Universidade Tuiuti do Paraná
Faculdade de Ciências Exatas

PROXY CACHE

http://www.squid-cache.org/

André Luiz – andreluizsp@gmail.com

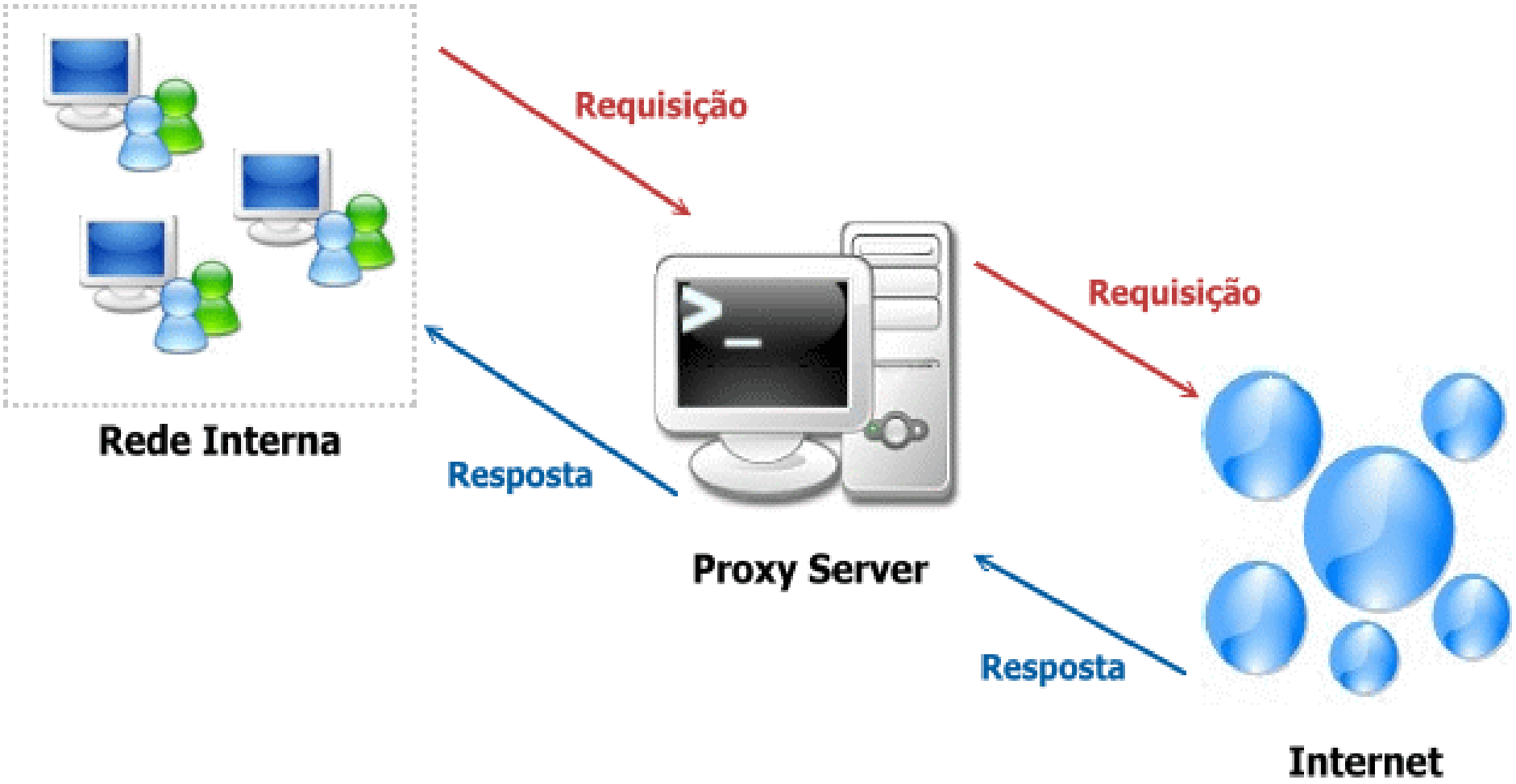
Conteúdo programático

- O que é um proxy cache?
- Instalando SQUID
- Conhecendo as principais TAGS da seção GLOBAL
- Conhecendo as principais ACLs
- Como usar a TAG http_access
- Configurando o squid.conf
- Script de configuração automática (proxy.dat)
- Configurando um Proxy transparente
- Autenticação com NCSA
- Relatórios com CALAMARIS
- Relatórios com SARG
- Protegendo acesso aos seus relatórios

O que é um Proxy Server?

Um proxy de caixa/cache HTTP ou em inglês caching proxy, permite por exemplo que o cliente requisite um documento na World Wide Web e o proxy procura pelo documento na sua caixa (cache). Se encontrado, a requisição é atendida e o documento é retornado imediatamente. Caso contrário, o proxy busca o documento no servidor remoto, entrega-o ao cliente e salva uma cópia na sua caixa (cache). Isto permite uma diminuição na latência, já que o servidor proxy, e não o servidor original, é requisitado, proporcionando ainda uma redução do uso da banda.

O que é um Proxy Server?



O que é um Proxy Server?

Um servidor proxy pode ser usado com basicamente três objetivos:

- ✔ Compartilhar a conexão com a Internet quando existe apenas um IP disponível (Apenas o Proxy com acesso WEB).
- ✔ Melhorar o desempenho do acesso através de um cache de páginas; Log de acesso, cache de memória e cache em disco;
- ✔ Impôr restrições de acesso (pornográficas, etc.) com base no horário, login, endereço IP e outras informações.

O que é o Squid?

O Squid é o servidor proxy HTTP mais comum em plataformas UNIX-Like. Ele surgiu de um projeto entre o governo americano e a Universidade do Colorado.

Atualmente o Squid trabalha com os protocolos HTTP, HTTPS, FTP, Gopher e WAIS e é o proxy que possui o maior número de co-projetos.

Instalação no Debian GNU/Linux (Squeeze)

```
# apt-get install squid
```

O arquivo squid.conf

O arquivo squid.conf é o principal arquivo de configuração do Squid. Ele zela pela simplicidade das tags, mas não muito pelo tamanho.

Ele é em si um manual das configurações disponíveis. Possui cerca de 2 mil linhas.

Antes de mexer no arquivo, é recomendável que você faça um backup do seu arquivo original, em caso quaisquer problemas:

 # cp /etc/squid/squid.conf /etc/squid/squid.conf.original

Tags mais usadas

http_port

Padrão: http_port 3128

Este parâmetro define a porta em que o serviço Squid irá escutar por requisições.

cache_mem

Padrão: cache_mem 8M

Este parâmetro configura a quantidade de memória utilizada para cache e objetos em trânsito, e não a quantidade de memória reservada para o Squid.

Tags mais usadas

cache_dir

Padrão: cache_dir ufs /var/spool/squid 100 16 256

Nesta opção são configurados os números de diretórios, subdiretórios e tamanho do cache. Desfragmentando a linha para estudo, ficaria assim:

Ufs: É a forma de armazenamento de cache.

/var/spool/squid - Diretório onde o cache do Squid ficará;

100: Espaço em disco que o cache do Squid poderá ocupar, contado em MB;

16: Quantidade de diretórios que o cache do Squid possuirá;

256: Quantidade de subdiretórios que o cache do Squid possuirá.

Tags mais usadas

cache_access_log

Padrão: cache_access_log /var/log/squid/access.log

Define o arquivo de log de acessos do Squid. Caso queira saber quem acessou determinada página da internet, é através deste arquivo que descobrirá.

cache_mgr

Padrão: cache_mgr email

Este parâmetro tem a finalidade de especificar o e-mail do administrador do proxy.

Tags mais usadas

cache_effective_user

Padrão: cache_effective_user squid

Informa ao Squid com qual nome de usuário ele deve rodar.

cache_effective_group

Padrão: cache_effective_group squid

Tem a mesma função da tag acima, mas ao invés de trabalhar com o usuário do Squid, ele vai trabalhar com o grupo.

Tags mais usadas

visible_hostname

Padrão: visible_hostname none

Ela é que define o hostname que fica visível nas mensagens de erro do Squid apresentadas para os clientes e,caso não seja setada, o Squid não inicia. Coloque alguma coisa nela parecida com:

visible_hostname seudominio.com.br

Access Control Lists (ACL's)

Uma ACL nada mais é do que a ferramenta que o Squid utiliza para especificar quem pode, quem não pode, o quê pode, o que não pode e quando.

ACL e controles

Formato da ACL:

acl *nomeacl tipo [“arquivo” | string]*

Formato do controle http_access:

http_access *[deny | allow] nomeacl*

Tipos de ACL's

src - Endereço IP de origem. Utilizada para especificar um determinado host ou uma determinada rede de origem.

dst - Endereço IP de destino. Utilizada para especificar um determinado host ou uma determinada rede de destino.

dstdomain - Valida domínio de destino.

srcdomain - Valida domínio de origem (cliente).

port - Número da porta de destino, usado para especificar acesso à determinada porta de um servidor.

url_regex - Utilizado para comparar uma string à uma URL inteira. Muito utilizado para fazer o bloqueio de sites indevidos.

Tipos de ACL's

urlpath_regex - Tem uma função semelhante à anterior, porém procura apenas em pedaços do caminho da URL. Muito utilizado para bloquear extensões.

proto - Especifica um protocolo de transferência.

proxy_auth - Somente utilizada caso você esteja utilizando autenticação. Serve para especificar nomes de usuários.

method - Especifica o tipo de método usado na requisição, como por exemplo GET, CONNECT ou POST;

time - Hora e dia da semana. Especifica um determinado horário.

Definindo ACL's

Dentro do arquivo de configuração do Squid, o squid.conf, você vai encontrar uma área que é a mais ideal para declarar as suas ACL's. Este espaço é onde as ACL's começam a ser definidas, facilmente identificada pela presença das mesmas.

Para declarar ACL's, a sintaxe básica é a seguinte:

acl <nome da acl> <tipo da acl> <string>|"<endereço de arquivo>"

Um exemplo prático de ACL

```
acl palavra_proibida url_regex -i sexo
```

A ACL acima bloqueia todos os sites que contenham em seu endereço a palavra "sexo".

tag http_access

É a tag http_access que trava ou libera o que a ACL está estipulando.

Exemplo:

```
http_access deny proibido
```

Se considerarmos o conjunto ACL + http_access, ficaria:

```
acl proibido url_regex -i sexo
http_access deny proibido
```

O que o conjunto acima faz é proibir que qualquer site que possua em seu endereço a palavra "sexo" seja exibido para o requisitante.

Squid.conf

1) Configuração básica:

http_port 3218

visible_hostname Proxy-ActiveInfo

acl all src 0.0.0.0/0.0.0.0

http_access allow all

2) Sempre que o arquivo de configuração for alterado o serviço squid deverá ser reiniciado (restart), ou deve reler o arquivo de configuração (reload):

service squid reload

Squid.conf

1) Configuração básica:

http_port 3218

visible_hostname Proxy-ActiveInfo

acl all src 0.0.0.0/0.0.0.0

http_access allow all

2) Sempre que o arquivo de configuração for alterado o serviço squid deverá ser reiniciado (restart), ou deve reler o arquivo de configuração (reload):

service squid reload

Configuração automática

Disponibilize na rede, em um servidor apache, o seguinte script proxy.dat :

```
function FindProxyForURL(url, host)
{
        if (shExpMatch(url, "*.dominio.local/*")) {return "DIRECT";}
        if (isInNet(host, "192.168.0", "255.255.255.0")) {return "DIRECT";}
        return "PROXY <url proxy>:<porta proxy>"
}
```

URL: http://<servidor>/proxy.dat

Ordem das ACL's

1 - O Squid vai ler todas as ACL's e testar se todas as ACL's declaradas possuem uma sintaxe correta e se elas estão sendo referenciadas por algum http_access;

2 - Depois disso, se ele iniciar normalmente (Pode ser que outros fatores impeçam isto), ele irá começar a testar todas as requisições que são feitas para ele e tentar casar as mesmas com as regras que as ACL's estipulam em conjunto com os http_access;

3 - Caso uma URL case com uma ACL, ele ignorará todas as outras ACL's para aquela requisição.

Ordem das ACL's

1 - O Squid vai ler todas as ACL's e testar se todas as ACL's declaradas possuem uma sintaxe correta e se elas estão sendo referenciadas por algum http_access;

2 - Depois disso, se ele iniciar normalmente (Pode ser que outros fatores impeçam isto), ele irá começar a testar todas as requisições que são feitas para ele e tentar casar as mesmas com as regras que as ACL's estipulam em conjunto com os http_access;

3 - Caso uma URL case com uma ACL, ele ignorará todas as outras ACL's para aquela requisição.

Ordem das ACL's

Uma outra maneira mais prática de tentar implementar isso é fazer da seguinte maneira:

1 - Primeiro coloque as ACL's que estipulam uma excessão à alguma regra de bloqueio que virá à seguir;

2 - Depois coloque as suas ACL's que vão bloquear sites e tudo o mais;

3 - Só então você coloca as suas ACL's liberando o acesso.

Habilitando o Squid

Para Squid iniciar, basta encontrar a linha:

http_access deny all

E alterar para:

http_access allow all

Depois reinicie o serviço:

/etc/init.d/squid restart

Restringindo o acesso ao Squid

Quando você encontrar a linha **http_access allow all**, ela vai estar liberando acesso à todos os hosts, já que a ACL **"all"** está especificando todos os hosts.

Para arrumar isto, você deve encontrar e comentar as linhas:

```
acl all src 0.0.0.0/0.0.0.0
http_access allow all
```

Restringindo o acesso ao Squid

Agora crie uma nova ACL do tipo **"src"**, especificando a rede interna:

```
acl redeinterna src 192.168.0.0/24
```

Agora autorize a ACL que você acabou de criar por meio de um http_access:

```
http_access allow redeinterna
```

Como visto acima, nós estamos somente permitindo o uso ao proxy pela rede interna. Agora, caso você queira especificar uma range de IP's, faça assim:

```
acl faixa_adm src 192.168.0.10-192.168.0.50
http_access allow faixa_adm
```

Bloqueando sites indevidos no proxy

O tipo de ACL **url_regex** serve para nós compararmos termos dentro de uma URL para que possamos compará-la e saber se esta palavra está ou não liberada e se os usuários vão ou não, visualizar a página.

Exemplo:

A seguinte ACL...

acl palavra url_regex -i sex

Bloqueie o acesso com o http_access:

http_access deny palavra

Lista de palavras proibidas

Vamos criar o arquivo texto que vai servir como lista de palavras bloqueadas e damos à ela permissões de leitura:

/etc/squid/lists/palavras_proibidas.lst

Nele insira todas as palavras proibidas. Lembre-se que você deve adicionar uma palavra por linha.

Após isto, nós criamos a ACL da seguinte maneira:

acl palavras url_regex -i "/etc/squid/lists/palavras_proibidas.lst“

Bloqueamos o acesso com o http_access:

http_access deny palavras

Exceções

Vamos criamos uma lista também para as palavras não bloqueadas.

/etc/squid/lists/palavras_liberadas

Depois você vai ter que juntar as duas ACL's em um único http_access, desta maneira:

http_access deny blocked !unblocked

Note a utilização do sinal de exclamação, significando uma inversão no sentido da regra.

Restringindo o horário de acesso

Para nós fazermos isto, nós fazemos o uso da ACL do tipo **time:**

```
acl horariopermitido time MTWHF 08:00-18:00
http_access deny !horariopermitido
```

Interpretando a regra fica assim: Ele vai negar o uso do proxy em todos os horários, **COM EXCESSÃO** do horário especificado na ACL horariopermitido.

Você deve estar estranhando o **"MTWHF"** na frente da ACL. Ela especifica os dias da semana conforme abaixo:

S - Sunday (Domingo), M - Monday (Segunda), T - Tuesday (Terça), W - Wednesday (Quarta), H - Thursday (Quinta), F - Friday (Sexta), A - Saturday (Sábado)

O chefe...

O seu chefe quer ter o acesso completo...

Para contornar a situação, faça o seguinte:

acl chefe src IP.DO.MICRO.CHEFE

http_access allow chefe

Só acessa site liberado

Primeiramente, especifique os sites que eles irão poder acessar:

acl bancos url_regex -i "/etc/squid/lists/bancos“

Adicione os endereços dos sites liberados na lista e especifique também o IP do computador do usuário:

acl peao src IP.DO.MICRO.USUARIO

Então junte os dois em um único http_access, desta maneira:

http_access deny !bancos peao

Bloqueando extensões e downloads

Bloquear extensões que os seus usuários baixam no computador por meio de HTTP ou de FTP e de quaisquer outros protocolos que o Squid suporte:

/etc/squid/lists/extensoes

Você deve escrever as extensões que você quer bloquear da seguinte maneira no arquivo:

\.mp3$

\.wav$

\.pif$

\.bat$

NOTA: O "\" é um eliminador de metacaracteres e serve para cancelar a função do ".". Já o "$" serve para que seja analisado até o final da string.

Bloqueando extensões e downloads

Agora nós vamos adicionar a ACL no Squid que vai bloquear as extensões efetivamente, juntamente com o seu http_access:

```
acl extensoes urlpath_regex -i "/etc/squid/lists/extensoes“

http_access deny extensoes
```

Ao invés do **url_regex**, foi utilizado o **urlpath_regex**.

Segurança em um servidor Squid

Todo servidor, independente de plataforma ou serviço que executa, precisa de certas configurações de segurança.

Os principais erros de segurança que as pessoas deixam passar quando estão configurando o Squid, serão mostrados a seguir.

Erro I: Definição de relay do servidor Squid

Assim como a maioria dos servidores, o Squid também possui suas configurações de relay. Entenda por relay o "lado para qual o servidor está rodando, se é para a internet ou para a rede interna".

Claro que é interessante que você monte um servidor para a sua rede interna. Sendo assim, você deveria se preocupar com o relay do seu servidor Squid, pois geralmente os usuários costumam disfarçar seus IP's utilizando servidores proxy abertos pela internet, prática conhecida como IP Spoofing.

Erro I: Definição de relay do servidor Squid

Temos duas maneiras de resolver este problema.

1ª Maneira: http_port

Você pode editar esta tag para que o Squid só escute requisições vindas da rede interna, desta maneira:

http_port 192.168.1.1:3128

Erro I: Definição de relay do servidor Squid

2ª Maneira: acl all src 192.168.0.0/24

Você pode permitir o http_port como mencionado acima, mas deverá definir a sua rede interna quando for utilizar as suas ACL's, desta maneira:

acl all src 192.168.0.0/24

O Squid somente irá permitir conexões vindas da rede interna.

Erro II: http_access libera geral

Se o Squid não encontra uma ocasião pela qual encaixar uma ACL, ele simplesmente libera a sua utilização.

Coloque esta linha no final das declarações das suas ACL's para evitar essa situação:

http_access deny all

Erro III: Usuários

Geralmente usuários removem as configurações de proxy do navegador para navegarem sem restrições e somente pelo NAT ou procuram por proxies abertos na internet e configuram os navegadores para utilizarem estes endereços ou ainda vão em páginas como o "anonymizer" ou similares e navegam por lá.

Solução: bloqueio dessas páginas.

Erro IV - Proxy Transparente sem ser no gateway da rede

Os usuários podem alterar as configurações de rede e fazer com que o gateway da rede aponte para o roteador final, fazendo um caminho alternativo para os pacotes e fazendo com que os mesmos não passem pelo servidor Squid.

Solução: ...

Relendo o Squid.conf

Qualquer uma das opções abaixo irá fazer o squid validar as novas regras:

```
# squid -k reconfigure
# service squid reload
# service squid restart
```

Obs.: A terceira forma fecha as conexões ativas (stop/start).

Autenticação NCSA

1) Na sessão GLOBAL:

auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd

auth_param basic children 5

auth_param basic realm Servidor MeuProxy

auth_param basic credentialsttl 2 hours

2) Na seção das ACLs:

acl usuarios proxy_auth REQUIRED

3) Nas regras de acesso:

http_access allow usuarios

4) Criando usuários: # htpasswd /etc/squid/passwd <usuário>

Relatórios de acesso

CALAMARIS

apt-get install calamaris

mkdir /var/www/calamaris

calamaris -a -F html <caminho>/access.log > /var/www/calamaris/index.html

Acesse:

http://localhost/calamaris

Pode ser usado no cron para rodar diariamente.

Relatórios de acesso

SARG

1) Execute: # apt-get install sarg

2) Configure o arquivo /etc/sarg/sarg.conf
   Ex.: output_dir /var/www/squid-reports

3) Execute: # sarg

4) Acesse:

http://localhost/squid-reports

Pode ser usado no cron para rodar diariamente.

Restringindo acesso aos relatórios

Crie os dois arquivos abaixo:

 # cd /var/www/squid-reports

/var/www/squid-reports# htpasswd -c .passwd admin
Senha:
Repita a senha:

/var/www/squid-reports# vim .htaccess

AuthUserFile /var/www/arquivos/.passwd
AuthGroupFile /dev/null
AuthName "Digite sua senha"
AuthType Basic

require valid-user

Restringindo acesso aos relatórios

Adicione as linhas abaixo no arquivo *<u>/etc/apache2/sites-available/default</u>* :

DocumentRoot /var/www
<Directory />
Options FollowSymLinks
AllowOverride **AuthConfig**
</Directory>
<Directory /var/www/>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

<Directory "/var/www/squid-reports">
AllowOverride All
Options IncludesNoExec Indexes
AddOutputFilter Includes html
AddHandler type-map var
Order allow,deny
Allow from all
LanguagePriority pt-br
ForceLanguagePriority Prefer Fallback
</Directory>

Restringindo acesso aos relatórios

Reinicie o servidor apache:

 # service apache2 restart

Acesse:

http://<ip servidor>/squid-reports

creative commons

Atribuição-Compartilhamento pela mesma licença 2.5
(http://creativecommons.org/licenses/by-sa/2.5/deed.pt)

Você pode:

- copiar, distribuir, exibir e executar a obra
- criar obras derivadas
- fazer uso comercial da obra

Sob as seguintes condições:

Atribuição. Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

Compartilhamento pela mesma Licença. Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

- Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
- Qualquer uma destas condições podem ser renunciadas, desde que Você obtenha permissão do autor.

Qualquer direito de uso legítimo (ou "fair use") concedido por lei, ou qualquer outro direito protegido pela legislação local, não são em hipótese alguma afetados pelo disposto acima.

Este é um sumário para leigos da Licença Jurídica
(na íntegra: http://creativecommons.org/licenses/by-sa/2.5/br/legalcode).
Termo de exoneração de responsabilidade:
http://creativecommons.org/licenses/disclaimer-popup?lang=pt