

Documentação Infra IntraJur 2.0

Neste documento teremos:

- Descrição de instalação de ambiente de *desenvolvimento/servidor* para CakePHP 3.x
- Descrição de instalação de SQLserver em ambiente Linux (UBUNTU 16.04.5)
- Descrição de Instalação de servidor MySQL em ambiente Linux
- Configuração de odbc, odbc FreeTDS para comunicação com SQLserver em PHP 7.x
- Configuração de SSL em servidor local Linux

Conhecimentos desejados:

- Ambiente Linux;
- SQLserver, MySQL ou qualquer outro SGBD
- Apache + PHP

Sumário:

Configuração de Máquina ou VR:	3
Configuração mínima;	3
Instalação de distro Ubuntu 16.04;	3
 Instalação e configuração de Ambiente:	 4
Instalação e configuração Apache2 + PHP 7.x;	4
Instalação do Microsoft SQLserver;	5
Instalação do MySQL;	7
Configurando HTTPS;	7
Clonando repositório GIT;	10

Configuração de Máquina ou VR:

Inicialmente precisaremos fazer a instalação de uma distro Linux. Recomendamos fortemente o uso da mesma distro deste documento([Ubuntu 16.04.5](#)). Também é possível realizar este procedimento em uma máquina virtual, para isto, também será detalhado a configuração da mesma.

Configuração mínima;

Usaremos o [Oracle VM VirtualBox](#) em sua versão 5.2.22 para esse procedimento, no entanto, nossa recomendação é o uso da versão mais recente deste software.

A configuração da máquina necessitará de pelo menos **2GB de memória RAM**, processador com pelo menos **2 threads** e um **HD com 20GB** de espaço no mínimo, lembre-se de colocar a configuração da rede em modo Bridge. Feito isso teremos uma máquina virtual apta a realizar os procedimentos que serão descritos abaixo.

Instalação de distro Ubuntu 16.04;

Com a imagem (.ISO) da versão do [Ubuntu 16.04](#) já baixada, (no caso de usar uma máquina virtual, lembre-se de montar a imagem no disco no drive e ajustar a sequência de boot, no caso de uma máquina real, [grave um DVD](#) com a imagem ou faça um [pendrive bootável](#)) e a forma de boot ajustada, podemos iniciar com a instalação.

Ao iniciar a instalação, selecione a linguagem desejada, usaremos 'Português do Brasil' e clique em continuar. É possível instalar atualizações agora mas evite atualizar a versão do sistema acima do 16.04 ou a instalação de softwares de terceiros, principalmente se a máquina tiver a finalidade de servidor. No '*Tipo de instalação*', como usaremos o disco todo, deixaremos no padrão '*Apagar disco e reinstalar o Ubuntu*', depois de confirmar selecione seu UTC, Layout do teclado e os dados do usuário, concluído essa etapa, basta aguardar a conclusão da instalação.

Instalação e configuração de Ambiente:

Instalação e configuração Apache2 + PHP 7.x;

Agora com o OS instalado e funcional, vamos iniciar o procedimento de instalação do Apache2. No terminal:

```
sudo apt install apache2
sudo apt install git
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php7.2 php7.2-gd
sudo apt install php7.2-intl php7.2-mbstring php7.2-odbc php7.2-mysql php7.2-sqlite3 php7.2-xml
sudo phpenmod mbstring
sudo phpenmod intl
sudo phpenmod odbc
sudo phpenmod sqlite3
sudo phpenmod xml
sudo phpenmod gd
sudo a2enmod rewrite
sudo service apache2 restart
```

Seguindo todas as linhas, agora já será possível abrir <http://localhost>. Para testar o PHP e seus módulos vamos voltar ao terminal e ir até a pasta padrão do Apache2: `/var/www/html/`.

```
cd /var/www/html
sudo pico phpinfo.php
```

Agora com o PICO aberto colocaremos um código em PHP para que possamos identificar se todos os módulos que instalamos está devidamente ativado.

```
<?php
phpinfo();
?>
```

Para salvar o novo arquivo no sistema PICO, colocamos essas combinações de teclas respectivamente:

```
CTRL + X
S
<ENTER>
```

Agora podemos voltar ao navegador, no endereço: <http://localhost/phpinfo.php>. Nesta tela será possível ver todas as informações do PHP. Todas as extensões, arquivos de configuração e suas respectivas localizações.

Uma extensão que não instalamos, porém, se faz muito importante e já deveria vir com a compilação do PHP7.2 é o 'openssl', caso encontre uma tabela específica para essa extensão no phpinfo significa que já está instalado e funcionando. Caso não esteja:

```
sudo apt install openssl
sudo phpenmod ssl
php -i | grep -i openssl
sudo service apache2 restart
```

Agora precisamos configurar o modulo rewrite do Apache2. Para isso, ainda no terminal, vamos até a pasta das configurações de sites:

```
sudo pico /etc/apache2/apache2.conf
```

Neste arquivo precisamos editar uma configuração chamada 'AllowOverride' que deve está configurado como 'None', e devemos alterar para 'All', você pode encontrar várias configurações dessas, mas apenas o 'Directory' que estiver apontando para a localização de seu servidor deve ser alterado:

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    #altera linha abaixo
    AllowOverride All
    Require all granted
</Directory>
```

Usando o PICO, salve o arquivo, agora basta reiniciar o Apache2, no terminal:

```
sudo service apache2 reload
sudo service apache2 restart
```

Instalação do Microsoft SQLserver;

Para realizar a [instalação do SGBD SQLserver 2017](#) no linux, vamos seguir o tutorial disponibilizado pela Microsoft. Onde a primeira coisa a fazer é adicionar o repositório do SQLserver:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
sudo add-apt-repository "$(wget -qO-
https://packages.microsoft.com/config/ubuntu/16.04/mssql-server-2017.list)"
```

Agora atualizaremos a lista de pacotes do Linux, agora com o repositório da Microsoft e iniciaremos a instalação do SQLserver.

```
sudo apt update
sudo apt install -y mssql-server OR sudo apt-get install -y -t xenial mssql-server
```

Agora com o SQLserver 2017 já instalado, podemos realizar sua configuração e inicialização do serviço:

```
sudo /opt/mssql/bin/mssql-conf setup
```

Aqui selecionaremos qual a versão do SQLserver que instalaremos, em nosso caso, escolhemos a Developer (2) e selecionaremos a linguagem do servidor, em nosso caso usaremos Inglês(1) para o bom funcionamento dos campos de data dentro do CakePHP. Por último informaremos a senha do usuário SA, o administrador do SQLserver. Para testar o servidor usaremos o seguinte comando:

```
systemctl status mssql-server
```

Agora com o SQLserver funcionando, instalaremos o freeTDS, ele fará a comunicação com o SQLserver:

```
sudo -i  
apt-get install freetds-bin freetds-common tdsodbc odbcinst unixodbc unixodbc-dev  
mv /etc/odbcinst.ini /etc/odbcinst.ini.bak  
cp /usr/share/tdsodbc/odbcinst.ini /etc/
```

Para testar a conexão com o banco de dados pode usar o seguinte comando, substituindo o host, user, e password pelos respectivos dados(-D é opcional, caso não tenha um banco criado no servidor pode apagar essa parte):

```
tsql -H my_server_host -p 1433 -U my_user -P my_password -D my_database
```

Agora precisamos instalar o drive que usará essa configuração, para tal precisamos da extensão 'sqlsrv' do PHP, sua instalação envolve o PEAR. Seguiremos os seguintes comandos no terminal:

```
sudo apt install php7.2-dev g++ php-pear  
sudo pecl install sqlsrv  
sudo pecl install pdo_sqlsrv  
sudo -i  
php echo extension=sqlsrv.so >> `php --ini | grep "Scan for additional .ini files" | sed -e "s/.*/: \s*//" /20-sqlsrv.ini  
a2dismod mpm_event  
a2enmod mpm_prefork  
echo "extension=pdo_sqlsrv.so" >> /etc/php/7.2/apache2/conf.d/30-pdo_sqlsrv.ini  
echo "extension=sqlsrv.so" >> /etc/php/7.2/apache2/conf.d/20-sqlsrv.ini  
exit  
sudo service apache2 restart
```

Agora para concluir a configuração da conexão do CakePHP com o SQLserver, basta realizar a [instalação do msodbcsql17](#). Siga os comandos no terminal:

```
sudo -i
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
curl https://packages.microsoft.com/config/ubuntu/14.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list
exit
sudo apt update
sudo ACCEPT_EULA=Y apt-get install msodbcsql17
sudo apt install unixodbc-dev
```

Instalação do MySQL;

A instalação do servidor MySQL é feita toda ela pelo terminal. Este documento fará uso da versão 5.7 e é recomendado que se use a mesma. Fato importante é que no ato da instalação será solicitado a senha do usuário 'root'. Para tal, siga os comandos:

```
sudo apt install mysql-server
```

Configurar o módulo segurança do MySQL não é essencial caso já o tenha feito na instalação, se não, ou caso esteja criando um ambiente de servidor coloque o comando abaixo no terminal e siga as instruções:

```
sudo mysql_secure_installation
```

Configurando HTTPS;

A configuração do servidor Apache2 com o módulo SSL é essencial para o CakePHP 3.x, primeiramente vamos precisar [criar certificados SSL](#), voltando ao terminal:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Será necessário passar algumas configurações, abaixo as questões com suas respectivas respostas:

```
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:SaoPaulo
Locality Name (eg, city) []:SP
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ASPREV
Organizational Unit Name (eg, section) []:ASPREV
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:local@localhost.com
```

Devemos também criar um Diffie-Hellman, para isso usaremos o comando no terminal, pode demorar alguns minutos:

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

Agora precisamos criar um arquivo de configuração para o SSL, para isso vamos até o diretório de configurações do Apache2:

```
cd /etc/apache2/conf-available
sudo pico /etc/apache2/conf-available/ssl-params.conf
```

Neste arquivo vamos colocar o seguinte conteúdo:

```
# from https://cipherli.st/
# and https://raymii.org/s/tutorials/Strong_SSL_Security_On_Apache2.html

SSLCipherSuite ECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3
SSLHonorCipherOrder On
# from https://cipherli.st/ # and https://r # Disable preloading HSTS for now. You can use the
# commented out header line that includes
# the "preload" directive if you understand the implications.
#Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains"
Header always set X-Frame-Options SAMEORIGIN
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLSessionTickets Off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"
```

Salve o arquivo e saia do PICO. Agora precisamos modificar o arquivo de configuração SSL no Apache2, primeiramente vamos criar um backup do arquivo atual e depois vamos aplicar nossas mudanças:

```
sudo cp /etc/apache2/sites-available/default-ssl.conf
/etc/apache2/sites-available/default-ssl.conf.bak

sudo gedit /etc/apache2/sites-available/default-ssl.conf
```


Como já criamos um backup podemos alterar o conteúdo, já sem os comentários, altere o conteúdo para o seguinte alterando o ServerName para o IP de sua máquina:

```
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin local@localhost.com
    ServerName server_domain_or_IP
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>
    BrowserMatch "MSIE [2-6]" \
      nokeepalive ssl-unclean-shutdown \
      downgrade-1.0 force-response-1.0
  </VirtualHost>
</IfModule>
```

Agora vamos habilitaremos os seguintes módulos mod_ssl e mod_headers, siga os comandos no terminal:

```
sudo a2enmod ssl
sudo a2enmod headers
sudo a2ensite default-ssl
sudo a2enconf ssl-params
sudo service apache2 reload
sudo service apache2 restart
```

Se tudo der certo, podemos testar com o seguinte comando:

```
sudo apache2ctl configtest
```

O resultado esperado é:

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1.
Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

Clonando repositório GIT;

Neste ponto, concluímos todas as instalações e configurações, restando apenas a criação de uma cópia do repositório. Para isso vamos até a pasta responsável pelo public_html do Apache2 que por padrão é '/var/www/html/'(essa [pasta padrão do Apache2](#) pode ser alterada) pelo terminal e passaremos o seguinte código:

```
sudo git clone http://10.0.20.66/sj/intrajur2.git
```

Depois de clonar o servidor, será necessário dar permissões nas pastas 'vendor', 'logs', 'tmp' do CakePHP usando o seguinte comando no terminal:

```
sudo mkdir tmp
```

```
sudo mkdir logs
```

```
sudo chmod -R 777 .
```

EX.:

```
sudo chmod -R 777 intrajur2/*
```