORMAÇÕES

- [1] Cunningham & Cunningham, Inc.: <http://c2.com>
- [2] Ward Cunningham, Wikipédia: pt.wikipedia.org/wiki/Ward_Cunningham
- [3] Download do código wiki.py: www.linuxmagazine.com.br/issue/20/wiki.zip

AUTORES

José Pedro Orantes Casado cursa o 3º ano de Engenharia Técnica em Informática de Sistemas e, há muitos anos, usa linux como ferramenta de trabalho e sistema operacional de escritório. José María Ruiz está finalizando seu projeto de conclusão de curso de Engenharia Técnica em Informática de Sistemas e está há mais de 7 anos usando e desenvolvendo Software Livre (há dois anos no FreeBSD).