

# Manual de Instalação do e-cidade

## Instalando o e-cidade a partir dos fontes no Ubuntu

O e-cidade destina-se a informatizar a gestão dos Municípios Brasileiros de forma integrada. Esta informatização contempla a integração entre os entes municipais: Prefeitura Municipal, Câmara Municipal, Autarquias, Fundações e outros. A economia de recursos é somente uma das vantagens na adoção do e-cidade, além da liberdade de escolha dos fornecedores e garantia de continuidade do sistema, uma vez apoiado pelo Ministério do Planejamento.

## Instalação do Ubuntu Server 18.04 LTS ou 20.04 LTS

O Ubuntu Server é a versão do sistema operacional da Canonical voltada para servidores. Como o uso em servidores exige um tempo maior de suporte, a versão indicada para o uso com esta documentação é a Ubuntu Server 18.04 LTS ou 20.04 LTS.

## Obtendo a mídia de instalação

A mídia de instalação pode ser obtida diretamente do site da Canonical e gravada em CD, DVD ou pendrive. No site do [Ubuntu](#) esta disponível a versão mais recente, a versão 18.04 LTS disponível em [Ubuntu Bionic](#) e a 20.04 LTS esta disponível em [Ubuntu Focal](#).

## Instalando o Ubuntu Server 18.04 LTS

O processo de instalação do sistema operacional não é coberto por esta documentação, mas recomendamos a instalação mínima padrão, em "pt\_BR" Português do Brasil, somente com o software "OpenSSH server".

**\*\*Dica:\*\*** Maiores informações sobre a instalação e personalização do sistema operacional podem ser obtidas [:aqui](#)

## Editando os arquivos de configuração

Durante o processo de configuração descrito nesta documentação, é necessário editar e modificar diversos arquivos de texto. Para realizar estas modificações, sugerimos a utilização do editor "nano", mas você pode usar qualquer outro editor de texto de sua preferência.

**\*\*Dica:\*\*** Mais informações sobre o editor de textos "nano" podem ser obtidas em: <http://www.nano-editor.org>

## Instalação do PostgreSQL 9.5

PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de código aberto sem custo de licença. Este é o programa que vai gerenciar e

armazenar o banco de dados relacional utilizado pelo e-cidade.

**\*\*Dica:\*\*** Mais informações sobre o PostgreSQL podem ser obtidas em: <http://www.postgresql.org>

## Configurando o repositório

Para instalar o PostgreSQL 9.5 e suas dependências, utilizaremos o gerenciador de pacotes `apt`, que nos prove as últimas atualizações dos pacotes requeridos através de um canal seguro e confiável.

**Crie o arquivo pgdg.list através dos comandos abaixo.**

```
ubuntu@ubuntu:~$ RELEASE=$(cat /etc/lsb-release | grep DISTRIB_CODENAME | cut -c18-30)
ubuntu@ubuntu:~$ sudo touch /etc/apt/sources.list.d/pgdg.list
ubuntu@ubuntu:~$ echo "deb http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}-pgdg main" |
sudo tee /etc/apt/sources.list.d/pgdg.list
```

**\*\*Dica:\*\*** Mais informações sobre a configuração do repositório podem ser obtidas em: <http://www.postgresql.org/download/linux/ubuntu>

**Baixe e importe a chave de assinatura do repositório através do comando:**

```
ubuntu@ubuntu:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7FCC7D46ACCC4CF8
```

**\*\*Dica:\*\*** Mais informações sobre o gerenciador de pacotes "apt" podem ser obtidas em: <https://help.ubuntu.com/18.04/serverguide/package-management.html>

## 2.2 Ajustando o idioma

**\*\*Atenção:\*\*** Este procedimento só é necessário se você não instalou o sistema operacional em "pt\_BR - Português do Brasil" e tinha internet neste momento.

```
ubuntu@ubuntu:~$ sudo apt-get update
ubuntu@ubuntu:~$ sudo apt-get install language-pack-gnome-pt language-pack-pt-base myspell-pt
myspell-pt-br wbrazilian wportuguese
```

## Ajustando o encoding

### Configurando geração de locales

```
ubuntu@ubuntu:~$ sudo nano /usr/share/i18n/locales/pt_BR
```

Localize o bloco de configuração **LC\_COLLATE** e **END LC\_COLLATE**, e adicione dentro do bloco as seguintes linhas:

```
reorder-after <U00A0>
<U0020><CAP>;<CAP>;<CAP>;<U0020>
reorder-end
```

## Gerando locais

Para criação do cluster do e-cidade precisamos ter o locale **pt\_BR ISO-8859-1** disponível no sistema, para gerar os mesmo utilize os comandos abaixo.

```
ubuntu@ubuntu:~$ sudo localedef -i pt_BR -c -f ISO-8859-1 -A /usr/share/locale/locale.alias pt_BR
ubuntu@ubuntu:~$ sudo locale-gen pt_BR
ubuntu@ubuntu:~$ sudo dpkg-reconfigure locales
```

Marque neste ultimo comando "todas as opções" com a tecla space e de um TAB para ir até o OK

## Instalando o PostgreSQL 9.5

Para realizar a instalação do PostgreSQL 9.5 através do gerenciador de pacotes execute os comandos:

```
ubuntu@ubuntu:~$ sudo apt-get update
ubuntu@ubuntu:~$ sudo apt-get install postgresql-9.5 postgresql-client-9.5 postgresql-contrib-9.5
```

## Criando e configurando o cluster

O e-cidade utiliza encoding **LATIN1 (ISO-8859-1)** no cluster do PostgreSQL 9.5 Por isso vamos parar, remover e recriar o cluster quer será utilizado pela aplicação.

**\*\*Dica:\*\*** Mais informações sobre a configuração do cluster:

<http://www.postgresql.org/docs/9.5/static/sql-cluster.html>

**Vamos parar e remover o cluster criado pelo processo de instalação do PostgreSQL através do comando:**

```
ubuntu@ubuntu:~$ sudo pg_dropcluster --stop 9.5 main
```

**Vamos criar o novo cluster do banco de dados como LATIN1 usando o comando:**

```
ubuntu@ubuntu:~$ sudo pg_createcluster -u postgres -g postgres -e LATIN1 --
locale="pt_BR.ISO-8859-1" --lc-collate="pt_BR.ISO-8859-1" 9.5 ecidade
```

**Para permitir o acesso de qualquer local ao banco de dados é necessário modificar o arquivo `/etc/postgresql/9.5/ecidade/pg_hba.conf` usando o comando:**

```
ubuntu@ubuntu:~$ sudo nano /etc/postgresql/9.5/ecidade/pg_hba.conf
```

Localize e altere as linhas do arquivo de configuração que não estão comentadas (`#`` na frente da linha), substituindo os valores correspondentes por `trust`, conforme a tabela abaixo.

```
local | all | postgres | | trust
local | all | | | trust
host | all | all | 127.0.0.1/32 | trust
host | all | all | ::1/128 | trust
```

**Para a correta integração com o e-cidade, é necessário alterar algumas configurações do cluster criado através do comando:**

```
ubuntu@ubuntu:~$ sudo nano /etc/postgresql/9.5/ecidade/postgresql.conf
```

**\*\*Atenção:\*\*** Os demais parâmetros, não descritos aqui, não precisam ser alterados.

Localize e altere as linhas do arquivo de configuração, descomentando-as se necessário (remover o caractere `#`` do início da linha), substituindo os valores correspondentes conforme a indicação abaixo.

```
listen_addresses = '*'
max_connections = 20
bytea_output = 'escape'
max_locks_per_transaction = 256
default_with_oids = on
escape_string_warning = off
standard_conforming_strings = off
```

**Inicie o servidor PostgreSQL através do comando:**

```
ubuntu@ubuntu:~$ sudo /etc/init.d/postgresql start
```

**Verifique o resultado da configuração usando o comando:**

```
ubuntu@ubuntu:~$ psql -U postgres -h localhost -l
```

A saída do comando deve ser parecida com os valores abaixo.

Nome	Dono	Codificação	Collate	Ctype	Privilégios de acesso
postgres	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	
template0	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	=c/postgres + postgres=CTc/postgres
template1	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	=c/postgres + postgres=CTc/postgres

## Criando a estrutura de banco de dados para o e-cidade

Vamos criar os usuários necessários para o funcionamento do e-cidade através dos comandos:

**\*\*Atenção:\*\*** Em ambientes de produção, nunca utilize senhas que tenham o mesmo nome da conta de usuário. Mais informações sobre segurança de senhas podem ser obtidas em:  
<http://cartilha.cert.br/senhas/>.

```
ubuntu@ubuntu:~$ psql -U postgres -c "CREATE ROLE ecidade WITH SUPERUSER LOGIN  
PASSWORD 'ecidade';"  
ubuntu@ubuntu:~$ psql -U postgres -c "CREATE ROLE dbseller WITH LOGIN PASSWORD  
'dbseller';"  
ubuntu@ubuntu:~$ psql -U postgres -c "CREATE ROLE plugin WITH LOGIN PASSWORD 'plugin';"
```

Em seguida, utilize o comando abaixo para criar o banco de dados:

```
ubuntu@ubuntu:~$ psql -U postgres -c "CREATE DATABASE ecidade OWNER ecidade;"
```

Agora a saída deste comando

```
ubuntu@ubuntu:~$ psql -U postgres -h localhost -l
```

Deve ficar assim

Nome	Dono	Codificação	Collate	Ctype	Privilégios de acesso
ecidade	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	
postgres	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	
template0	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	=c/postgres + postgres=CTc/postgres
template1	postgres	LATIN1	pt_BR.ISO-8859-1	pt_BR.ISO-8859-1	=c/postgres + postgres=CTc/postgres

## Instalação do Apache

Apache é o servidor web mais usado em sistemas ubuntu. Servidores Web são usados para servir páginas Web que, normalmente, são solicitadas pelos computadores clientes através de navegador web, como o Firefox, Chromium, etc.

**\*\*Dica:\*\*** Mais informações sobre o Apache podem ser obtidas em: <http://www.apache.org/>.

## Instalando o Apache

Para realizar a instalação do servidor web Apache através do gerenciador de pacotes, execute o comando:

```
ubuntu@ubuntu:~$ sudo apt-get install apache2
```

## Configurando a diretiva VirtualHost

**Faça uma cópia de segurança do arquivo de configuração padrão, que é criado pelo Apache durante o processo de instalação através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/000-default.conf.dist
```

**Para a correta integração com o e-cidade, é necessário adicionar alguns parâmetros ao arquivo de configuração do `VirtualHost` através do comando:**

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

**\*\*Atenção:\*\*** Os demais parâmetros, não descritos aqui, não precisam ser alterados.

Adicione as linhas que não existirem e edite as existentes dentro do bloco `VirtualHost`, conforme é exibido abaixo.

```
<VirtualHost *:80>
...
AddDefaultCharset ISO-8859-1
LimitRequestLine 16382
LimitRequestFieldSize 16382
Timeout 12000
<Directory /var/www/html>
    Options -Indexes +FollowSymLinks +MultiViews
    AllowOverride All
    Require all granted
</Directory>
...
</VirtualHost>
```

**\*\*Dica:\*\*** Mais informações sobre o VirtualHost podem ser obtidas em: <http://httpd.apache.org/docs/current/vhosts/examples.html>

## Habilitar o módulo rewrite

Para habilitar o módulo rewrite execute o comando:

```
ubuntu@ubuntu:~$ sudo a2enmod rewrite
```

e reinicie o apache

```
ubuntu@ubuntu:~$ sudo /etc/init.d/apache2 restart
```

## Instalação do PHP 5.6

PHP é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

**\*\*Dica:\*\*** Mais informações sobre o PHP podem ser obtidas em: <http://www.php.net/>.

## Instalando o PHP 5.6

Para realizar a instalação do PHP 5.6 através do gerenciador de pacotes, precisamos adicionar o repositório PPA do mesmo, sendo assim execute os comandos:

```
ubuntu@ubuntu:~$ sudo add-apt-repository -y ppa:ondrej/php
ubuntu@ubuntu:~$ sudo apt-get update
ubuntu@ubuntu:~$ sudo apt-get install php5.6 php5.6-bcmath php5.6-bz2 php5.6-cli php5.6-common php5.6-curl php5.6-gd php5.6-interbase php5.6-json php5.6-mbstring php5.6-mcrypt php5.6-pgsql php5.6-soap php5.6-sqlite3 php5.6-xml php5.6-xmlrpc php5.6-zip unzip zip php5.6-common php5.6-interbase libapache2-mod-php5.6
```

## Configurando o PHP 5.6

**Vamos criar um diretório para armazenar os registros gerados pelas transações realizadas pelo PHP através dos comandos:**

```
ubuntu@ubuntu:~$ sudo mkdir /var/www/html/tmp
ubuntu@ubuntu:~$ sudo chown -R www-data:www-data /var/www/html/tmp
ubuntu@ubuntu:~$ sudo chmod -R 777 /var/www/html/tmp
ubuntu@ubuntu:~$ sudo mkdir /var/www/tmp
ubuntu@ubuntu:~$ sudo chown -R www-data:www-data /var/www/tmp
ubuntu@ubuntu:~$ sudo chmod -R 777 /var/www/tmp
```

**Para a correta integração com o e-cidade, é necessário alterar algumas configurações do PHP através do comando:**

```
ubuntu@ubuntu:~$ sudo nano /etc/php/5.6/apache2/php.ini
```

**\*\*Atenção:\*\*** Os demais parâmetros, não descritos aqui, não precisam ser alterados.

Localize e altere as linhas do arquivo do arquivo de configuração, descomentando-as se necessário (remover o caractere `;` do início da linha), substituindo os valores correspondentes conforme a indicação abaixo.

```
error_reporting = E_ALL & ~E_NOTICE & ~E_DEPRECATED
register_argc_argv = On
post_max_size = 200M
upload_max_filesize = 200M
default_socket_timeout = 60000
max_execution_time = 60000
max_input_time = 60000
memory_limit = 512M
display_errors = Off
log_errors = On
error_log = /var/log/apache2/php_errors.log
session.gc_maxlifetime = 7200
ignore_repeated_errors = On
short_open_tag = On
date.timezone = "America/Sao_Paulo"
```

E também o do cliente PHP

```
ubuntu@ubuntu:~$ sudo nano /etc/php/5.6/cli/php.ini
```

**\*\*Atenção:\*\*** Os demais parâmetros, não descritos aqui, não precisam ser alterados.

Localize e altere as linhas do arquivo do arquivo de configuração, descomentando-as se necessário (remover o caractere `;` do início da linha), substituindo os valores correspondentes conforme a indicação abaixo.

```
error_reporting = E_ALL & ~E_NOTICE & ~E_DEPRECATED
register_argc_argv = On
post_max_size = 200M
upload_max_filesize = 200M
default_socket_timeout = 60000
max_execution_time = 60000
```



```
max_input_time = 60000
memory_limit = 512M
display_errors = Off
log_errors = On
error_log = /var/log/php/php_errors.log
session.gc_maxlifetime = 7200
ignore_repeated_errors = On
short_open_tag = On
date.timezone = "America/Sao_Paulo"
```

Crie os arquivos e sete as permissões

```
ubuntu@ubuntu:~$ sudo touch /var/log/apache2/php_errors.log
ubuntu@ubuntu:~$ sudo chmod 777 /var/log/apache2/php_errors.log
ubuntu@ubuntu:~$ sudo mkdir /var/log/php
ubuntu@ubuntu:~$ sudo touch /var/log/php/php_errors.log
ubuntu@ubuntu:~$ sudo chmod 777 /var/log/php/php_errors.log
```

**Reinicie o Apache usando o comando:**

```
ubuntu@ubuntu:~$ sudo /etc/init.d/apache2 restart
```

## Instalação do LibreOffice

O LibreOffice é uma suite de escritório livre compatível com as principais suítes de escritório do mercado. Oferece todas as funções esperadas de uma suite profissional: editor de textos, planilha, apresentação, editor de desenhos e banco de dados. No e-cidade, o LibreOffice é utilizado para exportar documentos baseados em modelos de formulários.

**\*\*Dica:\*\*** Mais informações sobre o LibreOffice podem ser obtidas em: <https://pt-br.libreoffice.org/>.

## Instalando o LibreOffice

Para realizar a instalação do LibreOffice através do gerenciador de pacotes, execute o comando:

```
ubuntu@ubuntu:~$ sudo apt-get install libreoffice-writer python3-uno openjdk-8-jre ttf-mscorefonts-installer
```

## Configurando o LibreOffice

Vamos criar o arquivo de inicialização do LibreOffice através do comando

```
ubuntu@ubuntu:~$ sudo nano /etc/init.d/e-cidade_libreoffice-server
```

Adicione as linhas indicadas abaixo.

```

#!/bin/bash
### BEGIN INIT INFO
# libreoffice.org headless server script
# Provides: DBSeller (equipe.infraestrutura@dbseller.com.br)
# Required-Start:
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# X-Interactive: true
# Short-Description: start/stop/restart libreoffice.org headless server script
### END INIT INFO
OOo_HOME=/usr/bin
SOFFICE_PATH=$OOo_HOME/soffice
PIDFILE=/var/run/e-cidade_libreoffice-server/libreoffice-server.pid
set -e
[ -d /var/run/e-cidade_libreoffice-server ] || mkdir -p /var/run/e-cidade_libreoffice-server
case "$1" in
    start)
        if [ -e $PIDFILE ]; then
            echo "LibreOffice headless foi iniciado."
            sleep 2
            exit
        fi
        echo "Iniciando LibreOffice headless"
        $SOFFICE_PATH --headless --nologo --nofirststartwizard --
accept="socket,host=127.0.0.1,port=8100;urp" & > /dev/null 2>&1
        touch $PIDFILE
        ;;
    stop)
        if [ -e $PIDFILE ]; then
            echo "Parando LibreOffice headless."
            if [[ "$(ps aux | grep -v grep | grep soffice.bin | awk '{print $2}')" != "" ]]; then
                kill -9 $(ps aux | grep -v grep | grep soffice.bin | awk '{print $2}')
            fi
            rm -rf $PIDFILE
            exit
        fi
        echo "O LibreOffice headless não está executando."
        exit
        ;;
    restart)
        /etc/init.d/e-cidade_libreoffice-server stop
        /etc/init.d/e-cidade_libreoffice-server start
        exit
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac

```

```
exit 0
```

## Adicionar o LibreOffice a inicialização

Para adicionar a carga do serviço na inicialização do sistema execute os seguintes comandos:

```
ubuntu@ubuntu:~$ sudo chmod +x /etc/init.d/e-cidade_libreoffice-server
ubuntu@ubuntu:~$ sudo update-rc.d e-cidade_libreoffice-server defaults
```

Em seguida, inicie o serviço manualmente através do comando:

```
ubuntu@ubuntu:~$ sudo /etc/init.d/e-cidade_libreoffice-server start
```

**\*\*Dica:\*\*** Caso você utilize outra distribuição ubuntu, ou instalou o LibreOffice de outra forma, deve verificar o arquivo "/var/www/html/e-cidade/bin/oo2pdf/oo2pdf.sh" para ajustar corretamente o caminho do LibreOffice na variável "OOFFICE", e do Python na variável "OOPYTHON" para que a integração do e-cidade com o LibreOffice funcione corretamente.

## Instalação do e-cidade

### Obtendo o pacote de instalação

O pacote de instalação pode ser obtido diretamente do site do Portal do Software Público. No endereço da comunidade do [e-cidade](#) estão disponíveis as últimas versões e suas atualizações. Como estamos realizando uma nova instalação, você deve baixar o pacote `completo`, um exemplo de nome do pacote seria: `e-cidade-xxxx-x-linux.completo.tar.bz2`.

### Configurando o servidor

**Vamos criar um usuário administrativo para o e-cidade, neste exemplo criaremos o usuário `dbseller` através do comando:**

```
ubuntu@ubuntu:~$ sudo useradd -d /home/dbseller -g www-data -G sudo,adm,cdrom,dip,plugdev -k /etc/skel -m -s /bin/bash dbseller
```

**Em seguida, vamos definir uma senha para este novo usuário usando o comando:**

```
ubuntu@ubuntu:~$ sudo passwd dbseller
```

**\*\*Dica:\*\*** Neste exemplo, defina a senha como o mesmo nome de usuário, como "dbseller".

**\*\*Atenção:\*\*** Em ambientes de produção, nunca utilize senhas que tenham o mesmo nome da conta de usuário. Mais informações sobre segurança de senhas podem ser obtidas em: <http://cartilha.cert.br/senhas>

**Também devemos corrigir a máscara de criação de arquivos modificando os arquivos através dos comandos:**

```
ubuntu@ubuntu:~$ sudo nano /etc/login.defs
```

**\*\*Atenção:\*\*** Os demais parâmetros, não descritos aqui, não precisam ser alterados.

Localize e altere a linha do arquivo de configuração, descomentando-a se necessário (remover o caractere `#` do início da linha), substituindo o valor correspondente conforme a indicação abaixo.

```
UMASK      002
```

Em seguida, edite o arquivo de configuração de variáveis do apache com o comando abaixo:

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/envvars
```

Adicione a linha abaixo no final do arquivo.

```
umask 002
```

## **Instalando o e-cidade**

Vá até o diretório onde está o pacote compactado do e-cidade

**Descompacte o pacote do e-cidade através do comando:**

```
ubuntu@ubuntu:~$ tar xjf e-cidade-xxxx-x-linux.completo.tar.bz2
```

**Copie o diretório do e-cidade para o diretório raiz do Apache usando o comando:**

```
ubuntu@ubuntu:~$ sudo cp -r e-cidade-xxxx-x-linux.completo/e-cidade /var/www/html
```

Lembre de trocar os x pela versão baixada

## **Configurando o e-cidade**

**Crie o arquivo de configuração de acesso ao banco de dados através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidade/libs/db_conn.php.dist /var/www/html/e-cidade/libs/db_conn.php
```

Utilize o seguinte comando para editar o arquivo de configuração:

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidade/libs/db_conn.php
```

Em seguida, atualize as configurações conforme o exemplo abaixo.

```
$DB_USUARIO = 'ecidade';  
$DB_SENHA = 'ecidade';  
$DB_SERVIDOR = 'localhost';  
$DB_PORTA = '5432';  
$DB_PORTA_ALT = '5432';  
$DB_BASE = 'ecidade';
```

**Crie o arquivo de configuração de acesso dos plugins através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidade/config/plugins.json.dist /var/www/html/e-cidade/config/plugins.json
```

Utilize o seguinte comando para editar o arquivo de configuração:

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidade/config/plugins.json
```

Em seguida, atualize as configurações conforme o exemplo abaixo.

```
{  
  "AcessoBase" : {  
    "usuario" : "plugin",  
    "senha" : "plugin"  
  }  
}
```

**Configure o servidor de envio de e-mails do e-cidade através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidade/libs/config.mail.php.dist /var/www/html/e-cidade/libs/config.mail.php  
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidade/libs/config.mail.php
```

Em seguida, atualize as configurações, com os dados do seu servidor de e-mail, conforme o exemplo abaixo.

```
$sClass = 3;  
$sSsl = ''; // qual o tipo de ssl q sera utilizados ( ex: tls), por padrao é vazio  
$bAuth = true; // autenticacao habilitada  
$sUser = 'exemplo@exemplo.com.br';  
$sPass = 'senhaexemplo';  
$sHost = 'smtp.exemplo.com.br';  
$sPort = '25';
```

### **Faça a copia do arquivo que fala as extensões permitidas**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidade/config/require_extensions.xml.dist  
/var/www/html/e-cidade/config/require_extensions.xml
```

### **Faça a carga do `schema` do e-cidade através do comando:**

```
ubuntu@ubuntu:~$ psql -U ecidade -d ecidade -f e-cidade-xxxx-x-linux.completo/sql/e-cidade-xxxx-  
x.sql 2> /tmp/erros.txt
```

Após a carga, execute o comando abaixo para otimizar as consultas:

```
ubuntu@ubuntu:~$ psql -U ecidade -d ecidade -c "VACUUM ANALYZE VERBOSE;"
```

### **Ajuste as permissões dos diretórios através dos comandos:**

```
ubuntu@ubuntu:~$ sudo chown -R www-data.www-data /var/www/html/e-cidade  
ubuntu@ubuntu:~$ sudo chmod -R 775 /var/www/html/e-cidade  
ubuntu@ubuntu:~$ sudo chmod -R 777 /var/www/html/e-cidade/tmp
```

### **Agendando a migração da auditoria**

#### **Crie os diretórios e defina as permissões**

```
ubuntu@ubuntu:~$ sudo mkdir /var/www/html/e-cidade/integracao_externa/auditoria/log  
ubuntu@ubuntu:~$ sudo chmod -R 777 /var/www/html/e-cidade/integracao_externa/auditoria/log
```

#### **Edite o crontab do usuário `dbseller` através do comando:**

```
dbseller@ubuntu:~$ crontab -e
```

Em seguida, adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script para migracao do account para nova estrutura "db_auditoria"
*/30 * * * * cd /var/www/html/e-cidade/integracao_externa/auditoria; ./auditoria_migracao.sh 0 10 1
> log/auditoria_migracao.log 2>&1
*/10 * * * * cd /var/www/html/e-cidade/integracao_externa/auditoria; php -q
auditoria_adiciona_fila.php > log/auditoria_adiciona_fila.log 2>&1
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

### Agendando a verificação do gerenciador de tarefas

**Para criar o script de verificação execute os comandos:**

```
ubuntu@ubuntu:~$ sudo mkdir -p /opt/dbseller/scripts
ubuntu@ubuntu:~$ sudo nano /opt/dbseller/scripts/ativadorGerenciadorTarefasEcidade.sh
```

Em seguida, adicione as linhas indicadas abaixo.

```
#!/bin/bash
sDiretorioEcidade="/var/www/html/e-cidade"
if [ ! -d $sDiretorioEcidade ]; then
echo "Diretorio $sDiretorioEcidade nao existe"
exit 1;
fi
sTarefa=$(ps aux | grep "con4_gerenciadortarefas002.php" | grep -v "grep");
if [ "$sTarefa" == "" ]; then
echo "Agendador de tarefas parou! ZBX:ERRO:AGENDADOR-PARADO" >
$sDiretorioEcidade/ativadorGerenciadorTarefasEcidade.log
cd $sDiretorioEcidade;
/usr/bin/php -q FrontIntegracaoExterna.php --executable con4_gerenciadortarefas002.php >
tmp/log_gerenciadortarefas.log 2> tmp/erros_gerenciadortarefas.log &
sleep 10
sTarefa=$(ps aux | grep "con4_gerenciadortarefas002.php" | grep -v "grep");
if [ "$sTarefa" != "" ]; then
echo "Agendador de tarefas iniciado! ZBX:AGENDADOR-INICIADO" >
$sDiretorioEcidade/ativadorGerenciadorTarefasEcidade.log
fi
fi
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

**Antes de agendar o script de verificação, ajuste a permissão utilizando o seguinte comando:**

```
ubuntu@ubuntu:~$ sudo chmod +x /opt/dbseller/scripts/ativadorGerenciadorTarefasEcidade.sh
```

Em seguida, edite o crontab do usuário `root` através do comando:

```
ubuntu@ubuntu:~$ sudo crontab -e
```

E adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script de verificacao do gerenciador de tarefas
*/15 * * * * cd /opt/dbseller/scripts && su -s /bin/bash www-data -c "bash
ativadorGerenciadorTarefasEcidade.sh"
```

## **Acessando o e-cidade**

Após finalizar os procedimentos de instalação e configuração, abra um navegador e acesse o e-cidade através do endereço IP ou nome do servidor. EX: [http://ip\\_do\\_servidor/e-cidade](http://ip_do_servidor/e-cidade), utilize como nome de usuário `dbseller`, e deixe o campo senha em branco.

Atenção: Certifique-se que o bloqueador de pop-up esteja permitindo o acesso ao endereço IP ou nome do servidor em que o e-cidade foi instalado.

## **Instalação do e-cidadeOnline**

### **Instalando o e-cidadeOnline**

**Copie o diretório do e-cidadeOnline para o diretório raiz do Apache usando o comando:**

```
ubuntu@ubuntu:~$ sudo cp -r e-cidade-xxxx-x-linux.completo/e-cidadeonline /var/www/html
```

### **Configurando o e-cidadeOnline**

**Crie o arquivo de configuração de acesso ao banco de dados através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidadeonline/libs/db_conn.php.dist /var/www/html/e-
cidadeonline/libs/db_conn.php
```

Utilize o seguinte comando para editar o arquivo de configuração:

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidadeonline/libs/db_conn.php
```



Em seguida, atualize as configurações conforme o exemplo abaixo.

```
$DB_INSTITUICAO = 1;  
$DB_SERVIDOR = 'localhost';  
$DB_BASEDADOS = 'ecidade';  
$DB_USUARIO = 'ecidade';  
$DB_SENHA = 'ecidade';  
$DB_PORTA = '5432';
```

**Configure o servidor de envio de e-mails do e-cidadeOnline através do comando:**

```
ubuntu@ubuntu:~$ sudo cp /var/www/html/e-cidadeonline/libs/config.mail.php.dist /var/www/html/e-cidadeonline/libs/config.mail.php  
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidadeonline/libs/config.mail.php
```

Em seguida, atualize as configurações, com os dados do seu servidor de e-mail, conforme o exemplo abaixo.

```
$sClass = 3;  
$sSsl = ''; // qual o tipo de ssl q sera utilizados ( ex: tls), por padrao é vazio  
$bAuth = true; // autenticacao habilitada  
$sUser = 'exemplo@exemplo.com.br';  
$sPass = 'senhaexemplo';  
$sHost = 'smtp.exemplo.com.br';  
$sPort = '25';
```

**Ajuste as permissões dos diretórios através dos comandos:**

```
ubuntu@ubuntu:~$ sudo chown -R www-data.www-data /var/www/html/e-cidadeonline  
ubuntu@ubuntu:~$ sudo chmod -R 775 /var/www/html/e-cidadeonline  
ubuntu@ubuntu:~$ sudo chmod -R 777 /var/www/html/e-cidadeonline/tmp
```

### **Acessando o e-cidadeOnline**

Após finalizar os procedimentos de instalação e configuração, abra um navegador e acesse o e-cidadeOnline através do endereço IP ou nome do servidor. EX: [http://ip\\_do\\_servidor/e-cidadeonline](http://ip_do_servidor/e-cidadeonline), utilize como nome de usuário `dbseller`, e deixe o campo senha em branco.

### **Configurando a carga de integrações**

**\*\*Dica:\*\*** Preste atenção aos horários de agendamento do crontab de múltiplas integrações. Procure verificar o tempo de duração da execução de cada integração, ajustando o horário de execução conforme a necessidade.

**\*\*Dica:\*\*** Sempre que possível, execute manualmente a linha agendada para certificar o correto

funcionamento do agendamento.

## Configurando a integração de carga do cubo do BI

**Crie o arquivo de configuração de acesso FTP para a carga do cubo do BI através do comando:**

```
ubuntu@ubuntu:~$ sudo cp -a /var/www/html/e-cidade/libs/db_cubo_bi_config.php.dist  
/var/www/html/e-cidade/libs/db_cubo_bi_config.php
```

Utilize o seguinte comando para editar o arquivo de configuração:

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidade/libs/db_cubo_bi_config.php
```

Em seguida, atualize as configurações conforme o exemplo abaixo.

```
$configCuboBi['ftp']['server'] = "endereco.ip.do.servidor";  
$configCuboBi['ftp']['usuario'] = "nome-usuario-ftp";  
$configCuboBi['ftp']['senha'] = "senha-do-usuario-ftp";  
$configCuboBi['ftp']['diretorio'] = "diretorio-de-destino";  
$configCuboBi['ftp']['passive_mode'] = true
```

## Configurando a integração de carga do Webservice HORUS BASICO

**Crie o arquivo de configuração de através do comando:**

**\*\*Dica:\*\*** Mais informações sobre o Webservice HORUS BASICO podem ser obtidas em:  
<http://portalsaude.saude.gov.br/index.php/o-ministerio/principal/leia-mais-o-ministerio/218-sctie-raiz/daf-raiz/ceaf-sctie/qualifarsus-raiz/eixo-informacao/11-eixo-informacao/8508-envio-de-dados-webservice>

```
ubuntu@ubuntu:~$ sudo cp -a /var/www/html/e-cidade/libs/db_config_horus.php.dist  
/var/www/html/e-cidade/libs/db_config_horus.php
```

Utilize o seguinte comando para editar o arquivo de configuração:

```
ubuntu@ubuntu:~$ sudo nano /var/www/html/e-cidade/libs/db_config_horus.php
```

Em seguida, atualize as configurações conforme o exemplo abaixo.

**\*\*Atenção:\*\*** Consulte a documentação das URLs no portalsaude.saude.gov.br antes de utilizar a integração.

```
$sUrlWebService = "http://189.28.128.37/horus-ws-basico/RecebeDadosWS?wsdl";  
$sLogin = "nome-usuario-horus";  
$sSenha = "senha-do-usuario-horus";
```

## Configurando as integrações de limpeza de cadastros duplos CGM e CGS

Para agendar a limpeza de cadastros duplos CGM, edite o crontab do usuário `dbseller` através do comando:

```
dbseller@ubuntu:~$ crontab -e
```

Em seguida, adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script para limpeza de cadastro duplos CGM  
0 22 * * * cd /var/www/html/e-cidade/ ; /usr/bin/php FrontIntegracaoExterna.php --executable  
duplos.php > duplos.log 2>&1
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

## Configurando a integração GISS

O script de execução da integração GISS possui dois parametros de configuração:

1) Se o programa for agendado sem nenhum parâmetro, ou com o parametro `1`, a integração processará somente os registros alterados.

```
.../iss4_gissonline002.php 1 > /dev/null 2>&1
```

2) Se o programa for agendado com o parâmetro `2`, a integração processará todos os registros.

```
.../iss4_gissonline002.php 2 > /dev/null 2>&1
```

Edite o crontab do usuário `dbseller` através do comando:

```
dbseller@ubuntu:~$ crontab -e
```

Em seguida, adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script para execução da integração GISS  
0 22 * * * cd /var/www/html/e-cidade/ ; /usr/bin/php FrontIntegracaoExterna.php --dir  
integracao_externa/gissonline --executable integracao_externa/gissonline/iss4_gissonline002.php >  
/dev/null 2>&1
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

## Configurando a integração SIGCORP

**Edite o crontab do usuário `dbseller` através do comando:**

```
dbseller@ubuntu:~$ crontab -e
```

Em seguida, adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script para execução da integração SIGCORP
0 22 * * * cd /var/www/html/e-cidade/ ; /usr/bin/php FrontIntegracaoExterna.php --dir
integracao_externa/sigcorp --executable integracao_externa/sigcorp/iss4_sigcorp001.php > /dev/null
2>&1
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

## Configurando a integração WEBISS

**Edite o crontab do usuário `dbseller` através do comando:**

```
dbseller@ubuntu:~$ crontab -e
```

Em seguida, adicione o seguinte bloco ao final do arquivo:

```
# DBSeller - Agendamento do script para execução da integração WEBISS
0 22 * * * cd /var/www/html/e-cidade/ ; /usr/bin/php FrontIntegracaoExterna.php --executable
integracao_externa/webiss/iss4_webiss001.php > /dev/null 2>&1
```

**\*\*Atenção:\*\*** Lembre-se de ajustar o caminho para o diretório do e-cidade conforme o ambiente em que foi instalado.

## Configurando a integração débitos

O script de execução da integração débitos necessita de um arquivo de configuração para ser executado. Para criar o script de configuração, execute os comandos abaixo:

```
ubuntu@ubuntu:~$ cd /var/www/html/e-cidade/integracao_externa/debitos/lib
ubuntu@ubuntu:~$ cp -a debitos.conf.dist debitos.conf
```

Em seguida edite o arquivo através do comando:

```
ubuntu@ubuntu:~$ nano debitos.conf
```

Informe o valor das variáveis conforme as descrições abaixo:

```
# Permite definir a periodicidade com que a manutencao da tabela debitos
# sera executada. O valores permitidos sao: -1=desativa, 0=todos ou dia da semana (1=segunda,
# 2=terca ... 7=domingo).
#
# Exemplo:
# DEBITOS_PERIODO="7"
#
DEBITOS_PERIODO=""

# Permite definir o numero de dias no passado para manter os registros na debitos.
#
# Exemplo:
# DEBITOS_DIAS_MANTER="14"
#
DEBITOS_DIAS_MANTER=""

# Permite definir o codigo da(s) instituicao(oes) que deve(em) ser gerada debitos. Se houver
# mais de uma instituicao, os seus codigos devem ser separados por espaço.
#
# Exemplo:
# DEBITOS_INSTITUICOES="1"
# DEBITOS_INSTITUICOES="1 3"
#
DEBITOS_INSTITUICOES=""
```