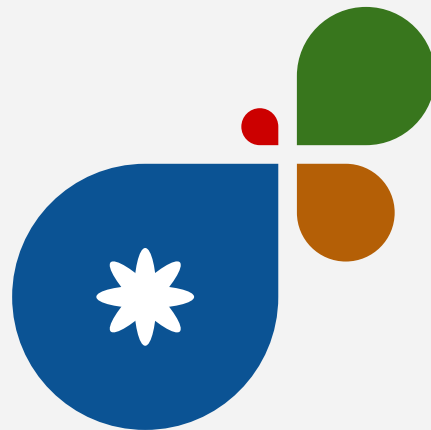


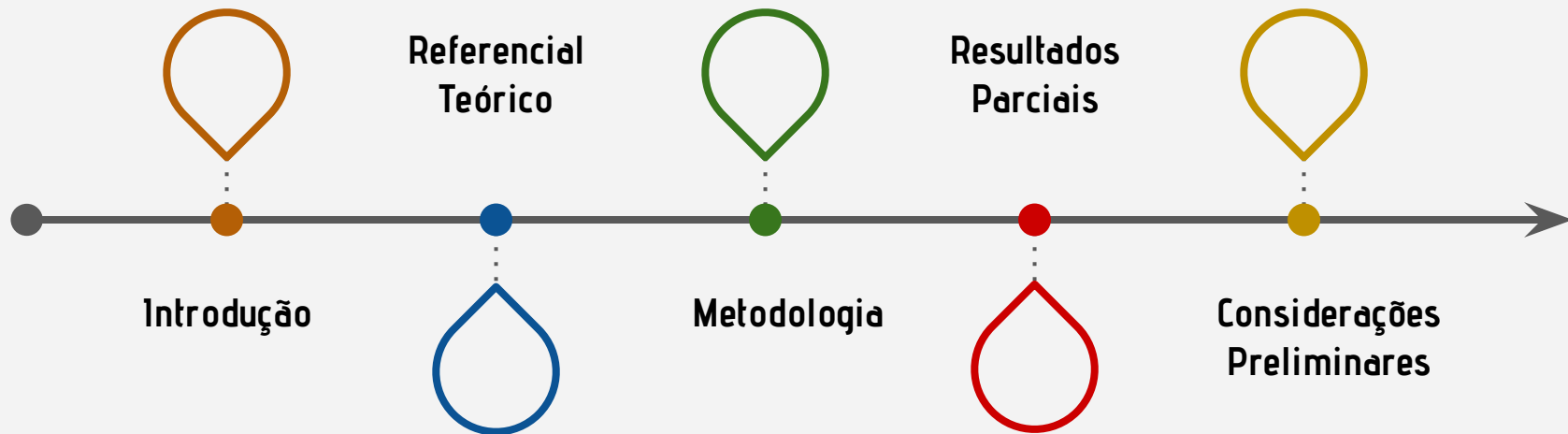
# Implantação automatizada de aplicações em sistema Debian GNU/Linux

Thiago de Souza Fonseca Ribeiro

Universidade de Brasília - Engenharia de  
Software



# Agenda





# Introdução

## Contextualização

- ❖ A forma como as equipes de desenvolvimento entregam software está mudando.
- ❖ Implantação do software ainda é uma atividade que pode ser complexa.
- ❖ Sistemas voltado para a Internet consistem em muitas partes que são complicadas de implantar.
- ❖ Em uma implantação de software, executar todas essas tarefas de forma manual pode ser uma atividade complicada
- ❖ Implantação de software em distribuições linux
- ❖ Diferentes tipos de perfis de usuários
- ❖ Usuário precise fazer algumas configurações específicas após a instalação
- ❖ Uso de alguma ferramenta que automatize todo esse processo de implantação



# Introdução

O problema

Como implantar aplicações web em sistema Debian GNU/Linux de forma automatizada e segura?



# Introdução

## Objetivos

Contribuição da construção de uma ferramenta livre, que possa automatizar instalação e configuração de aplicações web em sistemas Debian GNU/LINUX, a partir de pacotes que sejam distribuídos oficialmente pelo Debian.



# Referencial Teórico

## Implantação de Software

- ❖ Importância da fase de implantação de software
- ❖ Implantação automatizada de software
- ❖ Processo de implantação de software descrito pela OMG em 2016 possui Papéis + Fases
- ❖ Papéis:
  - Implantador
  - Ambiente Alvo
  - Nó
  - Pacote



# Referencial Teórico

## Implantação de Software

### ❖ Fases

- Planejamento
- Preparação
- Instalação
- Configuração
- Inicialização



# Referencial Teórico

DevOps

- ❖ Como resultado ao longo dos anos, o que aconteceu é que as equipes de desenvolvimento de software são capazes de entregar a um ritmo muito mais rápido, principalmente a adoção de métodos ágeis de desenvolvimento de software(VIRMANI, 2015)





# Referencial Teórico

DevOps





# Referencial Teórico

## DevOps

- ❖ DevOps pode ser visto como um conjunto de práticas e princípios para a entrega de software, onde a chave é o foco e a velocidade de entrega e automação, para pode auxiliar na capacidade de reagir a mudanças (VIRMANI,2015)
- ❖ Em (VIRMANI,2015) é abordado as práticas DevOps, que podem ser aplicadas diversas vezes dentro de um ciclo de desenvolvimento de software, as praticas DevOps podem ser resumidas em:
  - Planejamento Contínuo
  - Integração Contínua
  - Implantação Contínua
  - Teste Contínuo
  - Monitoração Contínua



# Referencial Teórico

Métodos e ferramentas para implantação de software

- ❖ Existem várias maneiras de se automatizar um processo de implantação software:
- ❖ Tais ferramentas permitem-nos controlar e automatizar a configuração de todos os elementos que compõem um sistema.
- ❖ Removem boa parte do esforço manual
- ❖ Exemplos de Ferramentas:
  - Linguagens de script como python e shell
  - Ferramentas como chef ou puppet
  - Sistemas de middleware especializados em determinados tipos de artefatos implantáveis



# Referencial Teórico

Métodos e ferramentas para implantação de software

- ❖ Além das ferramentas de automação, é importante também entender o empacotamento de programas, já que tanto chef como puppet utilizam a instalação de pacotes como recurso para poder instalar softwares
- ❖ É possível buscar os softwares desejados a partir dos pacotes disponíveis em uma distribuição linux, em seus repositórios.
- ❖ Existem pacotes que dependem do sistema operacional, como por exemplo, pacotes .deb para a distribuição Debian e pacotes .rpm para a distribuição Fedora



# Referencial Teórico

## Pacotes Debian

- ❖ O formato utilizado pelo Debian GNU/LINUX e seus derivados é o formato de pacotes binários conhecido como `.deb`.
- ❖ Um pacote `.deb` é composto de:
  - arquivos executáveis;
  - bibliotecas;
  - Documentação associada a um programa ou a um conjunto de programas;
  - Todos os dados e procedimentos necessários para instalar um programa.
- ❖ A estrutura dos pacotes e também os seus requisitos para que sejam distribuídos oficialmente pelo Debian estão especificados no Manual de Políticas do Debian



# Referencial Teórico

## Pacotes Debian

- ❖ A estrutura de um pacote debian pode ser observado a partir de qualquer pacote .deb, onde é possível encontrar três estruturas a seguir:
  - **debian-binary**: Este arquivo contém a versão da especificação do empacotamento implementado no pacote.
  - **control.tar.gz**: Este arquivo contém todas as informações disponíveis, como o nome e a versão do pacote. Essas informações servem para que as ferramentas de gerenciamento de pacotes determinarem se é possível instalar e desinstalar o pacote.
  - **data.tar.gz**: Este arquivo contém todos os arquivos para serem extraídos do pacote, ou seja, é onde ficam os arquivos executáveis, a documentação, os devidos diretórios em que tais arquivos devem ser copiados.



# Referencial Teórico

## Múltiplas Instâncias de Aplicações Web com Hospedagem Virtual

- ❖ Na implantação de software deve-se escolher um ambiente alvo, onde a aplicação será instalada, esse ambiente deve possuir um nó, que é um recurso computacional onde se implanta um componente
- ❖ Porém pode existir a necessidade de implantar várias aplicações no mesmo servidor
- ❖ Uma forma de solucionar esse problema é o uso de Hospedagem Virtual
- ❖ Alguns softwares possuem suporte para a implementação de hospedagem virtual, como por exemplo o Apache e o Nginx.



# Referencial Teórico

## Múltiplas Instâncias de Aplicações Web com Hospedagem Virtual

Exemplo de arquivo de configuração no Apache

```
<VirtualHost*:80>
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var /www/html
```

```
ErrorLog ${APACHE_LOG_DIR}/ error . log
```

```
CustomLog ${APACHE_LOG_DIR}/ access . log combined
```

```
</ VirtualHost>
```





# Referencial Teórico

Múltiplas Instâncias de Aplicações Web com Hospedagem Virtual

Exemplo de arquivo de configuração no Nginx

```
server {  
listen 80;  
root /var /www/example . com/public_html ;  
index index . html index . htm ;  
server_name example . com ;  
}
```



# Referencial Teórico

## Segurança na Implantação de Aplicações Web

- ❖ Um pacote também não pode prever quais são os procedimentos de segurança que devem ser tomados numa implantação.
- ❖ Uso de protocolos que utilizam criptografia para a segurança de dados,
- ❖ As aplicações web utilizam protocolos da camada de aplicação, incluindo os protocolos HTTP e SMTP, que são protocolos importantes neste trabalho, já que aplicações web utilizam o HTTP para transferência de arquivos e o SMTP para transferência de mensagens de correio eletrônico



# Referencial Teórico

## Segurança na Implantação de Aplicações Web

- ❖ HTTP é um protocolo da camada de aplicação da web, e é implementado em dois programas, um cliente e um servidor, conversando um com outro a partir de troca de mensagens
- ❖ Já o SMTP é protocolo responsável por transferir mensagens de servidores de correio remetentes para servidores de correio destinatários, também transferindo arquivos, de um servidor de correio para o outro.
- ❖ Uma diferença entre o HTTP e o SMTP é que o protocolo HTTP é um protocolo de recuperação de informações, enquanto o protocolo SMTP é um protocolo de envio de informações.
- ❖ Existem também os protocolos de acesso ao correio, entre eles o POP3 e IMAP.



# Referencial Teórico

## Segurança na Implantação de Aplicações Web

- ❖ Para adicionar uma camada de segurança, com sigilo e integridade de dados é necessário o uso de outro protocolo, o protocolo SSL, para complementar os protocolos HTTP e SMTP.
- ❖ Para poder utilizar o SSL é necessário que o seu servidor web possua um certificado, que possa responder algumas questões sobre a identidade do seu servidor
- ❖ As assinaturas e certificados digitais servem para agregar confiança e segurança nas trocas de dados pela internet
- ❖ Com as devidas configurações, tanto HTTP como SMTP estarão sobre uma camada adicional de segurança, que utiliza o protocolo SSL/TLS.



# Referencial Teórico

## Segurança na Implantação de Aplicações Web

- ❖ O que viabiliza a assinatura digital é o emprego de criptografia assimétrica ou criptografia de chaves públicas.
- ❖ Uma aplicação importante de assinaturas digitais é a certificação de chaves públicas, que certifica que uma chave pública pertence a uma entidade específica
- ❖ Uma maneira de se obter o certificado, é através de uma entidade conhecida como entidade certificadora
- ❖ Uma outra maneira de se obter certificados é a partir dos certificados autoassinados, que são os certificados gerados por conta própria através de ferramentas.



# Metodologia

## Metodologia do trabalho

- ❖ O problema e o objetivo como ponto de partida:
  - Como implantar aplicações web em sistema Debian GNU/Linux de forma automatizada e segura?
  - Contribuição da construção de uma ferramenta que possa automatizar instalação e configuração de aplicações web em sistemas Debian GNU/LINUX,
  - Essa construção será feita a partir dos conhecimentos adquiridos durante o curso de engenharia de software
  - Além disso a solução proposta deverá passar por uma validação dos resultados
  - Exemplos de uso onde cada exemplo de uso terá novos desafios para testar a solução proposta



# Metodologia

Metodologia do trabalho

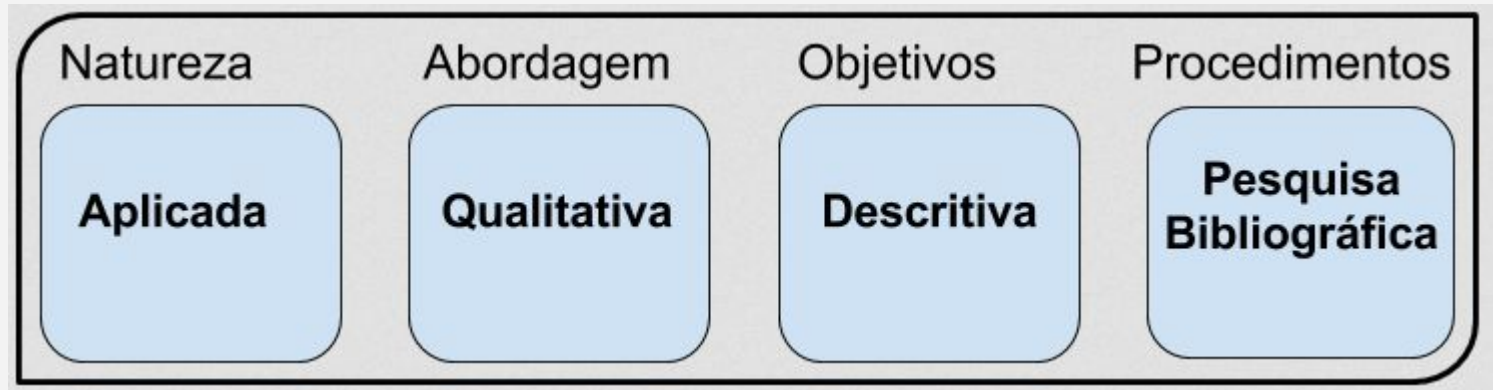
- ❖ E metodologia está organizada em
  - Pesquisa descritiva nos trabalhos relacionados
  - Metodologia de desenvolvimento para construção da solução
    - Métodos, Ferramentas e Processos. ( PRESSMAN, 2011)
  - Validação da Solução



# Metodologia

Trabalhos relacionados

- ❖ Compreender melhor como as soluções de automação de implantação de software funcionam.







# Metodologia

Trabalhos relacionados

- ❖ CHOReOS Enactment Engine (LEITE,2014)
- ❖ BItنامي
- ❖ SandStorm.io
- ❖ JuJu



# Metodologia

Construção da solução

A engenharia de software é composta por um conjunto de três elementos fundamentais:

- ❖ Métodos: Como fazer a implementação do software, com modelos, especificações e critérios para qualidade.
- ❖ Ferramentas: Objetivo de apoio automatizado para auxiliar as atividades de engenharia de software.
- ❖ Processos: Definem a sequência de práticas que serão utilizadas no desenvolvimento.



# Metodologia

Construção da solução – Métodos

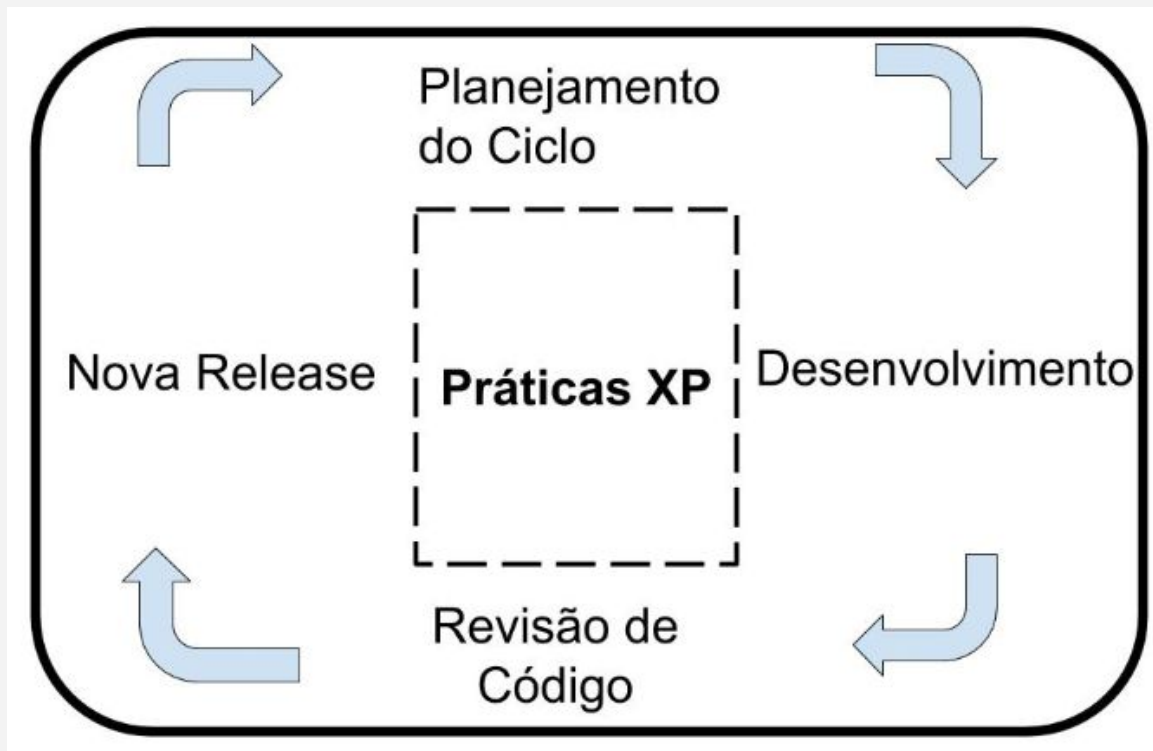
O método que será seguido para desenvolvimento de software é o método ágil chamado programação extrema (XP), de acordo com (BECK, 1999) essa metodologia contém um conjunto de práticas que auxiliam um engenheiro de software no desenvolvimento.

- ❖ Pequenas versões
- ❖ Design simples
- ❖ Programação em pares
- ❖ Revisão de código

Também será feito o uso de iterações semanais, ou seja, as atividades planejadas não devem ultrapassar o prazo de uma semana.

# Metodologia

Construção da solução - Métodos





# Metodologia

Construção da solução – Ferramentas de Apoio

- ❖ Ferramenta de controle de versão
  - ❖ Ferramenta para documentação de código
  - ❖ Ferramenta para gerenciamento de tarefas
  - ❖ Ferramenta de comunicação
- 
- As ferramentas escolhidas foram Gitlab + irc e listas de email



# Metodologia

## Construção da solução – Processos

Por fim é necessário definir a sequência de práticas e atividades que serão utilizadas no desenvolvimento da construção da solução.

- ❖ Definir as fases e os procedimentos para implantação automatizada
- ❖ Definir os procedimentos de segurança na implantação

Para isso alguns aspectos foram levados em consideração

- ❖ Alguns cuidados de segurança ao implantar aplicações web ex: uso de https
- ❖ Implantação automatizada de múltiplas instâncias com hospedagem virtual.
- ❖ Uso de aplicações empacotadas no debian
- ❖ Uso de boas práticas de implantação de software



# Metodologia

## Ferramenta Shak

A ferramenta para apoio a implantação de software escolhida será a ferramenta shak. Ela é composta de:

- ❖ Livro de receitas chef
- ❖ Código ruby
- ❖ Sevidor Web Nginx
- ❖ Pacotes debian
- ❖ Gems
- ❖ Códigos ShellScript



# Metodologia

## Validação da Solução e Exemplos de Uso

Para validar a solução proposta no trabalho serão feitos exemplos de uso com aplicações que possam servir para a execução da arquitetura construída. As aplicações escolhidas devem possuir as seguintes características:

- ❖ Aplicações empacotadas no debian:
- ❖ Servidor web compatível:
- ❖ Aplicações com comunidades ativas:
- ❖ Documentação do software:

Além disso, é necessário a escolha de uma ferramenta para gerenciar ambientes de desenvolvimento.



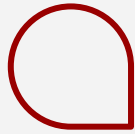


# Metodologia

## Validação da Solução e Exemplos de Uso

Para encontrar as aplicações que possam se encaixar dentro desses parâmetros devemos buscar por alguns exemplos de uso

- ❖ apt-cache search web | wc -l
  - 3470 pacotes de diversas aplicações ou módulos de aplicações
  - Ex: wordpress, chormium, drupal, mailman
- ❖ Analisando a documentação de algumas aplicações como wordpress, redmine, owncloud, mailman, wikimédia, foram escolhidas as aplicações wordpress e owncloud por atingirem completamente as características definidas anteriormente..



# Resultados Parciais

## Planejamento das atividades

Antes de iniciar as atividades, foi feito um planejamento, no qual teve o objetivo de levantar as principais atividades de desenvolvimento que poderiam ser realizadas, são elas:

- Issue 1 Livro de Receitas para instalação do Wordpress.
- Issue 7 Livro de Receitas para instalação do Owncloud.
- Issue 8 Livro de Receitas para instalação do servidor de e-mail.
- Issue 9 Forçar Wordpress a utilizar https.
- Issue 10 Forçar Owncloud a utilizar https.

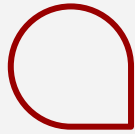
❖ A ferramenta escolhida para auxiliar a criação de um ambiente de desenvolvimento foi a ferramenta Vagrant.



# Resultados Parciais

## Aplicação Wordpress

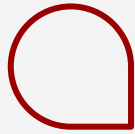
- ❖ WordPress é um dos maiores software de publicação de conteúdo, sendo hoje a maior plataforma de Gerenciamento de Conteúdo do mundo, com quase 70% do mercado, O wordpress foi a primeira ferramenta escolhida para automatizar a instalação.
- ❖ É necessário definir as fases e procedimentos para a implantação automatizada, para que possam ser automatizadas pelo Shak, são elas:
  - Planejamento
  - Preparação e Instalação de Pacotes
  - Configuração
  - Configuração de múltiplas instâncias
  - Inicialização



# Resultados Parciais

Aplicação Wordpress – Planejamneto

- ❖ O planejamento da implantação é uma fase para identificar os componentes necessários na implantação da aplicação.
  - Pacote wordpress para o Debian
  - Pacote Nginx para o Debian
  - Pacote mysql para o Debian
  - Arquivos de configuração do wordpress



# Resultados Parciais

## Aplicação Wordpress – Preparação e instalação

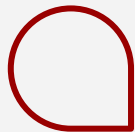
- ❖ São os procedimentos necessários para preparar o ambiente alvo para que o wordpress possa ser executado, isso envolve configuração do sistema operacional, instalação e configuração de dependências necessárias, e a transferência do componente para o servidor.
  - Instalação do php5
  - Pacote php5-fpm, mysql e nginx.
  - Serviços mysql e do php5-fpm
  - Livro de Receitas do wordpress com a estrutura básica necessária pelo chef
  - rake cookbook
  - Cookbook name : wordpress



# Resultados Parciais

## Aplicação Wordpress – Configuração

- ❖ Como levantado no planejamento é necessário a edição de arquivos de configuração do wordpress, arquivos de configuração do banco de dados e arquivo de configuração do nginx.
- ❖ Arquivos
  - config.php
  - database.sql
  - wordpress.conf
  - Inicialização
  - Configuração de múltiplas instâncias

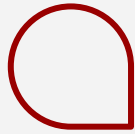


# Resultados Parciais

Aplicação Wordpress – Inicialização

```
shak install wordpress hostname=wordpress.dev
```

A partir disso basta checar o navegador no endereço <https://wordpress.dev>



# Resultados Parciais

## Aplicação Owncloud

- ❖ Owncloud é uma ferramenta para compartilhamento de arquivos, é um software livre em que é possível compartilhar um ou mais arquivos e pastas do seu computador na nuvem, e sincronizá-los com o seu servidor ownCloud,
- ❖ É necessário definir as fases e procedimentos para a implantação automatizada, para que possam ser automatizadas pelo Shak, são elas:
  - Planejamento
  - Preparação e Instalação de Pacotes
  - Configuração
  - Configuração de múltiplas instâncias
  - Inicialização





# Resultados Parciais

## Aplicação Owncloud - Planejamento

- ❖ O planejamento da implantação é uma fase para identificar os componentes necessários na implantação da aplicação.
  - Pacote owncloud para o Debian
  - Pacote Nginx para o Debian
  - Pacote postgresql para o Debian
  - Arquivos de configuração do owncloud

A diferença da escolha do banco de dados em relação ao wordpress é poder aumentar o suporte a diferentes bancos de dados. A escolha do postgresql vêm com a possibilidade de trabalhar com dois bancos de dados diferentes, aumentando o suporte do Shak



# Resultados Parciais

## Aplicação Owncloud - Preparação e Instalação

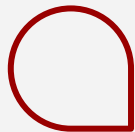
- ❖ São os procedimentos necessários para preparar o ambiente alvo para que o owncloud possa ser executado, isso envolve configuração do sistema operacional, instalação e configuração de dependências necessárias, e a transferência do componente para o servidor.
  - Instalação do php5
  - instalação do pacote php5-pgsql
  - Pacote php5-fpm, mysql e nginx.
  - Serviços mysql e do php5-fpm
  - Livro de Receitas do wordpress com a estrutura básica necessária pelo chef
  - rake cookbook e Cookbook name : wordpress



# Resultados Parciais

## Aplicação Owncloud – Configuração

- ❖ Como levantado no planejamento é necessário a edição de arquivos de configuração do owncloud, arquivos de configuração do banco de dados e arquivo de configuração do nginx.
- ❖ Arquivos
  - autoconfig.php
  - postgresql-conf.conf
  - owncloud.conf
  - Inicialização
  - Configuração de múltiplas instâncias

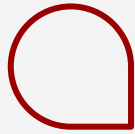


# Resultados Parciais

Aplicação Owncloud - Inicialização

```
shak install owncloud hostname=owncloud.dev
```

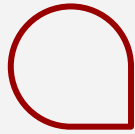
A partir disso basta checar o navegador no endereço <https://owncloud.dev>



# Resultados Parciais

## Servidor de e-mail

- ❖ Servidores de e-mail formam a infraestrutura do e-mail, sendo o SMTP o protocolo mais importante pois é o responsável por transferir as mensagens de servidores de e-mail remetentes para servidores de e-mail destinatários, existem protocolos de acesso ao correio como POP3, IMAP.
- ❖ Importância de servidor de e-mail no Shak.
- ❖ As fases e procedimentos para a implantação automatizada, são:
  - Planejamento
  - Preparação e Instalação de Pacotes
  - Configuração
  - Inicialização



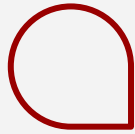
# Resultados Parciais

Servidor de e-mail – Planejmaneto

- ❖ O protocolo escolhido para IMAP, ao invés do imap.
- ❖ O servidor IMAP escolhido foi o Dovecot que é um servidor de e-mail IMAP
- ❖ Agente de transferências de e-mails (MTA) escolhido foi o Postfix,
- ❖ A aplicação que servirá como anti-spam será o Apache SpamAssassin

Logo os recursos necessários são:

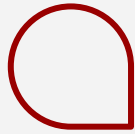
- ❖ Pacote dovecot para o Debian;
- ❖ Pacote postfix para o Debian;
- ❖ Pacote spamassasin para o Debian;
- ❖ Arquivos de configuração: Criação dos arquivos de configuração necessários para configurar o spamassasin, dovecot e postfix.



## Resultados Parciais

Servidor de e-mail - Preparação e instalação dos pacotes

- ❖ São os procedimentos necessários para preparar o ambiente alvo para que o servidor de e-mail possa ser executado, é preciso configurar as dependências assim como as demais aplicações, e são elas:
  - Instalação dos pacotes postfix + bsd-mailx
  - instalação do pacote dovecot-imapd
  - Instalação do pacote spamassasin + spamc
  - Serviços dovecot, spamassasin e postfix.
  - Livro de Receitas do e-mail com a estrutura básica necessária pelo chef
  - rake cookbook
  - Cookbook name : email

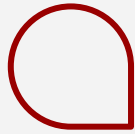


# Resultados Parciais

## Servidor de e-mail - Configuração

- ❖ Como levantado no planejamento é necessário a edição de arquivos de configuração do servidor de email, de todas as ferramentas citadas.
  - 10-mail.conf que é um arquivo de configuração para indicar onde será o local no qual os e-mail ficarão.
  - 10-ssl.conf que é o arquivo em que possui os caminhos do certificado ssl e da chave
  - 11-postfix-auth.conf, arquivo responsável por indicar o caminho do arquivo de autenticação do postfix.
  - 20-imap.conf onde será indicado o tamanho máximo de conexões por ip permitidas no servidor.
  - 20-disable-imap-non-ssl.conf que permitirá que o imap utilize sempre uma conexão segura utilizando o imaps (imap + ssl).





## Resultados Parciais

### Servidor de e-mail - Configuração

- ❖ Para o postfix são apenas dois arquivos de configuração, o main.cf e o master.cf
  - main.cf é aonde se configura os parâmetros mínimos de configuração, lá dizemos que vamos filtrar os e-mail com spamassassin e algumas configurações do SMTP
  - master.cf definimos como um programa cliente se conecta a um serviço, e qual o programa que é executado quando um serviço é solicitado. Neste caso é habilitado recursos como: smtpd usar TLS e autenticação, caminho dos arquivos de chave e certificados o destino das mensagens como o host do servidor de e-mail, o smtpd utilizar o dovecot e rejeitar conexões não autenticadas



## Resultados Parciais

### Servidor de e-mail - Configuração

- ❖ Por fim a configuração do spamassassin, são necessários dois arquivos de configuração  
spamassassin e o local.cf
  - No spamassassin você habilita o spamassassin e configura o caminho do arquivo de log, opções e habilitar a cronjob que será executada de tempos em tempos.
  - Já o local.cf possui um parâmetro importante, o parâmetro required\_score é um nível de 0 a 10 em que são classificados os e-mails como spam, por default esse valor é 5.



## Resultados Parciais

Servidor de e-mail - Inicialização

Para instalar e inicializar: `shak install e-mail hostname=owncloud.dev`

Para testar o funcionamento foram feitos dois procedimentos, o primeiro era utilizar um cliente de e-mail para testes, o cliente de e-mail escolhido foi o mutt, utilizando suas funções básicas de enviar e receber e-mails. O segundo passo foi realizar conexões com o servidor de e-mail via telnet, com os seguintes comandos:

```
telnet localhost imap
```

```
telnet localhost imaps
```

Pela configuração do servidor de e-mail não será possível abrir uma conexão telnet em imap, porém em imaps será possível.

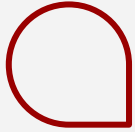


# Resultados Parciais

## Segurança

Na implantação das aplicações foram feitos dois procedimentos de segurança, o primeiro é forçar as aplicações web a sempre utilizarem o protocolo https e a segunda foi forçar o servidor de e-mail a não permitir a conexão via protocolos sem criptografia

- ❖ Para isso foi necessário gerar um certificado autoassinado.
- ❖ Adicionando suporte ao openssl ao Shak, possibilitando gerar certificados autoassinados com openssl.
- ❖ Com as chaves geradas basta configurar as aplicações indicando os caminhos dos certificados e das chaves, para forçar as aplicações web a utilizarem o https foi necessário fazer uma configuração específica no servidor web Nginx, já no servidor de email, apenas indicar o caminho dos certificados.

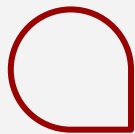


# Resultados Parciais

## Segurança

Configuração para forçar https nas aplicações via nginx:

```
server {  
    server _name <%= @hostname %>;  
    rewrite ^ https :// $server _name\ $request _uri ? permanent ;  
}  
  
server {  
    server _name <%= @hostname %>;  
    listen 443 ssl ;  
    ssl _certificate /etc/ssl/certs/<%= @hostname %>.pem ;  
    ssl _certificate_key /etc/ssl/ private/<%= @hostname %>.key ;  
}
```

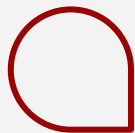


# Resultados Parciais

Exemplo: Wordpress

Neste exemplo, será mostrado como ficou o livro de receitas do wordpress, inicialmente a estrutura de arquivos do livro de receitas é a seguinte:

```
├── input.rb
├── metadata.rb
├── README.md
├── recipes
│   └── default.rb
├── templates
│   └── default
│       ├── config.php.erb
│       ├── mysql-conf.sql.erb
│       ├── nginx.conf.erb
│       └── _nginx.erb
3 directories, 8 files
```



# Resultados Parciais

Exemplo: Wordpress

No arquivo default.rb é aonde colocamos os comandos chef para instalar pacotes, criar arquivos de configuração, criar diretórios .

```
include_recipe "shak::php5_nginx"  
  
package 'wordpress'  
package 'mysql-server'
```



# Resultados Parciais

Exemplo: Wordpress

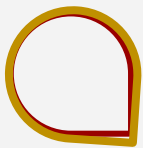
No arquivo default.rb é aonde colocamos os comandos chef para instalar pacotes, criar arquivos de configuração, criar diretórios.

```
template "#{app['id']}: create config.php" do
  path "/etc/wordpress/config-#{app['hostname']}.php"
  source "config.php.erb"
  variables :app => app
end

directory "#{app['id']}: create #{destination_dir}" do
  path destination_dir
  recursive true
end

execute "#{app['id']}: copy wp-content to #{destination_dir}" do
  command "sudo cp #{basedir}/wp-content/ #{destination_dir} -rf"
  not_if "test -d #{destination_dir}/wp-content"
end
```

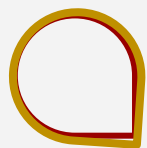




# Considerações Preliminares

## Considerações

- ❖ Todos esses resultados parciais são importantes para o objetivo do trabalho, que vão desde compreender como é o processo de implantação de um software até os cuidadosa se tomar ao implantar uma aplicação,
- ❖ Algumas dificuldades foram encontradas, principalmente quando as aplicações não davam suporte a múltiplas instâncias.
- ❖ Algumas melhorias foram feitas na ferramenta Shak.



# Considerações Preliminares

Cronograma TCC2

Atividade	Data de Início	Data de Término
Levantamento de bugs e soluções de bugs no Shak	01/03/2016	15/03/2016
Descentralização de receitas	15/03/2016	30/03/2016
Suporte a novas aplicações	01/04/2016	15/04/2016
Levantamento de bugs e soluções de bugs no Shak	15/04/2016	30/04/2016
Evolução da gerência de certificados	01/05/2016	15/05/2016
Levantamento de bugs e soluções de bugs no Shak	15/05/2016	30/05/2016
Finalizar documento	01/05/2016	30/06/2016



**Obrigado!**