

# Servidor RubyOnRails, com Ruby 1.9.2

Luiz Soares de Jesus  
[www.adsitape.com/contato](http://www.adsitape.com/contato)  
Cesário Lange-SP, dezembro/2011

1.	Avisos Legais.....	1
1.1.	Trademarks .....	1
1.2.	Responsabilidade.....	1
1.3.	Direitos de Cópia.....	1
2.	Proposta .....	2
2.1.	Por quê um Cloud Server?.....	2
2.2.	Comprando .....	3
3.	Configurando um Cloud Server para Rails .....	5
3.1.	Criando um modelo.....	5
3.2.	Criando a Máquina Virtual para Testes .....	6
3.2.1.	Aviso para login .....	11
3.3.	Clonando a Máquina-Base.....	11
4.	Procedimentos de Instalação .....	13
4.1.1.	Pré-Instalação.....	14
4.2.	SSH.....	15
4.3.	Instalando Ruby 1.9.2 .....	15
4.3.1.	Compilando manualmente o módulo OpenSSL .....	18
4.4.	Instalando o RubyGems.....	19
4.5.	Instalar o Rails .....	21
4.6.	PostGreSql 9.1.2 .....	22
4.6.1.	Pós-instalação : Script de Inicialização .....	24
4.6.2.	Pós-instalação: Localização do PID.....	25
4.7.	Apache .....	26
4.8.	Passenger.....	28
4.8.1.	Instalando .....	29
4.8.2.	Variável APXS2 .....	29
4.8.3.	Continuando.....	30

4.9.	Colocando a aplicação Rails para rodar .....	31
4.9.1.	Transferindo a aplicação para o servidor virtual .....	31
4.9.2.	SUCESSO!!! .....	43
5.	Máquina Real.....	45
5.1.	Último problema .....	45
5.2.	Concluindo .....	46

# 1. Avisos Legais <sup>1</sup>

## 1.1. Trademarks

Todas as marcas relacionadas neste documento (principalmente, mas não exclusivamente: Ruby, Rails, RubyOnRails, Locaweb, Debian, Apache, Phusion e Phusion Passenger), são propriedade exclusiva de seus respectivos proprietários, os quais em nenhum momento endossaram ou recomendaram este conteúdo.

Não foram citados os símbolos ® ou © nos devidos locais porquê... dá muito trabalho.

## 1.2. Responsabilidade

O autor assume que os procedimentos informados neste material funcionaram a contento para ele, em dezembro/2011. No entanto, não é dada nenhuma garantia de que os mesmos procedimentos funcionarão da mesma forma para qualquer pessoa que decidir utilizá-los.

## 1.3. Direitos de Cópia

```
/*
*-----
BEERWARE LICENSE 2 (revisão própria, baseada na revisão 43)
<luiz.jesus@fatec.sp.gov.br> escreveu este artigo. Enquanto você
retiver esta nota, você pode fazer o que quiser com este conteúdo.
Caso nos encontremos algum dia e você ache que este material lhe foi
útil de alguma forma, você poderá me comprar uma cerveja em
retribuição (ou mais de uma); caso nunca nos encontremos, você ainda
poderá tomar uma cerveja em minha honra. Luiz Soares de Jesus.
*-----
*/
```

Nos casos em que foi utilizado trabalho específico de terceiros, tentei manter as referências adequadas, diretamente no texto ou em notas de rodapé. Caso você encontre alguma falha nesse processo, peço o favor de me avisar para que possa fazer as devidas correções.

---

<sup>1</sup> Nos dois sentidos da palavra.

<sup>2</sup> <http://pt.wikipedia.org/wiki/Beerware> e <http://people.freebsd.org/~phk/>

## 2. Proposta

Demonstrar a configuração de um ambiente Cloud Server capaz de atuar como servidor para aplicação Ruby on Rails, utilizando sistema operacional Debian 6, Ruby 1.9.2, Rails 3.1.3, PostgreSQL 9.1.2, Http Server Apache 2.2.21 e Phusion Passenger.

### 2.1. Por quê um Cloud Server?

Para hospedar o site [www.adsitape.com](http://www.adsitape.com), a primeira idéia era utilizar o serviço de hospedagem compartilhada da Locaweb, muito barato, R\$27,00 por mês. Há milhares de reclamações na internet com os serviços da Locaweb, mas também contra os de outras empresas, então optei por começar em um ambiente conhecido.

Com o rascunho da aplicação pronto, e com a hospedagem compartilhada contratada, segui as instruções no wiki ([http://wiki.locaweb.com.br/pt-br/Como\\_colocar\\_minhas\\_aplica%C3%A7%C3%B5es\\_Rails\\_na\\_Locaweb%3F](http://wiki.locaweb.com.br/pt-br/Como_colocar_minhas_aplica%C3%A7%C3%B5es_Rails_na_Locaweb%3F)), e tudo que conseguia era uma página de erro do Passenger. Após pesquisar muito, concluí que o ambiente de hospedagem compartilhada da Locaweb não suporta aplicações Rails 3.1. É muito provável que esta conclusão esteja errada, e que na verdade eu é que estivesse fazendo alguma bobagem. Parece que é possível usar a versão 3.0.9, e com certeza dá pra usar a 2.3 (até já fiz um teste há uns 8 meses atrás), mas eu quero usar a versão 3.1.3 nesse projeto.

Pesquisando as informações disponíveis nos sites de outros grandes servidores de hospedagem (3 horas de busca), conclui que teria o mesmo problema em todos eles.

A opção que restava era configurar meu próprio servidor, onde pudesse pôr o que me desse na telha. A primeira solução seria ter um servidor em casa, uma máquina que rodasse 24 x 7. Acontece que o objetivo é focar no desenvolvimento da aplicação. Para ter um servidor próprio, precisaria me preocupar com questões como link estável, disponibilidade garantida de energia elétrica, upgrades, e outros itens que desviariam minha atenção.

Daí para optar por um Cloud Server não levou mais que alguns segundos. O custo ao longo do tempo é relativamente alto (a versão mais básica custa R\$ 149,00

mensais). Supondo que eu tivesse que comprar uma máquina para um servidor caseiro, gastaria em torno de R\$ 1500,00, porque teria que começar com uma máquina de porte razoável, se contar instalação elétrica reforçada e no-break respeitável, já vai para uns R\$ 3000,00, e ainda conviveria com um link speedy, pouco confiável para o objetivo. O valor de R\$ 3000,00 corresponde a 20 meses de aluguel do Cloud Server.

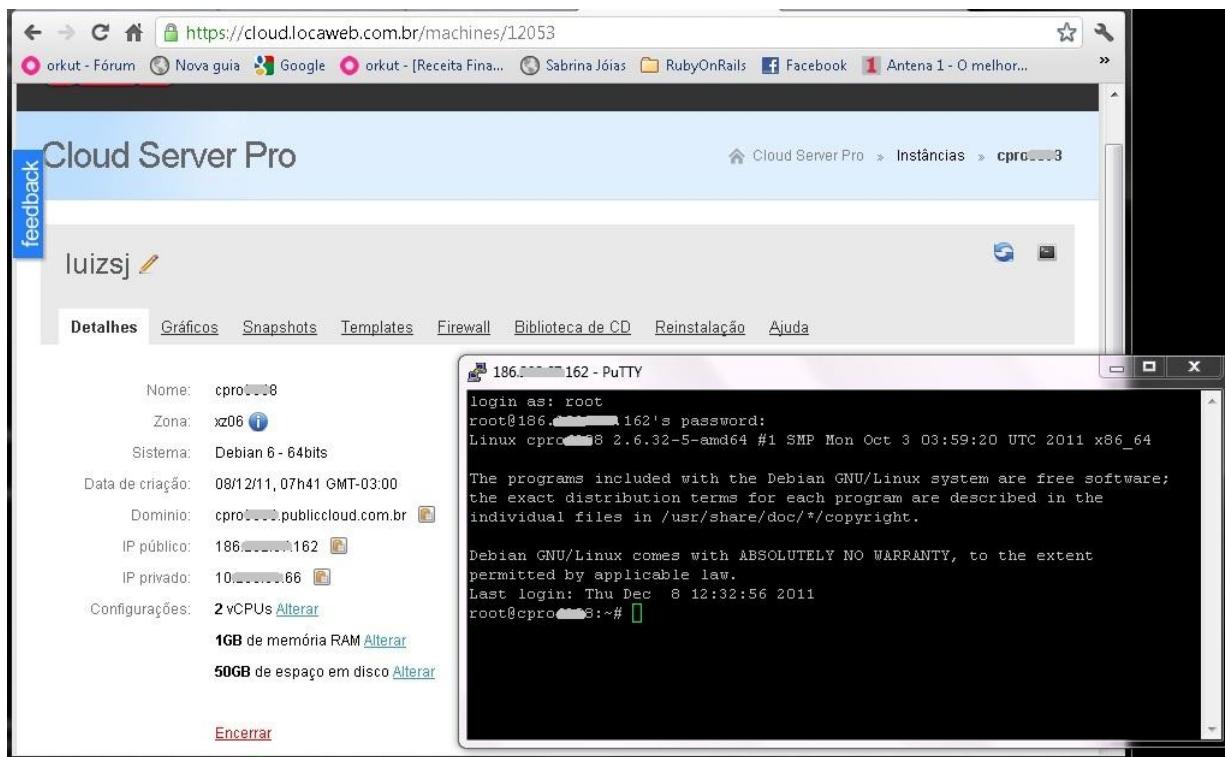
Se em 6 meses o projeto vingar, tenho a opção de fazer upgrade na máquina virtual, sem me preocupar em comprar e instalar o hardware. Se o projeto morrer, basta encerrar o aluguel e pronto, e terei gasto 1/3 do valor mínimo para ter servidor próprio. Não vi como é nas outras empresas, mas na Locaweb o sistema é como celular pré-pago: paga a mensalidade e tem 30 dias do servidor à disposição. A hora que não quiser mais é só não pagar, sem multas e nem nada.

Leia os avisos legais na página 1.

## 2.2. Comprando

Simples, rápido e indolor. Só seguir as instruções no site da Locaweb.  
Sequência:

- Minuto 0: cadastro preenchido
  - + 2 minutos: boleto da primeira mensalidade pago via net banking
  - + 2 minutos: comprovante de pagamento enviado para agilizar a instalação
    - + 20 horas: e-mail recebido avisando para fazer a configuração inicial (decidir nome para a máquina, senha do root, zona de ip (este último só é importante se for agregar uma nova máquina a um conjunto pré-existe)).
  - + 2 horas: máquina disponível para login via SSH:



Resultado: em menos de 24 horas uma máquina disponível na nuvem.

### 3. Configurando um Cloud Server para Rails

Aí começa o trabalho pesado, desnecessário quando se opta pela hospedagem compartilhada.

O sistema operacional é disponibilizado totalmente “cru”, contando apenas com firewall pré-configurado.

Preciso torná-lo um servidor web capaz de atender a aplicações Rails.

Itens necessários:

- servidor web Apache
- interpretador Ruby 1.9.2
- framework Rails 3.1.3
- banco de dados PostGreSql
- toque final: Passenger

O Passenger é uma ferramenta que interage com o Apache, indo até o Rails, pegando o que ele devolve e repassando de volta para que o Apache responda ao usuário que estiver acessando a aplicação.

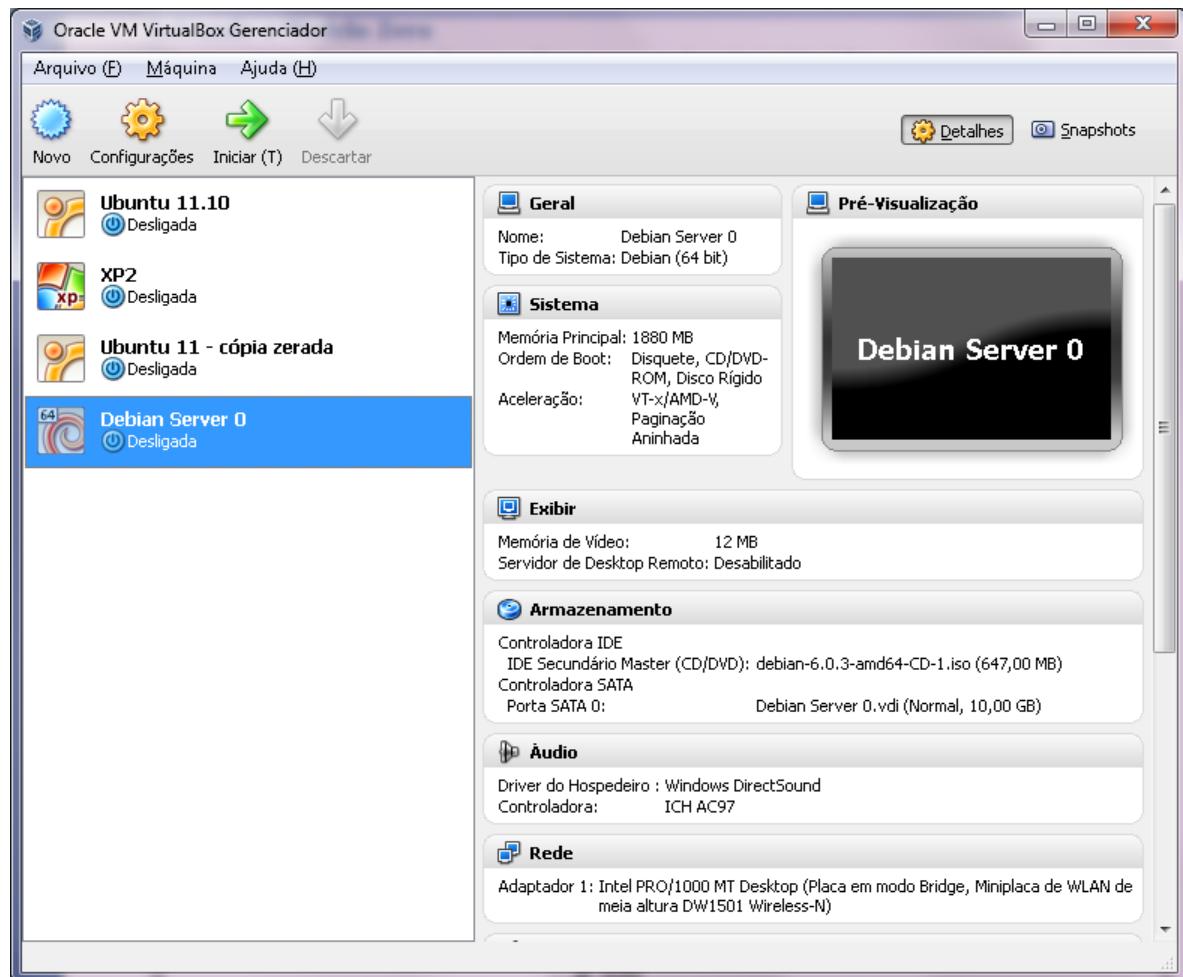
#### 3.1. Criando um modelo

Para não interagir direto com o Cloud Server (ficaria muito frustrado se fizesse algo muito errado e tivesse que solicitar à Locaweb uma “formatação” da máquina para reiniciar do zero), optei por configurar uma máquina virtual aqui no meu computador. Aí posso errar e reiniciar quantas vezes quiser, e quando for para o Cloud Server teoricamente já terei um passo-a-passo (obviamente ainda vão ocorrer coisas inesperadas quando chegar lá, mas o feijão-com-arroz já estará pronto).

Para isso, baixei a imagem ISO do CD-1 do Debian 6, e criei uma máquina virtual usando o Virtual Box. Para o download, usei BitTorrent, poderia fazer via FTP ou http normal, mas bittorrent aproveita o máximo da conexão (speedy 2 mbs).

### 3.2. Criando a Máquina Virtual para Testes

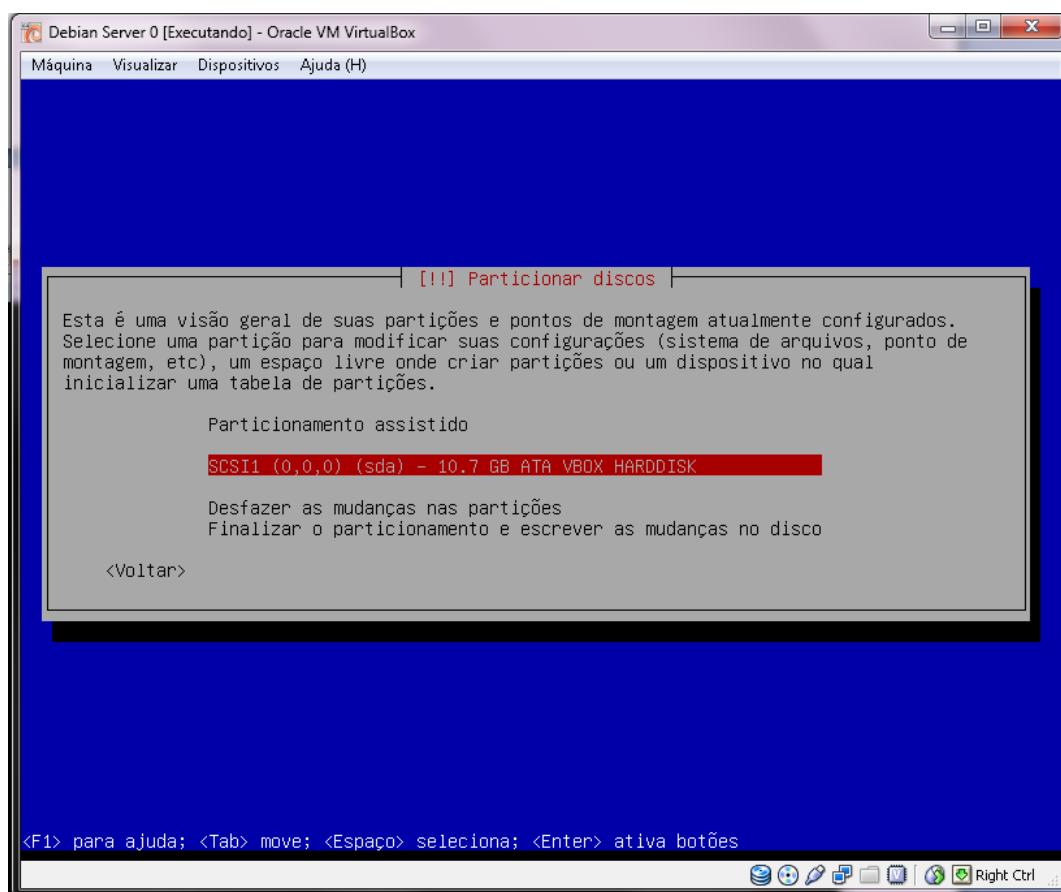
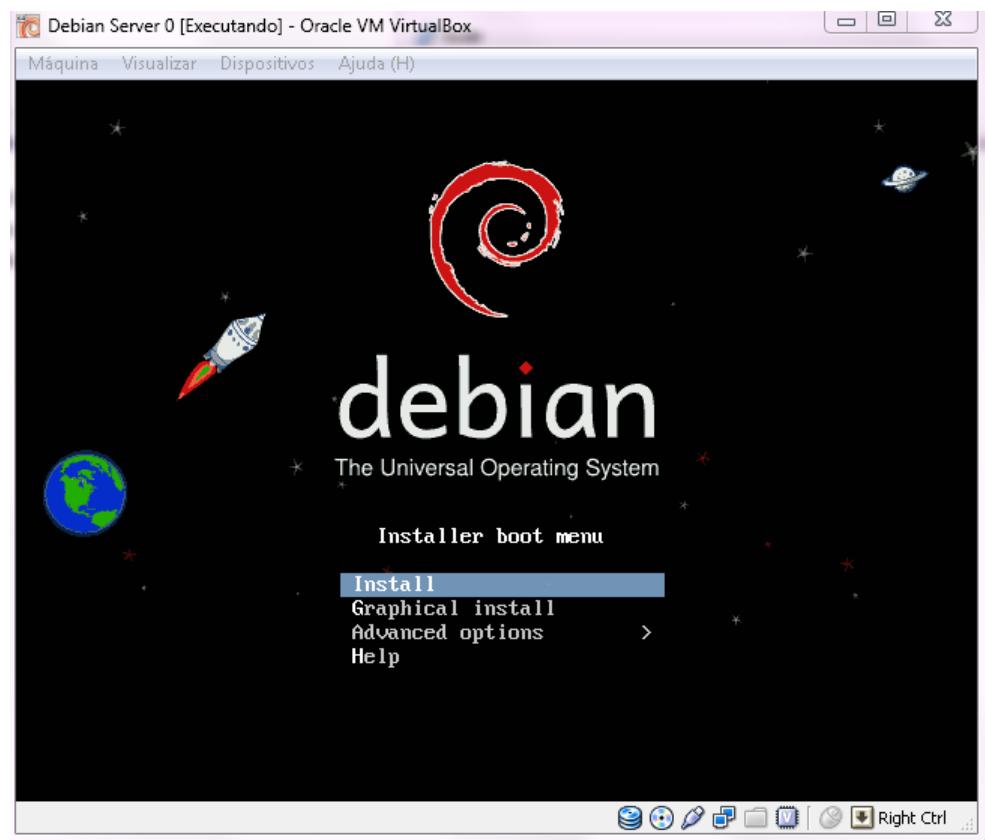
Para esta fase, estou partindo do princípio de que você esteja familiarizado com criação de máquinas virtuais utilizando o VirtualBox, ou qualquer outro programa de máquinas virtuais que você goste. Como a criação da máquina virtual, não é o foco deste tutorial, vou colocar aqui mais print-screens do que comentários.

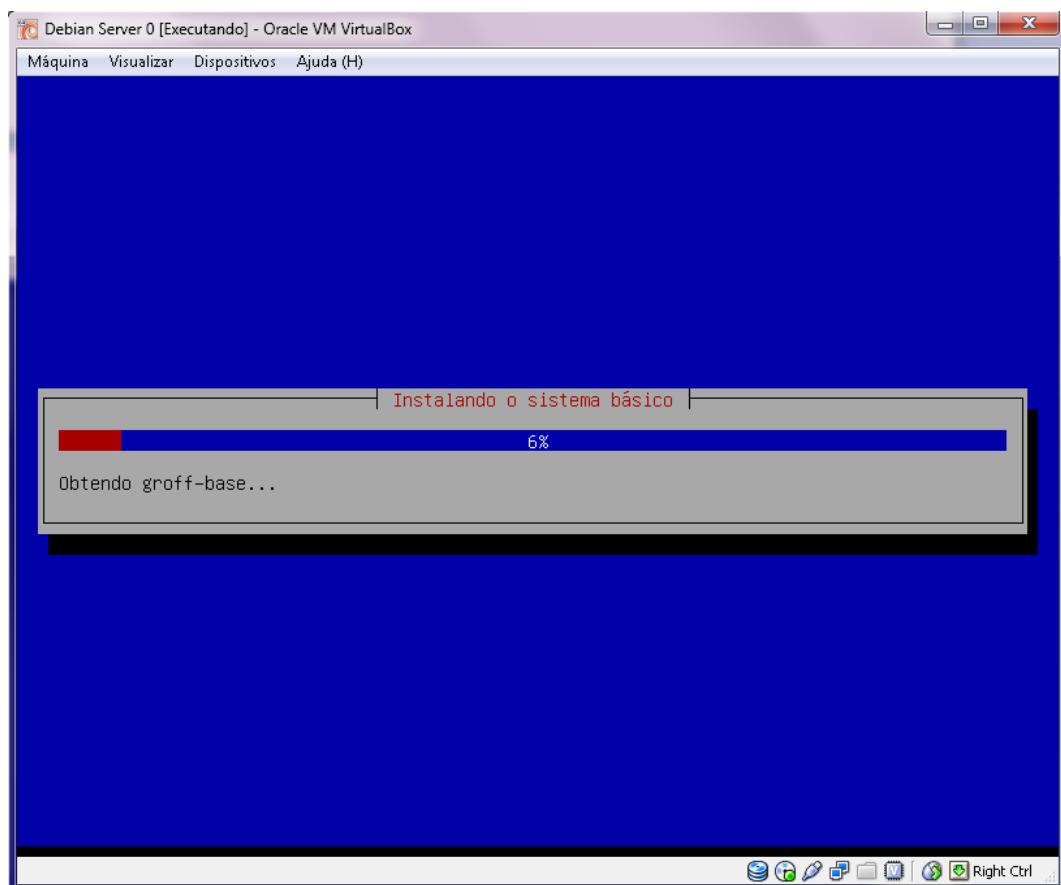
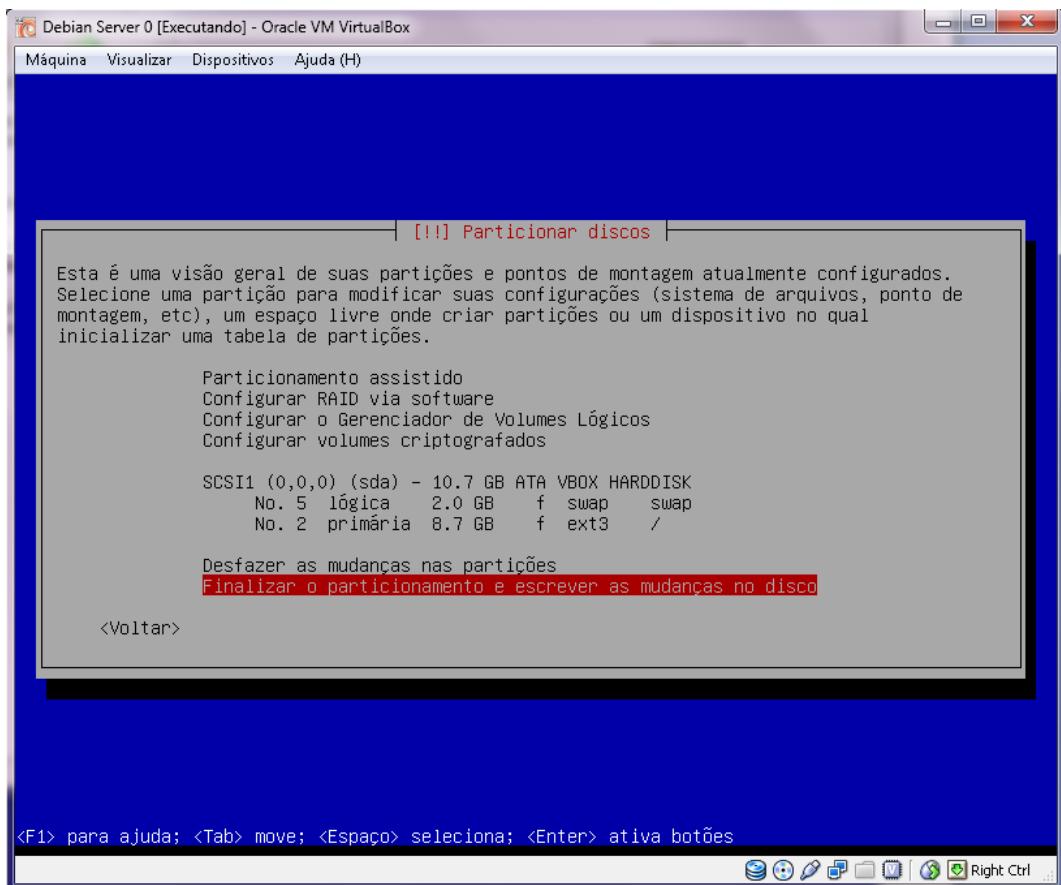


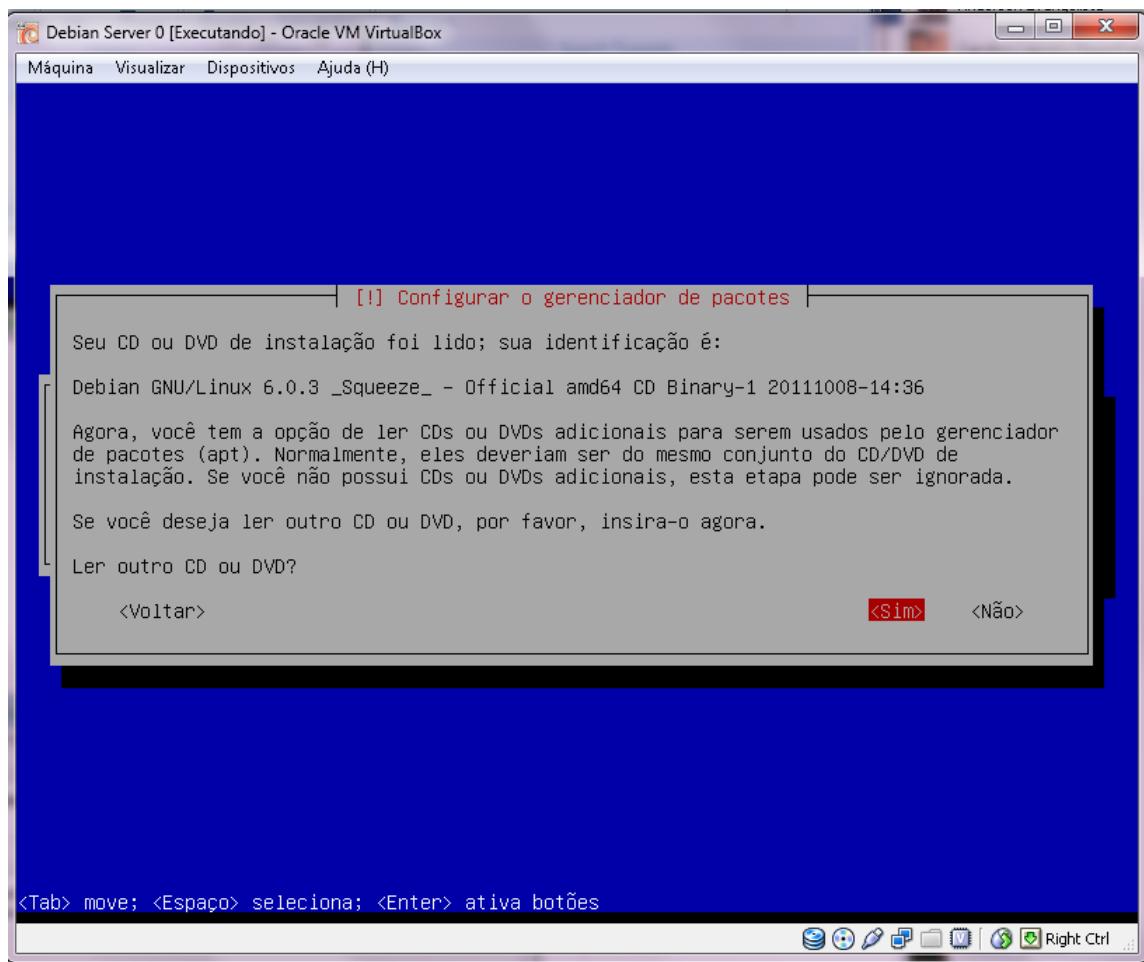
1.8 gb de RAM

Disco de 10 GB

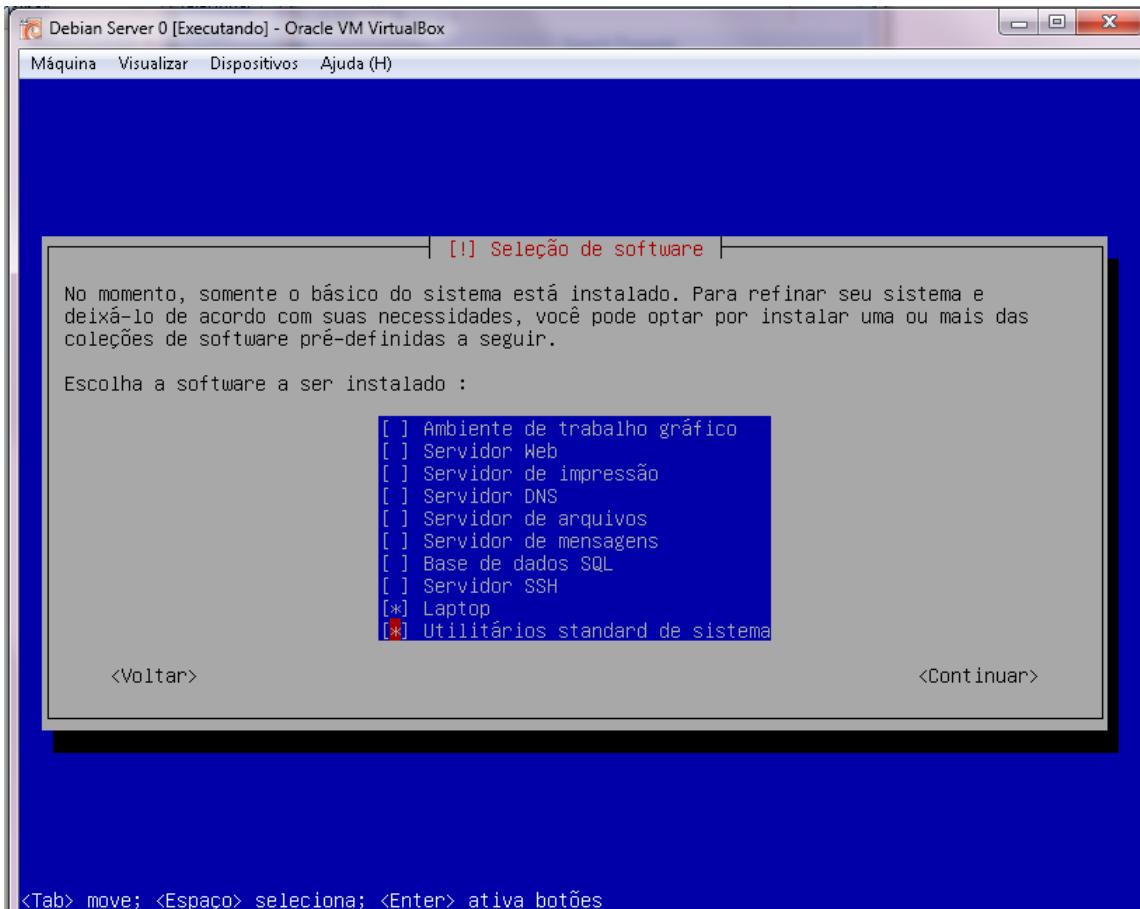
Placa de rede em modo bridge.



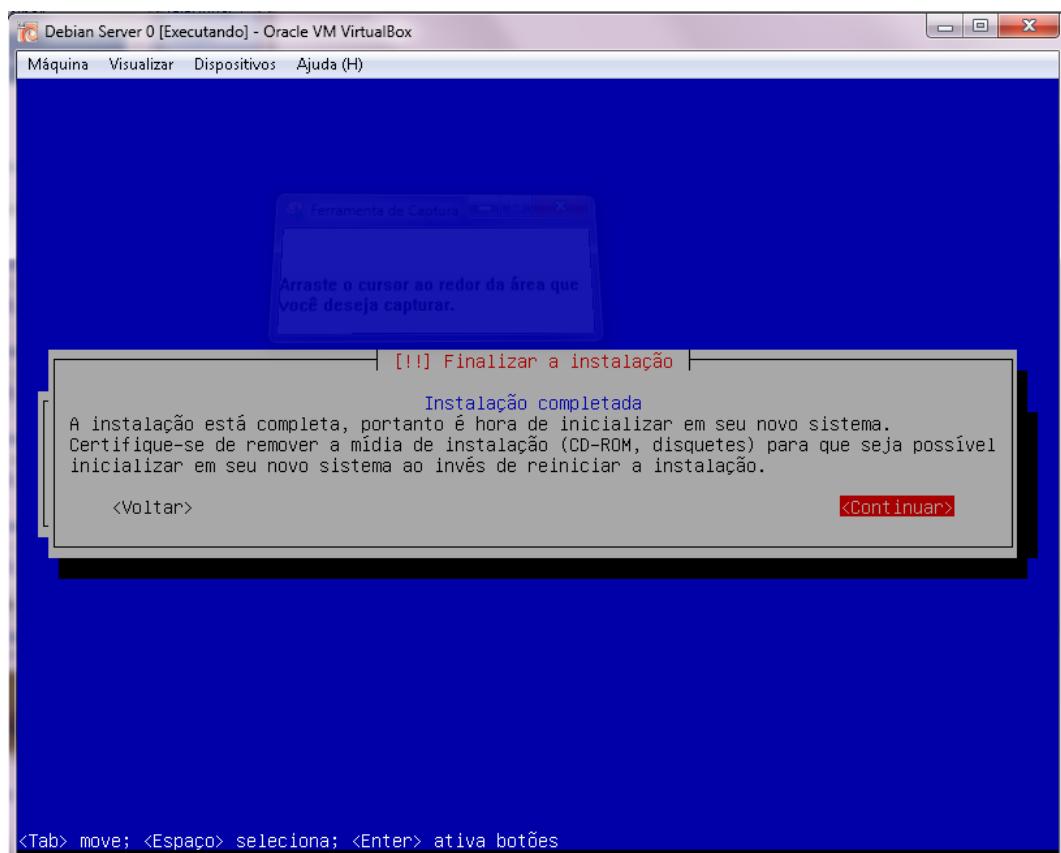




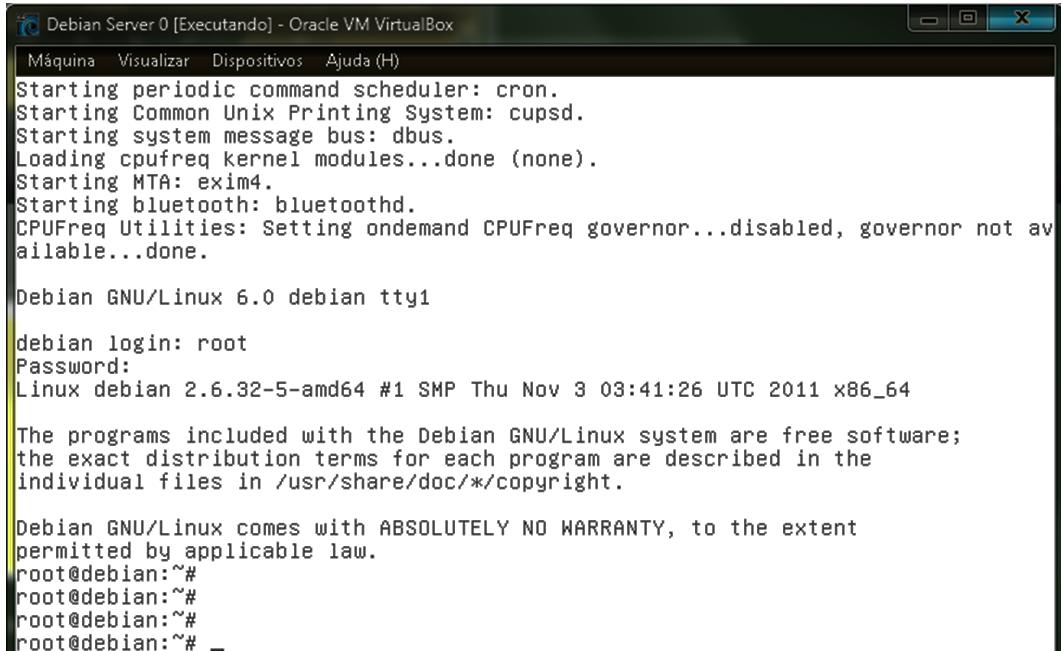
Depois de um tempão para a instalação do sistema básico, chega a tela onde se escolhe o perfil de instalação a ser utilizado. Vou deixar o sistema “cru” porque acho que o Cloud Server vai estar assim também.



Espera mais alguns minutos pra ele baixar e configurar os pacotes que faltam.



A máquina reinicia, e está pronto, um ambiente de testes para a configuração do Cloud Server:



```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
Starting periodic command scheduler: cron.
Starting Common Unix Printing System: cupsd.
Starting system message bus: dbus.
Loading cpufreq kernel modules...done (none).
Starting MTA: exim4.
Starting bluetooth: bluetoothd.
CPUFreq Utilities: Setting ondemand CPUFreq governor...disabled, governor not available...done.

Debian GNU/Linux 6.0 debian tty1

debian login: root
Password:
Linux debian 2.6.32-5-amd64 #1 SMP Thu Nov 3 03:41:26 UTC 2011 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~#
root@debian:~#
root@debian:~#
root@debian:~#
```

### 3.2.1. Aviso para login

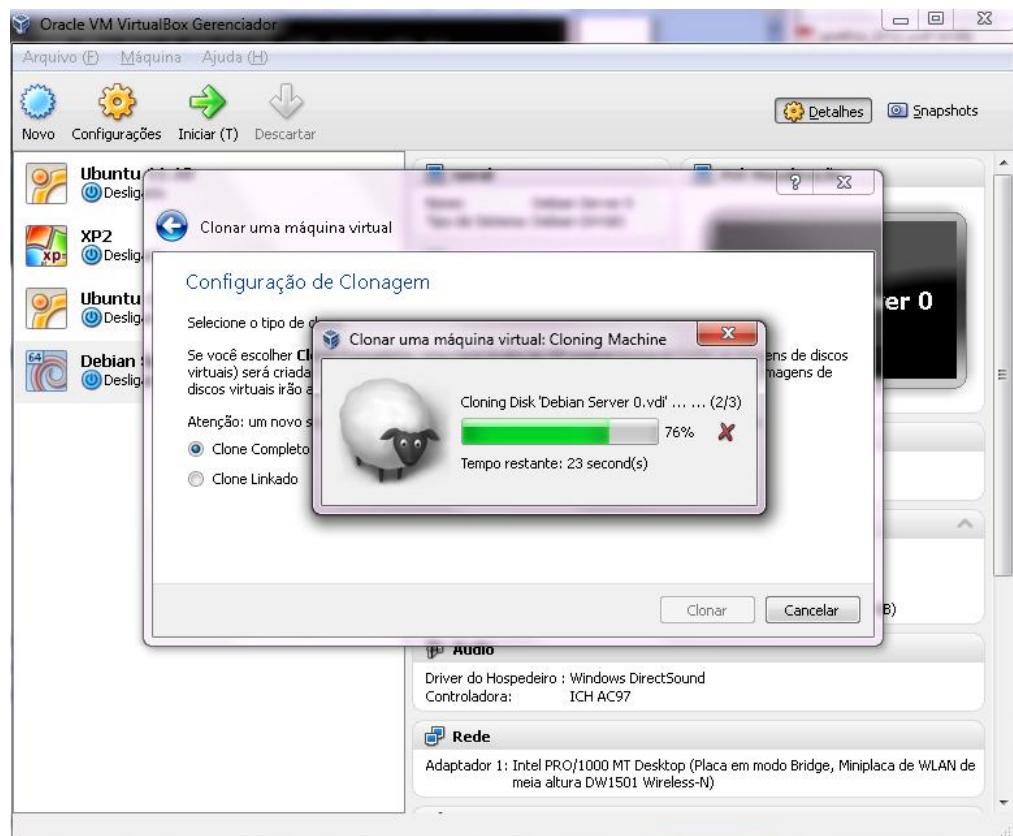
Para quem nunca chegou nesse ponto (sistema operacional Linux recém-instalado, operando em modo texto puro), pra fazer login tem que usar o usuário “root” e a senha de root que foi configurada na instalação.

Ok, não precisaria ser o usuário root, poderia ser o usuário adicional que também foi configurado na instalação; mas isso exigiria alguns ajustes na hora de rodar os comandos de instalação que vamos precisar mais à frente. Além do que, é uma máquina virtual, se ela parar de funcionar poderemos recomeçar a instalação do zero.

## 3.3. Clonando a Máquina-Base

Agora que a máquina-base está pronta (depois de uma hora!) a próxima coisa a fazer é criar um clone, através do menu do próprio Virtual-Box. Acho que essa é uma boa idéia, porque se eu fizer alguma caca daqui pra frente(o que é bastante provável) não preciso perder tempo criando uma máquina virtual do zero. É mais ou menos como se fizesse uma imagem (ghost) de uma máquina real.

Depois de desligar a máquina (com o comando ‘halt’), no menu do Virtual Box, clica em Máquina/Clonar. Aí é só seguir as instruções do assistente. Tarefa absurdamente simples, a clonagem demora uns 5 minutos.



## 4. Procedimentos de Instalação

Se a máquina estivesse pronta antes, ou tivesse sido instalada sem conexão de internet, seria o caso de rodar o comando “`apt-get update`” agora, mas como a instalação detectou a minha conexão de internet automaticamente, ela já fez isso por mim.

Com a lista do `apt-get` atualizada, poderia instalar o ruby e o rails simplesmente digitando:

```
apt-get install rails
```

Isso instalaria tudo o que é necessário para ter o Rails rodando na máquina. Mas que graça teria?

Então optei por um método um pouco mais “hard” (ou nerd, sei lá), fazendo a instalação dos pacotes principais à mão, deixando para utilizar o `apt-get` nos casos em que me pareceu mais apropriado (principalmente bibliotecas e outras dependências para os pacotes principais).

Seguindo as instruções no site RubyOnRails.org, preciso instalar o ruby 1.9.2, depois o ruby-gems, e só depois o Rails. Senão, vejamos:

[rubyonrails.org/download](http://rubyonrails.org/download)

orkut - Fórum Nova guia Google orkut - [Receita Fina... Sabrina Jóias RubyOnRails Facebook Antena 1 - O melhor...

[Overview](#) | [Download](#) | [Deploy](#) | [Bugs/Patches](#) | [Screencasts](#) | [Documentation](#) | [Ecosystem](#) | [Community](#) | [Blog](#)

## Get Ruby on Rails in no time



Rails is low on dependencies and prides itself on shipping with most everything you need in the box. To get started, just install Ruby, the language, and RubyGems, the package manager.

**Ruby** We recommend Ruby 1.8.7 or Ruby 1.9.2 for use with Rails. Ruby 1.8.6 and earlier are not supported, neither is version 1.9.1.



[Source: Compile it yourself](#)  
[Windows installer: Ruby, popular extensions, editor](#)  
[OS X 10.5+: Included with developer tools \(then do "gem update rails"\)](#)

**RubyGems** [RubyGems](#) is the standard Ruby package manager. It's similar to apt-get, emerge, and other OS package managers.



[Download](#)  
(extract, then run "ruby setup.rb")

**Rails** With RubyGems loaded, you can install all of Rails and its dependencies through the command line:

**gem install rails**

New versions of Rails can be installed the same way. Rails can also be downloaded as [stand-alone packages](#).

### 4.1.1. Pré-Instalação

Diversas bibliotecas e ferramentas serão necessárias para seguir os passos seguintes. Pelo que verifiquei, vai funcionar tudo bem desde que você execute as seguintes instalações

```
apt-get install zlib1g zlib1g-dev3
apt-get install libreadline5-dev4
```

---

<sup>3</sup> <http://stackoverflow.com/questions/769496/ubuntu-noob-rails-install-fails-on-zlib>. Se você pular este item, será forçado a executá-lo quando chegar na instalação do Rails, e também terá que repetir a instalação do Ruby.

```
apt-get install libopenssl-ruby  
apt-get install libapr1-dev  
apt-get install libaprutil1-dev  
apt-get install build-essential  
apt-get install libcurl4-openssl-dev
```

## 4.2. SSH

Para este tutorial, usei os comando digitando diretamente no console da máquina virtual. Depois de tudo, me ocorreu que talvez seria uma boa idéia utilizar uma conexão SSH para isso. Basicamente, trata-se de instalar o serviço na máquina virtual:

```
apt-get install ssh
```

E depois, utilizar um cliente ssh qualquer (eu gosto do putty<sup>5</sup>). É claro que você ainda teria que deixar a máquina virtual ligada para isso. Então, qual seria a vantagem? Simples, na janela do cliente ssh, como é um aplicativo do Windows e não da máquina virtual, você poderia dar `ctrl+c` nos comandos deste tutorial, e colar na janela do cliente ssh clicando o botão direito do mouse (`ctrl+v` não funciona, pelo menos não no putty).

## 4.3. Instalando Ruby 1.9.2

O link “source” na página acima inicia direto o download do Ruby na versão 1.8.7. Acontece que eu quero a versão 1.9.2. Por que, se a 1.8.7 é a recomendada

---

<sup>4</sup> Necessário para a instalação do PostgreSQL: <http://old.nabble.com/Problemas-ao-compilar-o-PostgreSQL-td15203771.html>

<sup>5</sup> <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

pelo próprio site? Porque não tem nenhum cliente me cobrando nesse projeto, então posso me dar ao luxo de “viver perigosamente”.<sup>6</sup>

De onde, abri o site [www.ruby-lang.org/pt](http://www.ruby-lang.org/pt) (aliás, um belo site, agradável como o ruby, merece um print-screen):

The screenshot shows the official Ruby website at [www.ruby-lang.org/pt](http://www.ruby-lang.org/pt). The header features the Ruby logo (a red gem) and the tagline "Ruby A Programmer's Best Friend". The navigation bar includes links for Downloads, Documentação, Módulos, Comunidade, Notícias, and Sobre Ruby. The main content area has a dark blue sidebar on the left containing text about Ruby's simplicity and productivity, a link to "Ler Mais...", and a "Ruby é..." section. The right side of the page contains several boxes: one showing Ruby code for printing city names; a "Download Ruby" button; a "Primeiros passos, é fácil!" section with links to Try Ruby!, Ruby em Vinte Minutos, and Ruby a partir de outras linguagens; an "Explora um novo mundo..." section with links to Documentação, Livros, Bibliotecas, and Histórias de sucesso; a "Participa numa comunidade simpática e em crescimento." section with links to Listas de Correio, Grupos de Utilizadores, and Núcleo de Ruby; and a "Top de projectos de Ruby" section with a "Saber mais..." link. The footer contains a "Sobre este sitio" link.

<sup>6</sup> Atualização para a versão 1.0: Na verdade, quando fui instalar o Passenger, descobri que é recomendado no site da Phusion que seja utilizado o Ruby Enterprise Edition versão 1.8.7; diz lá que isso pode economizar até 33% de memória do servidor. No entanto, em outros locais na web, li referências de que o Ruby 1.9.2 é mais eficiente que o 1.8.7 para rodar as aplicações Rails. Então fica a seu critério qual versão usar, estou disponibilizando um tutorial para cada um em [www.adstape.com/tutoriais](http://www.adstape.com/tutoriais). No meu servidor para a aplicação [www.adstape.com](http://www.adstape.com), optei por utilizar o Ruby Enterprise 1.8.7. O mais importante é que a versão do Ruby utilizada na máquina de desenvolvimento deve ser a mesma utilizada no servidor que irá hospedar a aplicação.

Clicando no link “Download Ruby” no alto à direita, caí em uma página com um link para fazer o download do pacote da versão 1.9.2p290: <http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz>

Na máquina virtual, criei uma pasta “install” (santa criatividade!). Entrei na pasta, e fiz o download via wget:

```
wget http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz
```

Aqui o download demorou uns 20 minutos, mas acho que foi culpa do speedy, são só 11MB de arquivo.

```
Máquina Visualizar Dispositivos Ajuda (H)
root@debian:~# pwd
/root
root@debian:~# ls
root@debian:~# mkdir install
root@debian:~# ls
install
root@debian:~# cd install
root@debian:~/install# wget http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz
--2011-12-08 22:14:55--  http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz
Resolvendo ftp.ruby-lang.org... 221.186.184.68
Conectando-se a ftp.ruby-lang.org|221.186.184.68|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 11182217 (11M) [application/x-tar]
Salvando em: "ruby-1.9.2-p290.tar.gz"

100%[=====] 11182217  9,42K/s   em 18m 21s
2011-12-08 22:33:17 (9,92 KB/s) - "ruby-1.9.2-p290.tar.gz" salvo [11182217/11182217]

root@debian:~/install# ls
ruby-1.9.2-p290.tar.gz
root@debian:~/install# _
```

Próximo passo, descompactar o arquivo com o comando tar

```
tar -zxvf ruby-1.9.2-p290.tar.gz
```

Agora dentro da pasta instal tem a pasta ruby-1.9.2-p290. A instalação segue o padrão de 3 passos dos códigos-fonte Linux: a) ./configure b) make c) make install

Primeiro, entra na pasta, depois digita os comandos

```
cd ruby-1.9.2-p290
./configure
```

(espera uns dois minutos)

```
make
```

(espera um tempão)

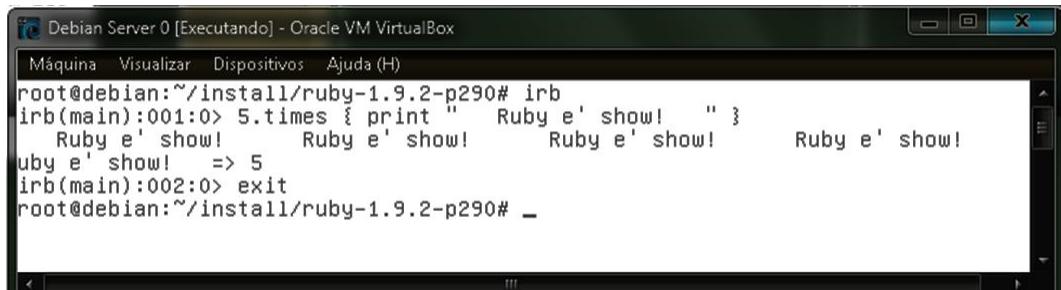
```
make install
```

(espera mais uns segundos).

Ruby ok, testar abrindo o irb

```
irb
```

```
5.times { print "    Ruby e' show!    " }
```



```
Máquina Visualizar Dispositivos Ajuda(H)
root@debian:~/install/ruby-1.9.2-p290# irb
irb(main):001:0> 5.times { print "    Ruby e' show!    " }
Ruby e' show!    Ruby e' show!    Ruby e' show!    Ruby e' show!
uby e' show! => 5
irb(main):002:0> exit
root@debian:~/install/ruby-1.9.2-p290# _
```

A máquina virtual não está aceitando acentos na digitação, mas isso não faz diferença.

Tudo OK até aqui, hora de desligar com 'halt' e fazer outro clone da Máquina Virtual.

#### 4.3.1. Compilando manualmente o módulo OpenSSL

Me pareceu muito estranho, mas na hora de instalar o Passenger lá na frente, ele dava por falta do módulo SSL do Ruby, e a instrução de instalação na funcionava.

Achei a solução aqui:  
<http://comments.gmane.org/gmane.comp.lang.ruby.rails/277677> e aqui:  
<http://askubuntu.com/questions/66934/passenger-mod-rails-cant-find-libopenssl-ruby>

É necessário compilar manualmente o módulo de ssl do ruby. Então voltei para a pasta onde está o código-fonte que usei na instalação, e executei a seqüência:

```
cd /root/install/ruby-1.9.2-p290
cd ext/openssl
ruby extconf.rb
```

```
make  
make install
```

#### 4.4. Instalando o RubyGems

RubyGems é o gerenciador de pacotes do Ruby (mais ou menos como o apt-get é para o Debian). Através dele o Rails e outras extensões podem ser instaladas com o comando “gem install alguma\_coisa” (que é o que vou usar mais pra frente).

Então, seguindo as instruções do site <http://rubyonrails.org/download>, cliquei lá no link para o RubyGems, que levou para o endereço <http://docs.rubygems.org> , onde cliquei no alto no link de downloads, o que me levou para esta página:

The screenshot shows a web browser window with the RubyForge website. The URL in the address bar is rubyforge.org/frs/?group\_id=126. The page title is "RUBYFORGE by Ruby Central". The navigation menu includes Home, My Page, Project Tree, Code Snippets, Projectos que Precisam de Ajuda, and RubyGems (which is highlighted). Below the menu, there are tabs for Resumo, Listas, and Files (which is also highlighted). The main content area is titled "Files | Admin". It displays a table of files for the "rubygems" package. The table has columns for Package, Release & Notes, Filename, Date, Size, D/L, Arch, and Type. The table shows several versions of the rubygems package, with the latest version being 1.8.12, released on 2011-12-02 at 14:48. Other versions shown include 1.8.11, 1.8.10, and 1.8.9.

Package	Release & Notes	Filename	Date	Size	D/L	Arch	Type
1.8.12	<a href="#">rubygems-1.8.12.tgz</a>		2011-12-02 14:48	245 KB	1,933 Any	Source	.zip
	<a href="#">rubygems-1.8.12.zip</a>			331 KB	2,387 Any		.zip
	<a href="#">rubygems-update-1.8.12.gem</a>			251 KB	652 Any		.gem (RubyGem)
1.8.11	<a href="#">rubygems-1.8.11.tgz</a>		2011-11-03 11:06	420 KB	8,424 Any	Source	.zip
	<a href="#">rubygems-1.8.11.zip</a>			669 KB	11,217 Any		.zip
	<a href="#">rubygems-update-1.8.11.gem</a>			250 KB	2,958 Any		.gem (RubyGem)
1.8.10	<a href="#">rubygems-1.8.10.tgz</a>		2011-08-26 17:50	244 KB	36,956 Any	Source	.zip
	<a href="#">rubygems-update-1.8.10.gem</a>			250 KB	12,425 Any		.gem (RubyGem)
1.8.9	<a href="#">rubygems-1.8.9.tgz</a>		2011-08-23 15:38	243 KB	2,084 Any	Source	.zip
	<a href="#">rubygems-1.8.9.zip</a>			220 KB	0 Any		.zip

De onde capturei o link para o rubygems-1.8.12.tgz:  
<http://rubyforge.org/frs/download.php/75573/rubygems-1.8.12.tgz>

Na máquina virtual, voltei para a pasta install com o comando “cd ..” (estava na subpasta ruby-1.9.2p290), e fiz o download do rubygems:

```
wget http://rubyforge.org/frs/download.php/75573/rubygems-1.8.12.tgz
```

```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
root@debian:~/install1/ruby-1.9.2-p290# cd ..
root@debian:~/install1# wget http://rubyforge.org/frs/download.php/75573/rubygems-1.8.12.tgz
--2011-12-08 23:01:59--  http://rubyforge.org/frs/download.php/75573/rubygems-1.8.12.tgz
Resolvendo rubyforge.org... 205.234.109.19
Conectando-se a rubyforge.org|205.234.109.19|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 302 Found
Localização: http://files.rubyforge.vm.bytemark.co.uk/rubygems/rubygems-1.8.12.tgz [redirecionando]
--2011-12-08 23:02:00--  http://files.rubyforge.vm.bytemark.co.uk/rubygems/rubygems-1.8.12.tgz
Resolvendo files.rubyforge.vm.bytemark.co.uk... 80.68.94.54
Conectando-se a files.rubyforge.vm.bytemark.co.uk|80.68.94.54|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 250496 (245K) [application/x-gzip]
Salvando em: "rubygems-1.8.12.tgz"

100%[=====] 250496      11,8K/s   em 13s
2011-12-08 23:02:14 (19,1 KB/s) - "rubygems-1.8.12.tgz" salvo [250496/250496]

root@debian:~/install1# ls
ruby-1.9.2-p290  ruby-1.9.2-p290.tar.gz  rubygems-1.8.12.tgz
root@debian:~/install1# _
```

### Sequência para instalação

```
tar -zxvpf rubygems-1.8.12.tgz
cd rubygems-1.8.12
ruby setup.rb
```

```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
rubygems-1.8.12/bin/update_rubygems
root@debian:~/install1# ls
ruby-1.9.2-p290  ruby-1.9.2-p290.tar.gz  rubygems-1.8.12  rubygems-1.8.12.tgz
root@debian:~/install1# cd rubygems-1.8.12/
root@debian:~/install1/rubygems-1.8.12# ls
bin          History.txt  MIT.txt    test
ci_build.sh    lib        Rakefile   UPGRADING.rdoc
cruise_config.rb LICENSE.txt  README.rdoc util
hide_lib_for_update Manifest.txt  setup.rb
root@debian:~/install1/rubygems-1.8.12# ruby setup.rb
RubyGems 1.8.12 installed

== 1.8.12 / 2011-12-02

* Bug fix:
  * Handle more cases where Syck's DefaultKey showed up in requirements
    and wasn't cleaned out.

-----
RubyGems installed the following executables:
  /usr/local/bin/gem

root@debian:~/install1/rubygems-1.8.12# _
```

Aparentemente está tudo ok até aqui. A instalação do RubyGems foi super-rápida e simples, teoricamente nem precisaria fazer um clone da máquina agora, mas já que defini o padrão, vou fazer mesmo assim.

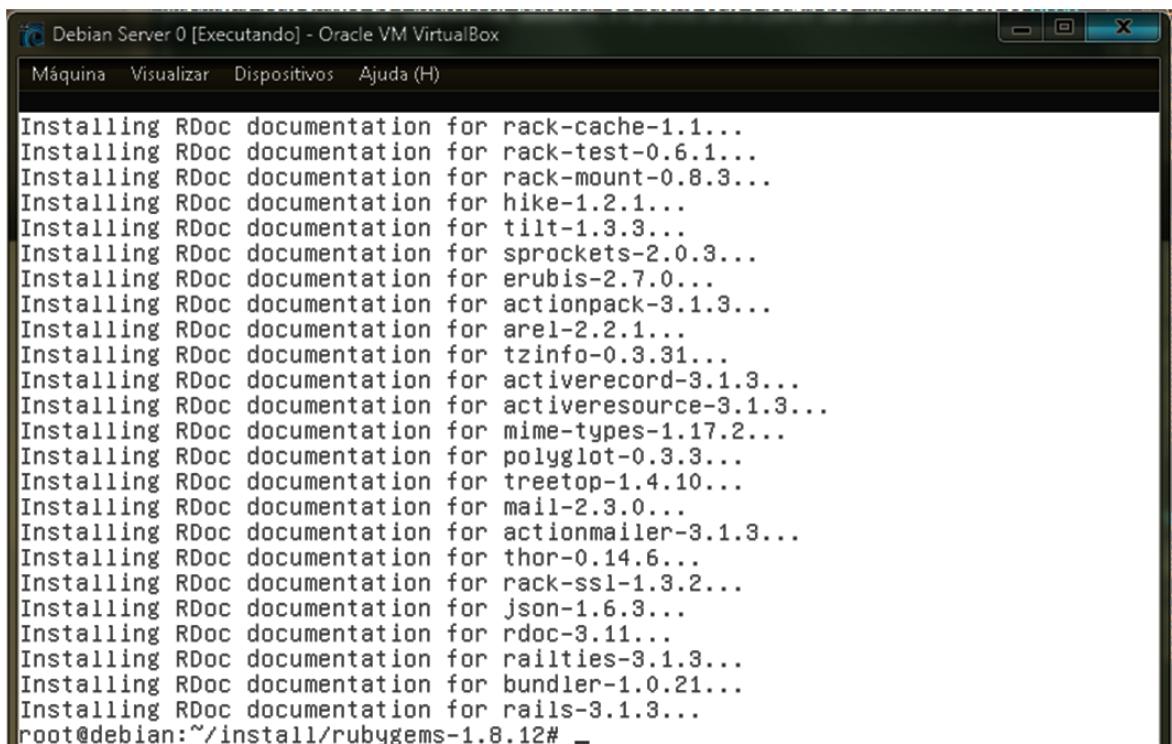
Desliga a máquina, faz um clone, liga a máquina de novo.

## 4.5. Instalar o Rails

A partir daqui as coisas podem começar a quebrar. Tecnicamente, bastaria digitar o comando a seguir, nem precisaria estar na pasta install

```
gem install rails -v 3.1.3
```

(o `-v 3.1.3` é necessário apenas para garantir que será instalada essa versão)



The screenshot shows a terminal window titled "Debian Server 0 [Executando] - Oracle VM VirtualBox". The window has a menu bar with "Máquina", "Visualizar", "Dispositivos", and "Ajuda (H)". The main area of the terminal displays the following text:

```
Installing RDoc documentation for rack-cache-1.1...
Installing RDoc documentation for rack-test-0.6.1...
Installing RDoc documentation for rack-mount-0.8.3...
Installing RDoc documentation for hike-1.2.1...
Installing RDoc documentation for tilt-1.3.3...
Installing RDoc documentation for sprockets-2.0.3...
Installing RDoc documentation for erubis-2.7.0...
Installing RDoc documentation for actionpack-3.1.3...
Installing RDoc documentation for arel-2.2.1...
Installing RDoc documentation for tzinfo-0.3.31...
Installing RDoc documentation for activerecord-3.1.3...
Installing RDoc documentation for activeresource-3.1.3...
Installing RDoc documentation for mime-types-1.17.2...
Installing RDoc documentation for polyglot-0.3.3...
Installing RDoc documentation for treeTop-1.4.10...
Installing RDoc documentation for mail-2.3.0...
Installing RDoc documentation for actionmailer-3.1.3...
Installing RDoc documentation for thor-0.14.6...
Installing RDoc documentation for rack-ssl-1.3.2...
Installing RDoc documentation for json-1.6.3...
Installing RDoc documentation for rdoc-3.11...
Installing RDoc documentation for railties-3.1.3...
Installing RDoc documentation for bundler-1.0.21...
Installing RDoc documentation for rails-3.1.3...
root@debian:~/install/rubygems-1.8.12# _
```

Se isso não funcionar desta forma para você, é provável que você tenha pulado a seção “Pré-instalação”, onde dizia que deveria instalar a biblioteca zlib1g-dev.

Desligar a máquina virtual, gerar clone, religar a máquina.

## 4.6. PostGreSql 9.1.2

Esse também poderia fazer com apt-get...

Mas vamos continuar no mesmo ritmo, e instalar pelo código-fonte.

Abrindo o site [www.postgresql.org](http://www.postgresql.org), e clicando no link download, no centro da página tem o link “source code”, que aponta para <http://www.postgresql.org/ftp/source/>. Lá tem uma lista de pastas com todas as versões disponíveis do postgresql. Abrindo a pasta v.1.9.2:



De onde capturei o link <http://ftp.postgresql.org/pub/source/v9.1.2/postgresql-9.1.2.tar.gz>

Então, na máquina virtual, que acabei de religar:

```
cd /root/install  
wget http://ftp.postgresql.org/pub/source/v9.1.2/postgresql-9.1.2.tar.gz
```

Olhando a página <http://www.postgresql.org/docs/9.1/static/install-short.html>,

The screenshot shows a web browser window with the URL [www.postgresql.org/docs/9.1/static/install-short.html](http://www.postgresql.org/docs/9.1/static/install-short.html). The page title is "PostgreSQL". The main content is titled "PostgreSQL 9.1.2 Documentation" and "Chapter 15. Installation from Source Code". A section titled "15.1. Short Version" contains a command-line sequence:

```
./configure  
gmake  
su  
gmake install  
adduser postgres  
mkdir /usr/local/pgsql/data  
chown postgres /usr/local/pgsql/data  
su - postgres  
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data  
/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &  
/usr/local/pgsql/bin/createdb test  
/usr/local/pgsql/bin/psql test
```

Below the command-line, a note says "The long version is the rest of this chapter." Navigation links include "Prev", "Up", "Home", "Next", and "Requirements".

Diz lá que é para compilar usando gmake ao invés de make... bom, eu usei make mesmo aqui e deu certo.

Continuando, descompacta o código-fonte com tar, entra na pasta, e depois roda a sequência de comandos que o manual acima mandou:

```
tar -zxvpf postgresql-9.1.2.tar.gz  
cd postgresql-9.1.2  
.configure  
make  
make install  
adduser postgres  
mkdir /usr/local/pgsql/data  
chown postgres /usr/local/pgsql/data  
su - postgres  
/usr/local/pgsql/bin/initdb -D /user/local/pgsql/data  
/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &  
/usr/local/pgsql/bin/createdb test
```

```
/usr/local/pgsql/bin/psql test  
su  
cd /root/install
```

#### 4.6.1. Pós-instalação : Script de Inicialização

Os comandos para parar/iniciar o postgres são bem longos, exigem mudar de usuário, digitar um comando complexo e retornar para o usuário root. Se tivesse instalado via apt-get, poderia utilizar simplesmente:

```
/etc/init.d/postgresql stop  
/etc/init.d/postgresql start
```

Na instalação manual que foi feita esse arquivo “postgresql” não vai para a pasta /etc/init.d, mas descobri aqui [http://www.slackware-brasil.com.br/web\\_site/artigos/artigo\\_completo.php?aid=3552](http://www.slackware-brasil.com.br/web_site/artigos/artigo_completo.php?aid=3552) que esse arquivo vem junto com o código fonte que foi baixado, com o nome “linux”. Então

```
cd /root/install  
cp postgresql-9.1.2/contrib/start-scripts/linux  
/etc/init.d/postgresql7  
chmod 777 /etc/init.d/postgresql
```

Agora o arquivo de log passa a ser gravado na pasta /usr/local/pgsql/data, com o nome de serverlog.

---

<sup>7</sup> Este comando é todo em uma única linha, mesmo que o console da máquina virtual quebre em mais de uma. Aqui está assim por causa das limitações do editor de texto utilizado (ou porque talvez eu não saiba utilizá-lo direito).

```
Máquina Visualizar Dispositivos Ajuda (H)
root@debian:~/install# /etc/init.d/postgresql stop
Stopping PostgreSQL: ok
root@debian:~/install# /etc/init.d/postgresql start
Starting PostgreSQL: ok
root@debian:~/install# cat /usr/local/pgsql/data/serverlog
LOG:  database system was shut down at 2011-12-11 17:08:00 BRST
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
LOG:  database system was interrupted; last known up at 2011-12-11 17:18:00 BRST
LOG:  database system was not properly shut down; automatic recovery in progress
LOG:  consistent recovery state reached at 0/16BD3F8
LOG:  record with zero length at 0/16BD3F8
LOG:  redo is not required
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
LOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
LOG:  shutting down
LOG:  database system is shut down
LOG:  database system was shut down at 2011-12-11 17:25:08 BRST
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
root@debian:~/install# _
```

Agora, para que o serviço seja iniciado sempre que a máquina for reiniciada, só precisa editar o arquivo `/etc/init.d/rc.local` e colocar no final dele a linha

```
/etc/init.d/postgresql start
```

#### 4.6.2. Pós-instalação: Localização do PID

Quando estiver colocando a aplicação para rodar, o Rails irá procurar o arquivo de pid (arquivo de identificação do processo) do PostgreSQL na pasta `/var/run/postgresql`, com o nome de “`.s.PGSQL.5432`”

De onde, vamos editar o arquivo<sup>8</sup> `/usr/local/pgsql/data/postgresql.conf` e definir que o caminho para o pid, colocando lá a linha para definir isso:

```
vi /usr/local/pgsql/data/postgresql.conf
```

Abre o editor, depois basta localizar a seção correta e inserir a linha:

```
external_pid_file = '/var/run/postgresql/.s.PGSQL.5432'
```

---

<sup>8</sup> Para quem não está acostumado com Linux em modo texto, edição de arquivo é feita geralmente com o uso do editor “`vi`”, e à primeira vista ele é um tanto enigmático. Um guia de referência rápida pode ser encontrado aqui: [http://www.primeirospassos.org/sessao3\\_4.html](http://www.primeirospassos.org/sessao3_4.html)



```

Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
#-----#
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
#data_directory = 'ConfigDir'          # use data in another directory
#                                     # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'    # host-based authentication file
#                                     # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
#                                     # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'          # write an extra PID file
#                                     # (change requires restart)
external_pid_file = '/var/run/postgresql/.s.PGSQL.5432'

#-----#
# CONNECTIONS AND AUTHENTICATION
#-----#
# - Connection Settings -
"/usr/local/pgsql/data/postgresql.conf" 556 lines, 19227 characters written
root@debian:/home/rails_app/adsitape#_

```

Depois salva o arquivo e fecha o editor com o comando “:wq”.

Agora precisa criar a pasta /var/run/postgresql, tornar o usuário postgres dono dela, e dar acesso de leitura para os outros usuários; depois reiniciar o serviço:

```

mkdir /var/run/postgresql

chown postgres /var/run/postgresql -R

chmod 755 /var/run/postgresql -R

/etc/init.d/postgresql restart

ls /var/run/postgresql -all

```

E no final verificamos que o arquivo de identificação do processo está onde o Rails pretenderá encontrá-lo.

Desligar máquina virtual, clonar, religar máquina

## 4.7. Apache

Outro que podia ser por apt-get, mas... Principalmente este não gosto de instalar com apt-get, nenhum motivo importante, questão de gosto mesmo. Na verdade acho que é só para poder escolher o diretório de instalação, opção que não tem quando instala via apt-get (se tem, eu não sei como usar).

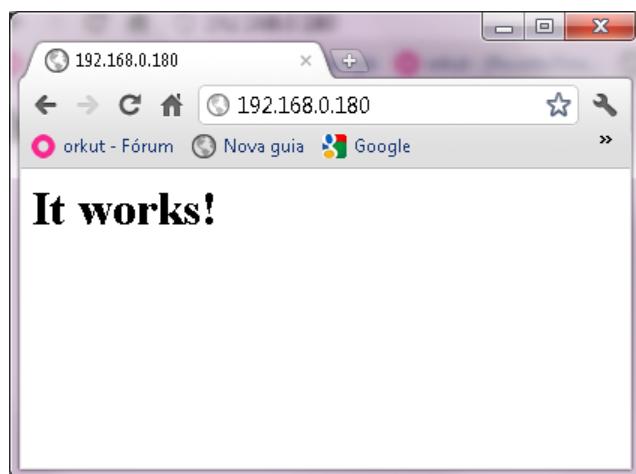
Abrindo a página [www.apache.org](http://www.apache.org), clica no link Download no alto à direita, abre uma página que no meu caso sugeriu o download através do mirror da Unicamp. Clicando no link sugerido, abre uma página com dezenas de pastas (todos os projetos de software da Fundação Apache), o que me interessa é a pasta “httpd”, que é o web-server propriamente dito. Clicando nela, abre a lista de arquivos disponíveis, de onde capturo o link para o código fonte da versão 2.2.21 <http://ftp.unicamp.br/pub/apache//httpd/httpd-2.2.21.tar.gz> Achei muito estranho aquela barra dupla entre apache e httpd, mas é assim mesmo.

```
cd install  
wget http://ftp.unicamp.br/pub/apache//httpd/httpd-2.2.21.tar.gz  
tar -zxvpf httpd-2.2.21.tar.gz  
dd httpd-2.2.21  
.configure --prefix=/apache2  
make  
make install
```

Tem que iniciar o serviço manualmente:

```
/apache2/bin/apachectl start
```

Para testar, na máquina real (porque a máquina virtual não tem um navegador gráfico, aliás não tem nada gráfico), vai no navegador de internet e tenta acessar <http://192.168.0.180> (porque esse é o ip da minha máquina virtual).



A máquina real consegue acessar o servidor web, que devolve o conteúdo do arquivo /apache2/htdocs/index.html.

Agora só precisa colocar o comando de inicialização do serviço no final do arquivo /etc/init.d/rc.local (para que seja iniciado automaticamente toda vez que a máquina for reiniciada).

## 4.8. Passenger

Ao dar uma olhada no site [www.modrails.com](http://www.modrails.com) (um site muito bonito, aliás, compete com o [www.ruby-lang.org](http://www.ruby-lang.org), também merece um print-screen)



Então, dando uma olhada por lá, descobri que ao invés do Ruby 1.9.2, o ideal seria usar o ruby enterprise edition, que só tem até a versão 1.8.7 (ooohhh... mas o site do rubyonrails.org sugeria isso, quem mandou eu ser teimoso e optar pela versão 1.9.2...)

O que diz lá é que com o Ruby Enterprise, o Passenger pode usar até 33% a menos de memória do que com outra versão do Ruby.

E agora? Recomeço tudo? Não. Vou continuar aqui com a versão 1.9.2 mesmo (sou teimoso), até porquê diz em alguns outros lugares que o Ruby 1.9.2 é mais eficiente que o 1.8.7.

#### 4.8.1. Instalando

A página de instalação é muito bonita, e as instruções são bem simples:

```
gem install passenger  
passenger-install-apache2-module
```

Este último comando inicia o instalador. Ele é muito legal: checa todas as dependências, e se estiver faltando alguma, ele te diz como instalar.

#### 4.8.2. Variável APXS2

Aqui o instalador disse que não estava instalado o Apache.

Dei uma olhada na documentação disponível no site, [http://www.modrails.com/documentation/Users%20guide%20Apache.html#\\_generic\\_installation\\_instructions](http://www.modrails.com/documentation/Users%20guide%20Apache.html#_generic_installation_instructions) e descobri que o passenger espera encontrar o Apache na pasta padrão. Mais especificamente, ele esperar encontrar o binário apxs na pasta /opt/apache2/bin.

Como instalei o apache em outra pasta, diferente da padrão, aqui o apxs está na pasta /apache2/bin. Seguindo a instrução no site, depois de verificar se eu realmente tinha o binário apxs com o comando

```
ls /apache2/bin
```

exportei uma variável de ambiente para que o passenger possa fazer a localização correta.

```
export APXS2=/apache2/bin/apxs
```

Importante: tem que adicionar essa mesma linha acima no final do arquivo /etc/init.d/rc.local, para que a variável seja criada novamente quando a máquina for reiniciada.

#### 4.8.3. Continuando

```
passenger-install-apache2-module
```

Encontrou todas as dependências e iniciou a instalação, terminando nas seguintes telas:

```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
.9.1/ruby/backward -I/usr/local/include/ruby-1.9.1 -I/usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11/ext/ruby -DHAVE_ALLOCA_H -DHAVE_RBIO_H -fPIC -O3 -ggdb -Wextra -Wno-unused-parameter -Wno-parentheses -Wpointer-arith -Wwrite-stings -Wno-missing-field-initializers -Wno-long-long -g -o passenger_native_support.o -c /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11/ext/ruby/passenger_native_support.c
gcc -shared -o passenger_native_support.so passenger_native_support.o -L. -L/usr/local/lib -Wl,-R/usr/local/lib -L. -rdynamic -Wl,-export-dynamic

-----
The Apache 2 module was successfully installed.

Please edit your Apache configuration file, and add these lines:

LoadModule passenger_module /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11/ext/apache2/mod_passenger.so
PassengerRoot /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11
PassengerRuby /usr/local/bin/ruby

After you restart Apache, you are ready to deploy any number of Ruby on Rails applications on Apache, without any further Ruby on Rails-specific configuration!

Press ENTER to continue.
```

```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
Deploying a Ruby on Rails application: an example

Suppose you have a Rails application in /somewhere. Add a virtual host to your Apache configuration file and set its DocumentRoot to /somewhere/public:

<VirtualHost *:80>
  ServerName www.yourhost.com
  DocumentRoot /somewhere/public      # <-- be sure to point to 'public'!
  <Directory /somewhere/public>
    AllowOverride all                # <-- relax Apache security settings
    Options -MultiViews             # <-- MultiViews must be turned off
  </Directory>
</VirtualHost>

And that's it! You may also want to check the Users Guide for security and optimization tips, troubleshooting and other useful information:

  /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11/doc/Users guide Apache.html

Enjoy Phusion Passenger, a product of Phusion (www.phusion.nl) :-
http://www.modrails.com/

Phusion Passenger is a trademark of Hongli Lai & Ninh Bui.
root@debian:~/install/ruby-1.9.2-p290/ext/openssl# _
```

## 4.9. Colocando a aplicação Rails para rodar

As instruções do Passenger são claras e objetivas. Parece mesmo bem fácil, mas passei por uma boa quantidade de problemas nesta parte. Algumas já coloquei nas instruções acima de preparação do ambiente (como o arquivo de PID do PostgreSql, que consumiu umas duas horas pra ser resolvido). Como esta é a parte mais importante do tutorial, vou descrever os problemas e as soluções que funcionaram no meu caso<sup>9</sup> <sup>10</sup>.

### 4.9.1. Transferindo a aplicação para o servidor virtual

Minha aplicação Rails está em outra máquina virtual, rodando Ubuntu 11.10. Então, tenho que startar essa máquina, compactar o conteúdo da aplicação em um arquivo zip e trazê-lo para dentro da máquina Debian. Para fazer a transferência, vou usar scp, que realiza essa tarefa sobre o protocolo SSH, então tem que ter o serviço de ssh rodando na máquina Debian:

```
apt-get install ssh
```

Também é uma boa idéia criar na máquina Debian uma pasta base para a aplicação:

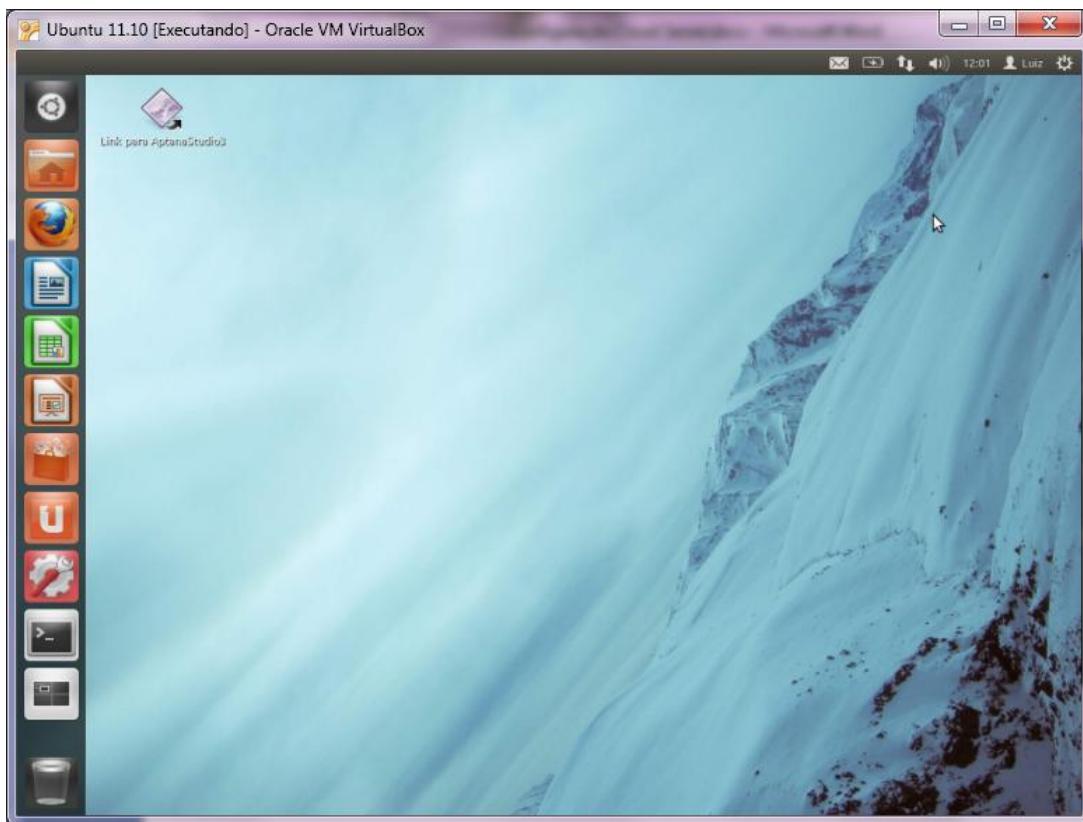
```
mkdir /home/rails_app
```

Abrindo a máquina virtual Ubuntu (sem desligar a máquina Debian):

---

<sup>9</sup> Em qualquer erro que seja encontrado, é importante lembrar de verificar os arquivos /apache2/logs/error\_log e, na pasta da aplicação o arquivo log/production.log (enquanto este último não existir, significa que o Apache não está passando requisições para o Passenger, ou este não está encontrando a aplicação Rails).

<sup>10</sup> Leia os avisos legais no Capítulo 1.



A maneira mais fácil de fazer a transferência é através de uma janela de terminal.

```
luiz@luiz-VirtualBox: ~/Aptana Studio 3 Workspace
luiz@luiz-VirtualBox:~/Aptana Studio 3 Workspace$ ls
adsitape ads teste sample_app
luiz@luiz-VirtualBox:~/Aptana Studio 3 Workspace$ zip -r adsitape.zip adsitape
adding: adsitape/ (stored 0%)
adding: adsitape/config.ru (deflated 15%)
adding: adsitape/README (deflated 59%)
adding: adsitape/config/ (stored 0%)
adding: adsitape/config/application.rb (deflated 51%)
adding: adsitape/config/environment.rb (deflated 29%)
adding: adsitape/config/database.yml (deflated 54%)
adding: adsitape/config/boot.rb (deflated 24%)
adding: adsitape/config/environments/ (stored 0%)
adding: adsitape/config/environments/production.rb (deflated 54%)
adding: adsitape/config/environments/test.rb (deflated 52%)
adding: adsitape/config/environments/development.rb (deflated 50%)
adding: adsitape/config/initializers/ (stored 0%)
adding: adsitape/config/initializers/backtrace_silencers.rb (deflated 42%)
adding: adsitape/config/initializers/wrap_parameters.rb (deflated 41%)
adding: adsitape/config/initializers/mime_types.rb (deflated 30%)
adding: adsitape/config/initializers/inflections.rb (deflated 39%)
adding: adsitape/config/initializers/secret_token.rb (deflated 32%)
adding: adsitape/config/initializers/session_store.rb (deflated 42%)
adding: adsitape/config/locales/ (stored 0%)
```

Desculpem, mas não consegui inverter as cores dessa imagem com o Gimp.

Usei o comando “cd pasta” para entrar na pasta acima de onde está a aplicação (que está na pasta adsitape). Depois o comando

```
zip -r adsitape.zip adsitape
```

O que criou o arquivo adsitape.zip contendo todo o conteúdo da pasta adsitape

Para fazer a transferência, preciso que as duas máquinas estejam na mesma faixa de endereço ip. Isso pode ser verificado digitando o comando

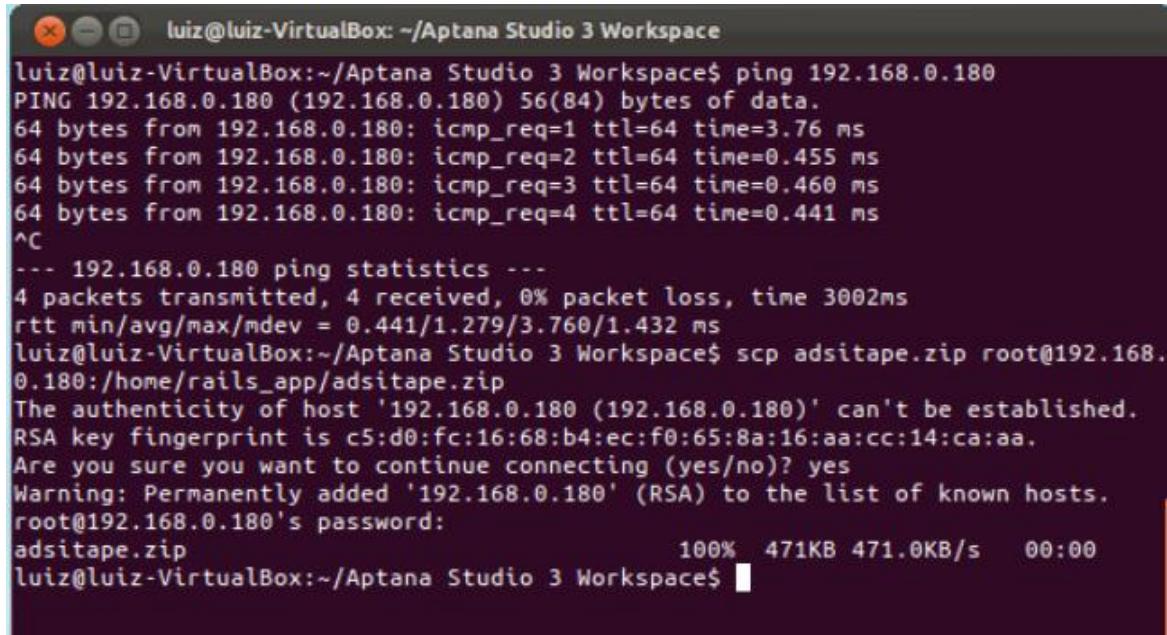
```
ifconfig
```

De onde verifiquei que a máquina Debian está com o ip 192.168.0.180, e a máquina Ubuntu está com o ip 192.168.0.186, ok. Se não estivesse, teria que configurar o ip de qualquer uma delas com o comando

```
sudo ifconfig eth1 192.168.0.xxx netmask 255.255.255.0
```

E depois transferir o arquivo com o comando:

```
scp adsitape.zip root@192.168.0.180:/home/rails\_app/adsitape.zip
```



```
luiz@luiz-VirtualBox:~/Aptana Studio 3 Workspace$ ping 192.168.0.180
PING 192.168.0.180 (192.168.0.180) 56(84) bytes of data.
64 bytes from 192.168.0.180: icmp_req=1 ttl=64 time=3.76 ms
64 bytes from 192.168.0.180: icmp_req=2 ttl=64 time=0.455 ms
64 bytes from 192.168.0.180: icmp_req=3 ttl=64 time=0.460 ms
64 bytes from 192.168.0.180: icmp_req=4 ttl=64 time=0.441 ms
^C
--- 192.168.0.180 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.441/1.279/3.760/1.432 ms
luiz@luiz-VirtualBox:~/Aptana Studio 3 Workspace$ scp adsitape.zip root@192.168.0.180:/home/rails_app/adsitape.zip
The authenticity of host '192.168.0.180 (192.168.0.180)' can't be established.
RSA key fingerprint is c5:d0:fc:16:68:b4:ec:f0:65:8a:16:aa:cc:14:ca:aa.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.180' (RSA) to the list of known hosts.
root@192.168.0.180's password:
adsitape.zip                                         100%  471KB 471.0KB/s   00:00
luiz@luiz-VirtualBox:~/Aptana Studio 3 Workspace$
```

Tranquilo até aqui.

Desliga a máquina Ubuntu, e na máquina Debian entra na pasta /home/rails\_app e descompacta o arquivo adsitape.zip

```
cd /home/rails_app
```

```
apt-get install unzip
```

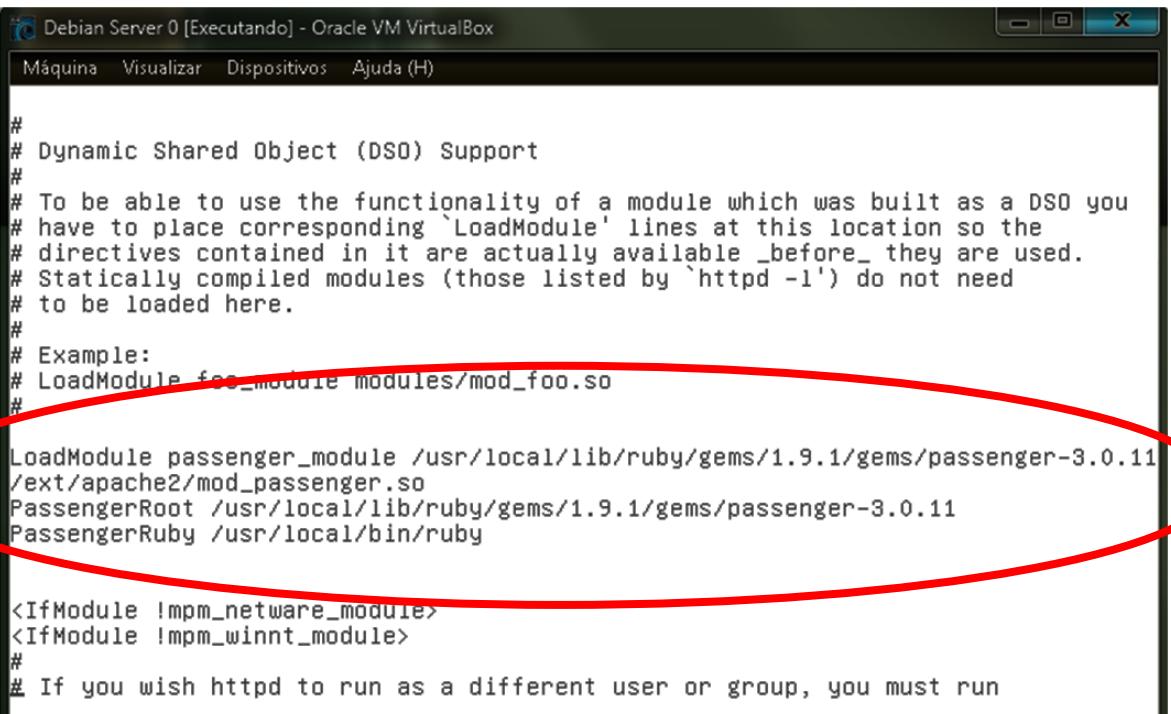
```
unzip adsitape.zip
```

O que vai criar a pasta /home/rails\_app/adsitape e dentro dela todo o conteúdo da aplicação Rails.

Agora tem que configurar o Apache para localizar e disponibilizar essa aplicação. Abri o arquivo httpd.conf

```
vi /apache2/conf/httpd.conf
```

Teclei `/LoadModule` para localizar a seção do arquivo que trata disso, e lá adicionei as linhas



```
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule passenger_module /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11
/ext/apache2/mod_passenger.so
PassengerRoot /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11
PassengerRuby /usr/local/bin/ruby

<IfModule !mpm_netware_module>
<IfModule !mpm_winnt_module>
#
# If you wish httpd to run as a different user or group, you must run
```

```
LoadModule passenger_module
/usr/local/lib/ruby/gems/1.9.1/gems/passenger-
3.0.11/ext/apache2/mod_passenger.so
```

(isso é tudo uma linha só, mesmo que o editor a quebre na visualização, não pode quebrar com retorno de linha (enter)).

```
PassengerRoot /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.11
PassengerRuby /usr/local/bin/ruby
```

Depois, no mesmo arquivo, tem que configurar o VirtualHost. Fui até o final do arquivo httpd.conf e adicionei:



```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

<VirtualHost *:80>
    ServerName www.testeadsitape.com
    DocumentRoot /home/rails_app/adsitape/public

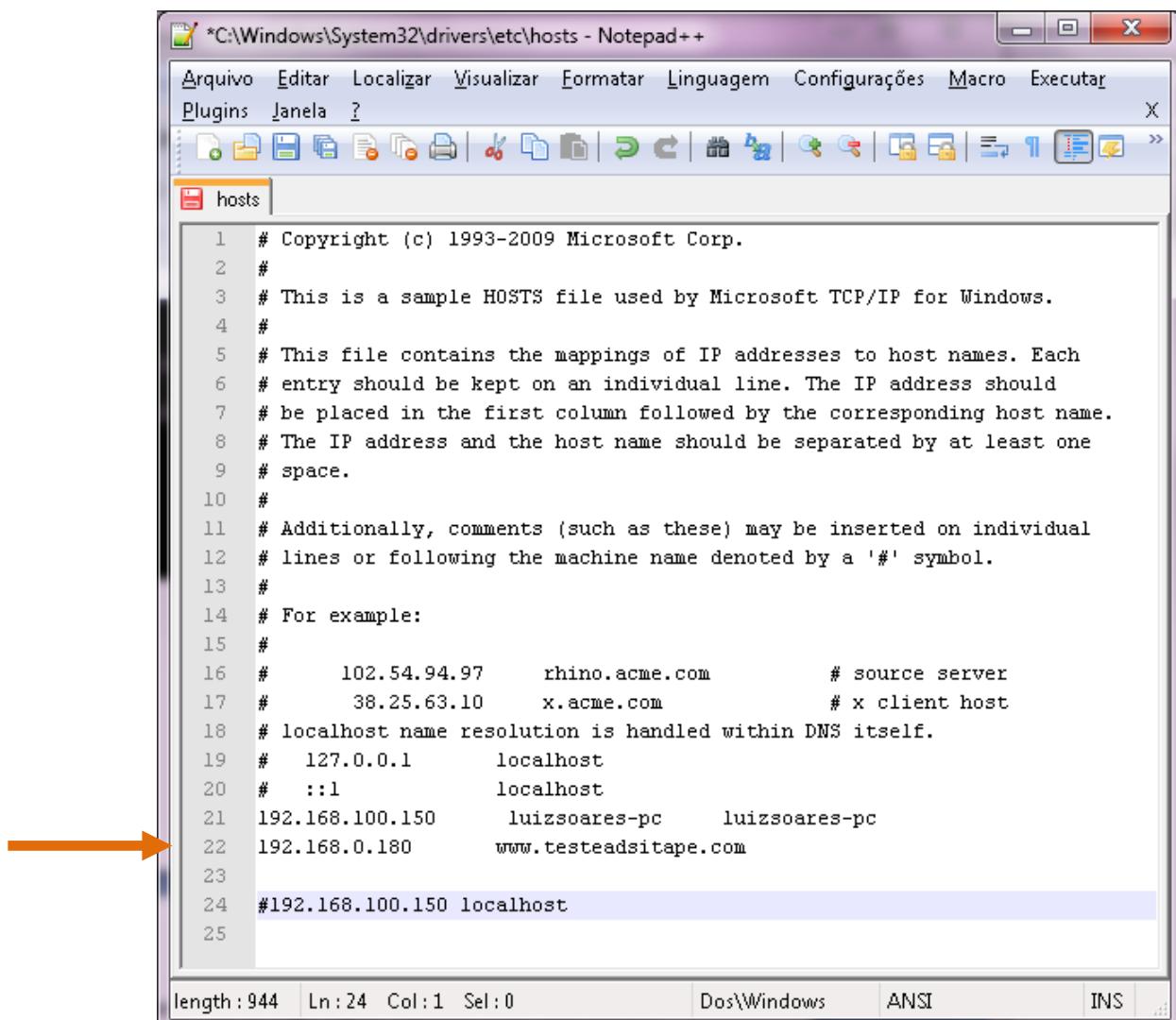
    <Directory /home/rails_app/adsitape/public>
        AllowOverride all
        Options -MultiViews
    </Directory>
</VirtualHost>
```

Depois reinicia o apache com o comando

```
/apache2/bin/apachectl restart
```

Agora, para o teste final (só não vou cruzar os dedos porque aí não consigo digitar aqui), é necessário dizer para a máquina Windows que o endereço [www.testeadsitape.com](http://www.testeadsitape.com) (que não existe na internet), está na máquina com ip 192.168.0.180.

Pra isso, no Windows, abre no bloco de notas o arquivo `c:\windows\system32\drivers\etc\hosts` e acrescenta uma linha para fazer o mapeamento



```
*C:\Windows\System32\drivers\etc\hosts - Notepad++
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Macro Executar
Plugins Janela ?
hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97    rhino.acme.com        # source server
17 #       38.25.63.10      x.acme.com            # x client host
18 # localhost name resolution is handled within DNS itself.
19 #       127.0.0.1      localhost
20 #       ::1           localhost
21 192.168.100.150     luizsoares-pc      luizsoares-pc
22 192.168.0.180       www.testeadsitape.com
23
24 #192.168.100.150 localhost
25
```

length : 944 Ln:24 Col:1 Sel:0 Dos\Windows ANSI INS

No Servidor real, isso terá que ser resolvido configurando o BIND para responder corretamente ao DNS<sup>11</sup>.

Agora testa no navegador:

---

<sup>11</sup> No caso do meu Cloud Server na Locaweb, onde já tinha o domínio registrado, bastou editar a configuração do serviço de DNS, mas este tutorial não cobre este tópico.



Ops. Falta alguma coisa. Dei uma olhada no log do apache na máquina virtual:

```
cat /apache2/log/error_log
```

E vi lá a mensagem “client denied by Server configuration /home/rails\_app/adsitape/public/favicon.ico”

Meu primeiro palpite é acreditar na mensagem que o apache devolveu para o navegador “você não tem permissão para acessar a pasta / neste servidor.”.

Quando descompactei a aplicação Rails, estava logado como usuário root. Isso significa que as permissões da pasta estão setadas para o root. Dei uma olhada no manual do Passenger que fala sobre isso, e decidir tornar o usuário “luiz” dono da pasta adsitape.

```
cd /home/rails_ap  
chown luiz adsitape -R  
/apache2/bin/apachectl restart
```

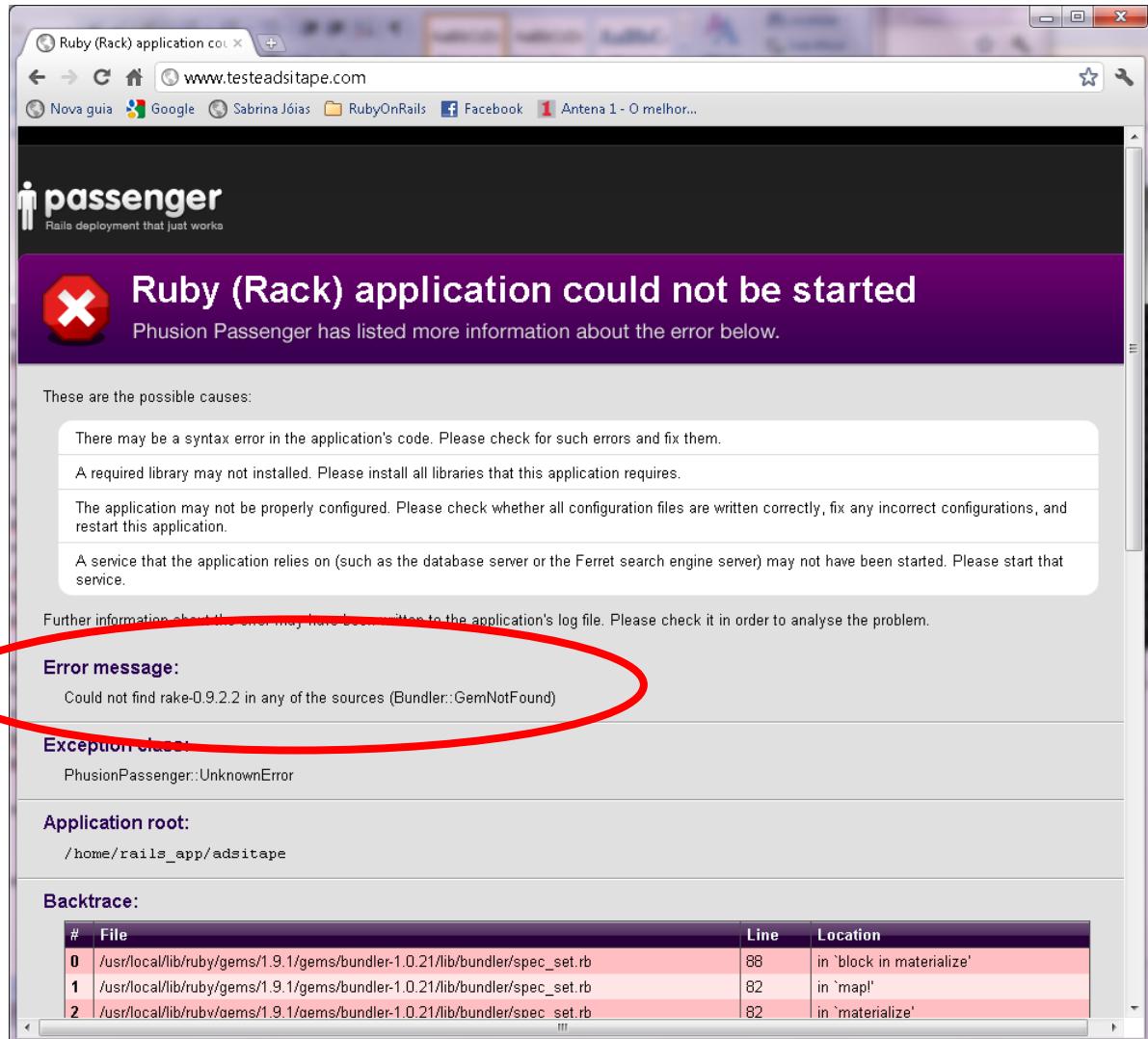
Voltei para o navegador e o resultado foi a mesma página de erro de permissão de acesso, e o arquivo de log do apache também continua informando o mesmo erro.

Tentei aumentar um pouco as permissões:

```
chmod 755 adsitape -R  
/apache2/bin/apachectl restart
```

Isso garante acesso completo para o dono da pasta (o usuário luiz) e acesso de somente leitura para outros usuários.

Voltando para o navegador, agora melhorou:



Ruby (Rack) application could not be started

Phusion Passenger has listed more information about the error below.

These are the possible causes:

- There may be a syntax error in the application's code. Please check for such errors and fix them.
- A required library may not be installed. Please install all libraries that this application requires.
- The application may not be properly configured. Please check whether all configuration files are written correctly, fix any incorrect configurations, and restart this application.
- A service that the application relies on (such as the database server or the Ferret search engine server) may not have been started. Please start that service.

Further information about the error may have been written to the application's log file. Please check it in order to analyse the problem.

**Error message:**

Could not find rake-0.9.2.2 in any of the sources (Bundler::GemNotFound)

**Exception class:**

PhusionPassenger::UnknownError

**Application root:**

/home/rails\_app/adsitape

**Backtrace:**

#	File	Line	Location
0	/usr/local/lib/ruby/gems/1.9.1/gems/bundler-1.0.21/lib/bundler/spec_set.rb	88	in `block in materialize'
1	/usr/local/lib/ruby/gems/1.9.1/gems/bundler-1.0.21/lib/bundler/spec_set.rb	82	in `map!'
2	/usr/local/lib/ruby/gems/1.9.1/gems/bundler-1.0.21/lib/bundler/spec_set.rb	82	in `materialize'

Melhorou 50%: essa mensagem significa que o Apache está recebendo a requisição e repassando para o Passenger, mas este último não conseguiu rodar a

O próximo passo é ter certeza de que a aplicação rails tem todas as gems de que precisa disponíveis:

```
cd /home/rails_app  
cd adsitape  
bundle install
```

Este último comando consulta o arquivo “GemFile” que está na pasta raiz da aplicação, e faz a instalação de todas as gems de que a aplicação precisa para rodar.

```
Máquina Visualizar Dispositivos Ajuda (H)
Using activerecord (3.1.3)
Using activeresource (3.1.3)
Installing ansi (1.4.1)
Using bundler (1.0.21)
Installing coffee-script-source (1.1.3)
Installing execjs (1.2.11)
Installing coffee-script (2.2.0)
Using rack-ssl (1.3.2)
Using json (1.6.3)
Using rdoc (3.11)
Using thor (0.14.6)
Using railties (3.1.3)
Installing coffee-rails (3.1.1)
Installing jquery-rails (1.0.19)
Installing libv8 (3.3.10.4)
Installing pg (0.11.0) with native extensions
Using rails (3.1.3)
Installing sass (3.1.11)
Installing sass-rails (3.1.5)
Installing therubyracer (0.9.9) with native extensions
Installing turn (0.8.2)
Installing uglifier (1.1.0)
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem
is installed.
root@debian:/home/rails_app/adsitape# _
```

O que me lembra que também é necessário criar o banco de dados (mesmo que a aplicação ainda não use ele para nada):

```
rake db:create
```

O que estranhamente retornou um erro de conexão com o banco de dados. Depois de pesquisas por umas duas horas<sup>12</sup>, achei a solução aqui: <http://www.ruby-forum.com/topic/611073>: como sempre acontece nesses casos, era uma tremenda bobagem. Basta acrescentar a linha `host: localhost` no arquivo de configuração de banco de dados do Rails:

```
vi /home/rails_app/adsitape/config/database.yml
```

---

<sup>12</sup> Na verdade antes disso foram umas três horas resolvendo o problema de localização do arquivo de PID, mas já coloquei a solução na seção de pós-instalação do PostgreSQL.



```
# development:
#   adapter: postgresql
#   host: localhost
#   encoding: unicode
#   database: adsitape_development
#   pool: 5
#   username: postgres
#   password: m6rpt2

# Connect on a TCP socket. Omitted by default since the client uses a
# domain socket that doesn't need configuration. Windows does not have
# domain sockets, so uncomment these lines.
#host: localhost
#port: 5432
```

Fiz isso nas três seções: development, test, e production (talvez não fosse necessário, mas também não fez nenhum mal). Isso não foi necessário na máquina de desenvolvimento, mas por via das dúvidas fiz lá também.

Não foi necessário incluir a linha port: 5432.

Tentar de novo criar o banco de dados com

```
rake db:create
```

Sucesso! Não retornou nenhuma mensagem de erro, o que significa que executou.

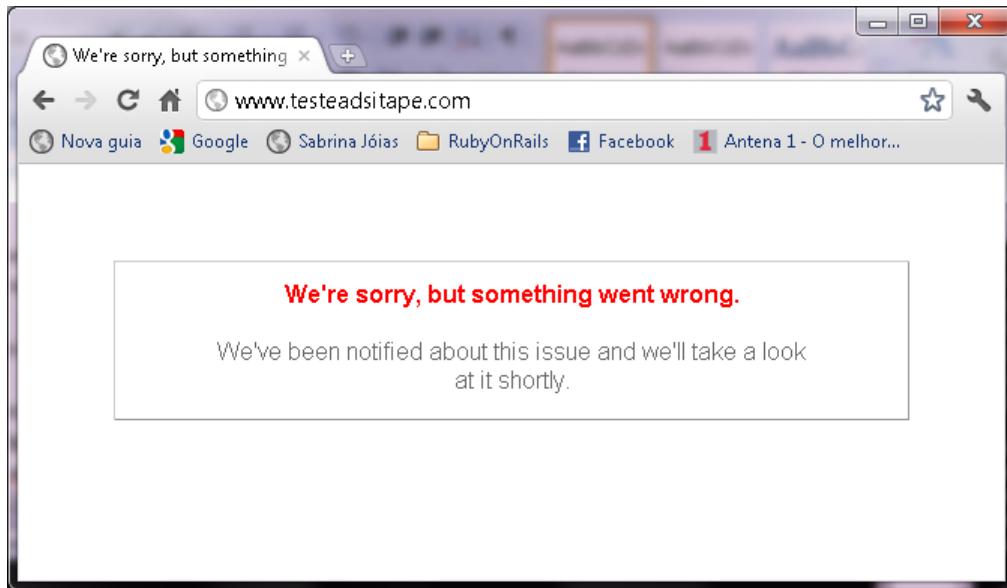
Agora precisa criar o banco de dados de produção (o comando anterior criou somente o banco development)

```
rake RAILS_ENV='production' db:create
```

Agora vamos tentar rodar a aplicação de novo:

```
/apache2/bin/apachectl restart
```

E agora aparece o erro.



O que é muito bom, porque essa mensagem veio do Rails, então significa que pelo menos o trio Apache+Passenger+Rails já está conseguindo conversar entre si.

Examinando o arquivo log/procution.log:

```
Debian Server 0 [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)

Started GET "/" for 192.168.0.197 at 2011-12-10 15:43:57 -0200
Processing by CronogramaController#basico as HTML
Rendered cronograma/basico.html.erb within layouts/application (71.3ms)
Completed 500 Internal Server Error in 138ms

ActionView::Template::Error (sem_logo.gif isn't precompiled):
  5:         </p>
  6:     </header>
  7:         <div class="emconstrucao">
  8:             <% logo = image_tag("sem_logo.gif", :alt => "Alguém pode
criar um logo pra pôr aqui?", :class => "logotipo")%>
  9:                 <%= link_to logo, root_path %>
 10:                 <h1>AdsItape.com</h1>
 11:                 <h1>Este site está em Construção</h1>
app/views/cronograma/basico.html.erb:8:in `/_app_views_cronograma_basico_html_e
rb__1540391795249023204_37895480'

root@debian:/home/rails_app/adsitape# ls assets
ls: impossível acessar assets: Arquivo ou diretório não encontrado
root@debian:/home/rails_app/adsitape# ls app/assets
images javascripts stylesheets
root@debian:/home/rails_app/adsitape# ls app/assets/images
rails.png sem_logo.gif
root@debian:/home/rails_app/adsitape# _
```

Bom, o arquivo de imagem está lá onde deveria estar, mas não sei o que significa a linha

```
ActionView::Template::Error (sem_logo.gif isn't precompiled):
```

Então, google nele, e encontrei isto:  
<http://stackoverflow.com/questions/7464544/ruby-on-rails-rake-assetsprecompile-error>, de onde deduzi que deveria executar o comando:

```
bundle exec rake assets:precompile RAILS_ENV=production
```

Tentando de novo rodar a aplicação:

```
/apache2/bin/apachectl restart
```

Ainda não deu certo, agora o production.log diz o mesmo  
Application::Template::Error apontando para o arquivo blueprint/screen.css

Achei a solução aqui: <http://excid3.com/blog/heroku-actionviewtemplateerror-css-isnt-precompiled/> onde dizia que devia configurar no arquivo config/environments/production.rb a linha

```
# Don't fallback to assets pipeline if a precompiled asset is missed
config.assets.compile = true
```

Isto feito, restart no apache, e agora o erro é:

```
Application::Template::Error permission denied
/home/rails_app/adsitape/tmp/cachê/assets/DAF
```

Achei a solução aqui: <http://stackoverflow.com/questions/7524262/permission-denied-in-tmp> . Como o Apache roda sob o usuário daemon (isso é configurado no arquivo /apache2/conf/httpd.conf), tornei esse usuário dono da pasta da aplicação:

```
chown -R daemon:daemon /home/rails_app/adsitape
```

Ok, novo restart do Apache, e novo teste no navegador

#### 4.9.2. SUCESSO!!!

The screenshot shows a Microsoft Internet Explorer browser window with the following details:

- Title Bar:** Adsltape
- Address Bar:** www.testeadsitape.com
- Toolbar:** Nova guia, Google, Sabrina Jóias, RubyOnRails, Facebook, Antena 1 - O melhor...
- Page Content:**
  - Header: mande sugestões: luiz.jesus@fatec.sp.gov.br
  - Main Title: Adsltape.com
  - Section: Este site está em Construção
  - Text: Só para o caso de você não ter percebido...
  - Call-to-action button: Logotipo para Adsltape.com? Me mande um!!! (highlighted with a red box)
  - Text area:
    - Logotipo:** preciso de um logotipo para o site. Você pode criar um (imagem em formato .gif ou .jpg) e me mandar por e-mail
    - Estilização:** Que tal brincar com CSS e me ajudar a criar um bom estilo para este site? No seu navegador, clique Arquivo/ Salvar Como (ou algo parecido), salve esta página em uma pasta no seu computador. Abra a pasta e dê clique duplo no arquivo index.html para ele abrir no seu navegador. Volte para a pasta, deve ter um subpasta adsltape\_files (ou algo parecido), dentro dela deve ter um arquivo custom.css. Abra esse arquivo com o bloco de notas, altere alguma coisa (um item de cada vez!), salve o arquivo, volte para o navegador e pressione F5 para ver o que mudou. Quando achar que terminou, me mande o seu arquivo css por e-mail!
- Section: Solicitação de Ajuda**
- Section: Cronograma**

Ação	Data
Página de aviso de construção	05/12/2011
Estilização de página de aviso e rascunho do logotipo	06/12/2011
Colocar a primeira página no ar (para o mundo!)	
A empresa onde pretendia hospedar o site não suporta Rails 3.1	07/12/2011
Então por enquanto vai html puro mesmo, até achar outra solução pra isso	
Tutorial para criação de ambiente para desenvolvimento Ruby on Rails	

Babyloni



## 5. Máquina Real

O que fiz aqui foi repetir tudo em uma nova máquina virtual Debian Limpa (o primeiro clone gerado antes de qualquer instalação de pacotes), colocando em ordem as dependências de pacotes e fazendo as correções necessárias nas configurações.

Depois disso parti para o Cloud Server Real.

Não vou repetir tudo aqui, basta dizer que segui o caminho exato descrito neste tutorial, desde a primeira instrução.

### 5.1. Último problema

Na fase de transferir o arquivo zipado da aplicação para o servidor real, encontrei um problema que me pareceu insolúvel. A transferência com o comando scp, a partir da máquina virtual Ubuntu para a máquina real não funcionava, dava 100% mas não finalizava a transferência, quando ia olhar no servidor estava criado lá um arquivo com tamanho zero bytes.

Tentei uma porção de coisas: primeiro transferir o arquivo da máquina virtual Debian para a máquina real, o que não funcionou da mesma forma.

Depois configurei um serviço de FTP na máquina real, e tentei transferir o arquivo utilizando o programa Filezilla, o que também não dava certo, não finalizava o envio. O que é estranho, porque uma transferência via FTP a partir da máquina Windows funcionava normalmente.

Aí tentei, ainda na máquina Ubuntu, enviar o arquivo por e-mail para mim mesmo, de forma que eu pudesse abri-lo na máquina Windows e transferi-lo via FTP; mas isso também não funcionou, não conseguia enviar e-mail a partir do Ubuntu com arquivos anexados (e-mails sem anexos funcionavam normalmente).

Até que por último, na máquina Windows, usei o pscp para puxar o arquivo da máquina Ubuntu, e aí fiz a transferência via FTP.

Não consegui resolver esse problema da máquina virtual (tanto a Ubuntu quanto a Debian), de não conseguir fazer transferência de arquivos para fora (entre

elas funcionava normalmente). Mesmo pesquisando por horas na internet, não achei nem mesmo referências a essa situação.

Dias depois, o problema ficou bem mais grave, quando não consegui utilizar o git<sup>13</sup> pra transferir a aplicação da máquina Ubuntu, que uso para desenvolvimento, para o meu repositório GitHub.

E foi aí que me deu um estalo. Não fazia o menor sentido, mas resolvi tentar: alterei a configuração da máquina virtual para utilizar placa de rede em modo NAT, ao invés do modo Bridge. E isso resolveu o problema (!!!), tudo voltou a funcionar normalmente na transferência de arquivos, scp, anexo em e-mail, transferência via FTP.

Por que essa estória toda? Só pra reforçar: o melhor de trabalhar com sistemas é enfrentar esse tipo de problema (desde que você não tenha um chefe ou cliente exigindo o cumprimento de um prazo).

## 5.2. Concluindo

Não se esqueça de ler os avisos legais no Capítulo 1.

Dúvidas, sugestões e correções são bem-vindas, e podem ser enviadas diretamente ao meu e-mail [luiz.jesus@fatec.sp.gov.br](mailto:luiz.jesus@fatec.sp.gov.br). No momento da edição deste artigo (19/12/2011), ainda não está pronta a função na página [www.adstape.com/tutoriais](http://www.adstape.com/tutoriais) que permitirá comentários diretamente lá, mas acredito que estará pronto até 31/01/2012.

---

<sup>13</sup> <https://github.com/>