

# Instalando lggr com syslog-ng no FreeBSD 10.3

Por Bruno Ferreira.

Baseado no [how-to oficial](#) do criador do projeto **Kai Kretschmann**.

O autor do projeto fez um tutorial em inglês para Ubuntu Linux, apesar de ser baseado no documento oficial este guia difere um pouco do original. Use com cautela ;).

## [Pré-Requisitos]

- syslog-ng (com o módulo de extensão syslog-ng-mod-sql )
- stunnel (opcional)
- mysql
- apache

A instalação e configuração do lggr, pronuncia-se logger, difere entre servidor e cliente. Este guia explica as duas instalações.

## Servidor

### [Instalando os pré-requisitos]

#### [Apache]

```
root@SRV001:/usr/home/brunof # pkg install apache24-2.4.23
```

```
Updating FreeBSD repository catalogue...
```

```
FreeBSD repository is up-to-date.
```

```
All repositories are up-to-date.
```

```
Updating database digests format: 100%
```

```
The following 10 package(s) will be affected (of 0 checked):
```

```
New packages to be INSTALLED:
```

```
  apache24: 2.4.23
```

```
  libxml2: 2.9.3
```

```
  expat: 2.2.0
```

```
  apr: 1.5.2.1.5.4
```

```
  gdbm: 1.12
```

```
  indexinfo: 0.2.4
```

```
  gettext-runtime: 0.19.8.1
```

```
  db5: 5.3.28_4
```

```
  perl5: 5.20.3_13
```

```
  pcre: 8.39
```

```
Number of packages to be installed: 10
```

```
The process will require 136 MiB more space.
```

```
33 MiB to be downloaded.
```

Proceed with this action? [y/N]: y  
Fetching apache24-2.4.23.txz: 100% 5 MiB 1.6MB/s 00:03  
Fetching libxml2-2.9.3.txz: 100% 789 KiB 807.4kB/s 00:01  
Fetching expat-2.2.0.txz: 100% 100 KiB 102.1kB/s 00:01  
Fetching apr-1.5.2.1.5.4.txz: 100% 402 KiB 411.8kB/s 00:01  
Fetching gdbm-1.12.txz: 100% 144 KiB 147.7kB/s 00:01  
Fetching indexinfo-0.2.4.txz: 100% 5 KiB 5.0kB/s 00:01  
Fetching gettext-runtime-0.19.8.1.txz: 100% 148 KiB 151.6kB/s 00:01  
Fetching db5-5.3.28\_4.txz: 100% 12 MiB 2.6MB/s 00:05  
Fetching perl5-5.20.3\_13.txz: 100% 13 MiB 2.8MB/s 00:05  
Fetching pcre-8.39.txz: 100% 1 MiB 1.2MB/s 00:01  
Checking integrity... done (0 conflicting)  
[1/10] Installing indexinfo-0.2.4...  
[1/10] Extracting indexinfo-0.2.4: 100%  
[2/10] Installing gettext-runtime-0.19.8.1...  
[2/10] Extracting gettext-runtime-0.19.8.1: 100%  
[3/10] Installing expat-2.2.0...  
[3/10] Extracting expat-2.2.0: 100%  
[4/10] Installing gdbm-1.12...  
[4/10] Extracting gdbm-1.12: 100%  
[5/10] Installing db5-5.3.28\_4...  
[5/10] Extracting db5-5.3.28\_4: 100%  
[6/10] Installing libxml2-2.9.3...  
[6/10] Extracting libxml2-2.9.3: 100%  
[7/10] Installing apr-1.5.2.1.5.4...  
[7/10] Extracting apr-1.5.2.1.5.4: 100%  
[8/10] Installing perl5-5.20.3\_13...  
[8/10] Extracting perl5-5.20.3\_13: 100%  
[9/10] Installing pcre-8.39...  
[9/10] Extracting pcre-8.39: 100%  
[10/10] Installing apache24-2.4.23...  
===> Creating groups.  
Using existing group 'www'.  
===> Creating users  
Using existing user 'www'.  
[10/10] Extracting apache24-2.4.23: 100%  
Message from perl5-5.20.3\_13:  
The /usr/bin/perl symlink has been removed starting with Perl 5.20.  
For shebangs, you should either use:

```
#!/usr/local/bin/perl
```

or

```
#!/usr/bin/env perl
```

The first one will only work if you have a /usr/local/bin/perl,

the second will work as long as perl is in PATH.

Message from apache24-2.4.23:

To run apache www server from startup, add `apache24_enable="yes"` in your `/etc/rc.conf`. Extra options can be found in startup script.

Your hostname must be resolvable using at least 1 mechanism in `/etc/nsswitch.conf` typically DNS or `/etc/hosts` or apache might have issues starting depending on the modules you are using.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

- apache24 default build changed from static MPM to modular MPM
- more modules are now enabled per default in the port
- icons and error pages moved from WWWDIR to DATADIR

If build with modular MPM and no MPM is activated in `httpd.conf`, then `mpm_prefork` will be activated as default MPM in `etc/apache24/modules.d` to keep compatibility with existing php/perl/python modules!

Please compare the existing `httpd.conf` with `httpd.conf.sample` and merge missing modules/instructions into `httpd.conf`!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```
root@SRV001:/usr/home/brunof # echo apache24_enable="yes" >> /etc/rc.conf
```

Por fim deve ser configurado a autenticação básica do apache. Para isso inclua na diretiva `Directory` as linhas:

```
AllowOverride AuthConfig
Order allow,deny
Allow from all
```

E Configure um usuário para o lggr

```
htpasswd -c /usr/local/www/apache24/data/lggr-master/.htpasswd usuario
```

Altere a permissão para que somente o usuario do apache tenha acesso.

```
chown www:www /usr/local/www/apache24/data/lggr-master/.htpasswd
chmod 0660 /usr/local/www/apache24/data/lggr-master/.htpasswd
```

Ainda no diretório raiz do lggr, cria o arquivo `.htaccs` com as seguintes linhas:

```
AuthType Basic
AuthName "Restricted Access"
AuthUserFile /usr/local/www/apache24/data/lggr-master/.htpasswd
Require user usuario
```

Restarte o apache e faça o teste de autenticação.

Caso dê algum erro veja no arquivo de logs o que pode ser `/var/log/httpd-error.log`

## [PHP]

```
pkg install mod_php56-5.6.23 php56-5.6.23 php56-gd-5.6.23 php56-curl-5.6.23 php56-json-5.6.23 php56-ldap-5.6.23 php56-mysql-5.6.23 php56-mysqli-5.6.23 php56-iconv-5.6.23 php56-redis-2.2.8 php56-gettext-5.6.23
```

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

rehash

Mude o directory index para:  
DirectoryIndex index.php index.html

Adicione ao final do arquivo de configuração do apache

```
<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
```

Restarte no apache

Para testar se o php está ok basta criar um arquivo test.php com o conteúdo:  
<?php phpinfo(); ?>

Abra no navegador e veja se aparece as informações do php:  
<http://ipdoseuservidor/test.php>

Se estiver tudo ok vamos para o próximo passo, caso contrário revise com calma os passo acima.

## [REDIS]

```
pkg install redis-3.0.7
root@SRV001:/usr/local/www/apache24/data/lggr-master # echo redis_enable="YES" >>
/etc/rc.conf
/usr/local/etc/rc.d/redis start
```

## [Bibliotecas de terceiros]

O lgger usa várias bibliotecas de terceiros para criar uma interface harmoniosa e moderna, porém para quem não está acostumado pode ser um pouco confuso. Você pode usá-las localmente ou externamente via url.

Atualmente o logger faz uso dos projetos WEB:

- bootstrap
- chartjs
- jqcloud

- jquery
- jqueryui
- timepicker

Vá pra o diretório raiz do projeto. Ex: root@SRV001:/usr/local/www/apache24/data/lggr-master

wget -c [https://lggr.io/wp-content/uploads/2015/06/lggr\\_contrib.tar.gz](https://lggr.io/wp-content/uploads/2015/06/lggr_contrib.tar.gz)

Extraia o conteúdo:

tar -xvzf lggr\_contrib.tar.gz

e acerte as permissões

chown -R www:www \*

## [Mysql]

```
root@SRV001:/usr/home/brunof # pkg install mysql57-client-5.7.13_1
```

Updating FreeBSD repository catalogue...

FreeBSD repository is up-to-date.

All repositories are up-to-date.

The following 4 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:

mysql57-client: 5.7.13\_1

liblz4: 131

libevent2: 2.0.22\_1

libedit: 3.1.20150325\_2,1

Number of packages to be installed: 4

The process will require 52 MiB more space.

2 MiB to be downloaded.

Proceed with this action? [y/N]: y

Fetching mysql57-client-5.7.13\_1.txz: 100% 2 MiB 944.1kB/s 00:02

Fetching liblz4-131.txz: 100% 75 KiB 76.6kB/s 00:01

Fetching libevent2-2.0.22\_1.txz: 100% 258 KiB 264.2kB/s 00:01

Fetching libedit-3.1.20150325\_2,1.txz: 100% 121 KiB 123.7kB/s 00:01

Checking integrity... done (0 conflicting)

[1/4] Installing liblz4-131...

[1/4] Extracting liblz4-131: 100%

[2/4] Installing libevent2-2.0.22\_1...

[2/4] Extracting libevent2-2.0.22\_1: 100%

[3/4] Installing libedit-3.1.20150325\_2,1...

[3/4] Extracting libedit-3.1.20150325\_2,1: 100%

[4/4] Installing mysql57-client-5.7.13\_1...

[4/4] Extracting mysql57-client-5.7.13\_1: 100%

Message from mysql57-client-5.7.13\_1:

\*\*\*\*\*

This is the mysql CLIENT without the server.

for complete server and client, please install databases/mysql57-server

\*\*\*\*\*

```
root@SRV001:/usr/home/brunof # pkg install mysql57-server-5.7.13_1
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 1 package(s) will be affected (of 0 checked):
```

```
New packages to be INSTALLED:
  mysql57-server: 5.7.13_1
```

```
Number of packages to be installed: 1
```

```
The process will require 141 MiB more space.
14 MiB to be downloaded.
```

```
Proceed with this action? [y/N]: y
Fetching mysql57-server-5.7.13_1.txz: 100% 14 MiB 2.9MB/s 00:05
Checking integrity... done (0 conflicting)
[1/1] Installing mysql57-server-5.7.13_1...
==> Creating groups.
Creating group 'mysql' with gid '88'.
==> Creating users
Creating user 'mysql' with uid '88'.
[1/1] Extracting mysql57-server-5.7.13_1: 100%
Message from mysql57-server-5.7.13_1:
```

\*\*\*\*\*

Remember to run `mysql_upgrade` the first time you start the MySQL server after an upgrade from an earlier version.

Initial password for first time use of MySQL is saved in `$HOME/.mysql_secret` ie. when you want to use `"mysql -u root -p"` first you should see password in `/root/.mysql_secret`

\*\*\*\*\*

\*\*\*\*\*

The default location for `my.cnf` has changed from `"/var/db/mysql/my.cnf"` to `"/usr/local/etc/mysql/my.cnf"`. Existing `my.cnf` files must be merged manually with the new default and moved to the new location. To continue using the `my.cnf` file at the old location, set `"mysql_optfile"` in `/etc/rc.conf` to point to the location of the existing `my.cnf` file.

\*\*\*\*\*

```
root@SRV001:/usr/home/brunof # echo mysql_enable="YES" >> /etc/rc.conf
```

```
root@SRV001:/usr/home/brunof # /usr/local/etc/rc.d/mysql-server start
Starting mysql.
```

```
root@SRV001:/usr/home/brunof # cat /$HOME/.mysql_secret
# Password set for user 'root@localhost' at 2016-07-12 13:41:35
c5o!QgXe!rQa
```

### [syslog-ng]

```
root@SRV001:/usr/home/brunof/lggr-master # pkg install syslog-ng-3.7.3_4
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 7 package(s) will be affected (of 0 checked):
```

New packages to be INSTALLED:

```
syslog-ng: 3.7.3_4
glib: 2.46.2_2
python27: 2.7.12
libffi: 3.2.1
libiconv: 1.14_9
eventlog: 0.2.12_1
json-c: 0.12_2
```

Number of packages to be installed: 7

The process will require 90 MiB more space.  
14 MiB to be downloaded.

Proceed with this action? [y/N]:

```
Fetching syslog-ng-3.7.3_4.txz: 100% 503 KiB 515.0kB/s 00:01
Fetching glib-2.46.2_2.txz: 100% 3 MiB 1.4MB/s 00:02
Fetching python27-2.7.12.txz: 100% 10 MiB 2.2MB/s 00:05
Fetching libffi-3.2.1.txz: 100% 35 KiB 36.2kB/s 00:01
Fetching libiconv-1.14_9.txz: 100% 591 KiB 605.5kB/s 00:01
Fetching eventlog-0.2.12_1.txz: 100% 14 KiB 14.7kB/s 00:01
Fetching json-c-0.12_2.txz: 100% 38 KiB 38.7kB/s 00:01
Checking integrity... done (0 conflicting)
[1/7] Installing libffi-3.2.1...
[1/7] Extracting libffi-3.2.1: 100%
[2/7] Installing python27-2.7.12...
[2/7] Extracting python27-2.7.12: 100%
[3/7] Installing libiconv-1.14_9...
[3/7] Extracting libiconv-1.14_9: 100%
[4/7] Installing glib-2.46.2_2...
[4/7] Extracting glib-2.46.2_2: 100%
No schema files found: doing nothing.
[5/7] Installing eventlog-0.2.12_1...
```

[5/7] Extracting eventlog-0.2.12\_1: 100%

[6/7] Installing json-c-0.12\_2...

[6/7] Extracting json-c-0.12\_2: 100%

[7/7] Installing syslog-ng-3.7.3\_4...

[7/7] Extracting syslog-ng-3.7.3\_4: 100%

Message from python27-2.7.12:

```
=====
=====
```

Note that some standard Python modules are provided as separate ports as they require additional dependencies. They are available as:

bsddb	databases/py-bsddb
gdbm	databases/py-gdbm
sqlite3	databases/py-sqlite3
tkinter	x11-toolkits/py-tkinter

```
=====
=====
```

Message from syslog-ng-3.7.3\_4:

syslog-ng is now installed! To replace FreeBSD's standard syslogd (/usr/sbin/syslogd), complete these steps:

1. Create a configuration file named /usr/local/etc/syslog-ng.conf (a sample named syslog-ng.conf.sample has been included in /usr/local/etc). Note that this is a change in 2.0.2 version, previous ones put the config file in /usr/local/etc/syslog-ng/syslog-ng.conf, so if this is an update move that file in the right place

2. Configure syslog-ng to start automatically by adding the following to /etc/rc.conf:

```
syslog_ng_enable="YES"
```

3. Prevent the standard FreeBSD syslogd from starting automatically by adding a line to the end of your /etc/rc.conf file that reads:

```
syslogd_enable="NO"
```

4. Shut down the standard FreeBSD syslogd:

```
kill `cat /var/run/syslog.pid`
```

5. Start syslog-ng:

```
/usr/local/etc/rc.d/syslog-ng start
```



```
root@SRV001:/usr/home/brunof/lggr-master # echo syslog_ng_enable="YES" >>
/etc/rc.conf
root@SRV001:/usr/home/brunof/lggr-master # echo syslogd_enable="NO" >> /etc/rc.conf
root@SRV001:/usr/home/brunof/lggr-master # kill `cat /var/run/syslog.pid` && echo morto
morto
root@SRV001:/usr/home/brunof/lggr-master # /usr/local/etc/rc.d/syslog-ng start
Starting syslog_ng.
```

### [lggr.io]

Beleza, se conseguiu executar tudo até aqui basta baixar a última versão do lggr.io.  
Atenção: Sempre verifique antes no endereço: <https://lggr.io/source/>

```
root@SRV001:/usr/home/brunof/ # cd /usr/local/www/apache24/data/
root@SRV001:/usr/local/www/apache24/data/ # wget -c --no-check-certificate
https://github.com/kkretsch/lggr/archive/master.zip
root@SRV001:/usr/local/www/apache24/data/ # unzip master.zip
root@SRV001:/usr/local/www/apache24/data # cd lggr-master/
```

## [Configuração]

### [mysql]

Crie um banco de dados chamado **logger**  
ps. Não esqueça de alterar a senha do root.

```
root@localhost [(none)]> SET PASSWORD = PASSWORD(xxx);
Query OK, 0 rows affected, 1 warning (0.15 sec)
root@localhost [(none)]> CREATE DATABASE logger;
Query OK, 1 row affected (0.02 sec)
```

Depois de criar a base execute o script db.sql para criar a estrutura da base, localizado em doc/db.sql.

```
root@SRV001:/usr/local/www/apache24/data/lggr-master/doc # ls
db.sql      user.sql
root@SRV001:/usr/local/www/apache24/data/lggr-master/doc # mysql -u root -p < db.sql
Enter password:
```

Ok, se você verificar foi criado 5 tabelas.

```
root@localhost [(none)]> use logger
Database changed
root@localhost [logger]> show tables
-> ;
```

```
+-----+
| Tables_in_logger |
+-----+
| archived      |
| lasthour      |
```

```
| newlogs      |
| today        |
| week         |
+-----+
5 rows in set (0.02 sec)
```

Depois disse você terá que criar pelo menos dois usuário Mysql com permissões diferentes: Um para escrita dentro da tabela e outro para leitura pela aplicação web e outro usado internamente para limpeza. (logger, logviewer e loggeradmin). Vamos usar o script doc/user.sql que vem incluso.

Dê um vi no arquivo e altere as senhas 'xxx'

```
root@SRV001:/usr/local/www/apache24/data/lggr-master/doc # vi user.sql
```

Lembre-se use sempre senhas fortes.

Mude os valores que achar pertinente nos arquivos *inc/config\_class.php* e *inc/adminconfig\_class.php*.

Por exemplo, eu traduzi e mudei a lingua de alemão para português no *config\_class.php*:

```
// Set your preferred language en_US or de_DE
    $this->setLocale('pt_BR');
```

Mude a senha root assim que possível e teste.

Verifique se os usuarios foram criados corretamente e estão se autenticandousando o comando

```
mysql -u logger -p
```

### [syslog-ng]

```
root@SRV001:/usr/local/www/apache24/data/lggr-master # mkfifo /tmp/mysql.pipe
```

Crie o script:

```
vi /usr/local/etc/rc.d/040.mysql-syslog-ng.sh
```

```
#!/bin/sh
```

```
#Script para criar canal de comunicacao com o BD MySQL.
```

```
#Add este script na inicializacao.Ele deve ser executado DEPOIS do script de inicializacao do MySQL.
```

```
if [ ! -e /tmp/mysql.pipe ]
```

```
then
```

```
mkfifo /tmp/mysql.pipe
```

```
fi
```

```
while [ -e /var/log/mysql.pipe ]
```

```
do
```

```
mysql -u root --password=xxx logger < /tmp/mysql.pipe > /dev/null
```

```
done
```

Adicione no final do arquivo syslog-ng.conf as seguintes linhas:

```
vi /usr/local/etc/syslog-ng.conf
```

```
source s_net {
    tcp( ip("0.0.0.0") port(514) max-connections(20) log-iw-size(2000) );
};

destination d_newmysql { pipe("/tmp/mysql.pipe" template("INSERT INTO logger (date,
facility, level, host, program, p
id, message) VALUES ('${R_YEAR}-${R_MONTH}-${R_DAY}
${R_HOUR}:${R_MIN}:${R_SEC}', '$FACILITY', '$LEVEL', '$HOST', '$P
ROGRAM', '$PID', '$MSGONLY');n") template-escape(yes) ); };
    log {
        source(s_net); destination(d_newmysql);
    };
root@SRV001:/usr/local/www/apache24/data/lggr-master # /usr/local/etc/rc.d/syslog-ng
restart
```

Com essa configuração o syslog-ng logará todas as mensagens TCP recebidas via rede (source s\_net). No exemplo na diretiva log, eu habilitei grava no Mysql apenas mensagens oriundas via rede mas nada impede que você coloque outras fontes como a fonte local de logs. Além disso você pode limitar na aplicação a origem dos logs mudando de 0.0.0.0 para um ip específico ou via firewall na porta 514.

## Apache

Lá em cima nós já extraímos o conteúdo do logger no diretório de páginas do Apache. Agora é só ajustar a conexão com a base de dados de acordo com sua configuração editando o arquivo *inc/config\_class.php*. Use a conta de somente leitura do mysql (se não lembrar, abra o arquivo *doc/user.sql*).

```
root@SRV001:/usr/local/www/apache24/data/lggr-master # vi inc/config_class.php
```

Agora você precisa proteger a interface gui, o método utilizado no logger é via autenticação básica do apache usando o arquivo *.htaccess*.

```
htpasswd -c /var/www/webuser lggr
```

E entre com uma senha segura.

Para aumentar a segurança o ideal é usar SSL/https, assim se algum se alguém estiver sniffando o meio não vai capturar sua credenciais em texto claro.

Por último dê permissão de escrita para o usuário web no diretório cache.

## PHP

Usar pelo menos a versão 5.4. O desenvolvedor utiliza a 5.6.14.

## stunnel (Opicional)

Criar um tunel é algo mais complexo. Você terá que criar um infraestrutura de CA com chaves e certificados, distribuí-las entre os cliente e configurar corretamente.

Para informações mais detalhadas dê uma olhada em [snippet.wiki](http://snippet.wiki).

Para dar algumas dicas:

Habilite o stunnel `/usr/local/etc/defaults/stunnel` e crie o arquivo de configuração `/usr/local/etc/stunnel/stunnel.conf`:

```
CAfile = /usr/local/etc/stunnel/cacert.pem
```

```
CAspath = /usr/local/etc/stunnel/certs/
```

```
cert = /usr/local/etc/stunnel/logserver_cert.pem
```

```
key = /usr/local/etc/stunnel/logserver_nopwd_key.pem
```

```
verify = 3
```

```
debug = 5
```

```
[5140]
```

```
accept = 10.10.10.10:5140
```

```
connect = 127.0.0.1:514
```

O IP 10.10.10.10 é seu endereço externo. Agora o stunnel deve escutar na porta 5140 para conexões externas e encaminhar a conexão descriptografada para a porta local tcp do syslog.

## Client

Basta configurar o apontamento normal de um syslog, ip e porta.

### syslog-ng (I just dont understand why this is necessary =/ )

Criar o arquivo `/etc/syslog-ng/conf.d/10lggr.conf` com

```
destination d_net { tcp("127.0.0.1" port(514) log_fifo_size(1000)); };
```

```
log { source(s_src); destination(d_net); };
```

## stunnel

Você deve ter esse arquivos:

- cacert.pem (the personal root certificate of the logging server structure)
- client\_cert.pem (the signed certificate request)
- client\_npwd\_key.pem (the private key for that client without password)

Reference to that files within a new configuration file `/etc/stunnel/syslog.conf`:

```
client = yes
```

```
CAfile = /etc/stunnel/cacert.pem
```

```
cert = /etc/stunnel/client_cert.pem
```

```
key = /etc/stunnel/client_npwd_key.pem
```

```
verify = 2
```

```
debug = 5
```

```
[5140]
```

```
accept = 127.0.0.1:514
```

```
connect = 10.10.10.10:5140
```

Where the IP 10.10.10.10 is the public external IP of your logging server again.

## Cron jobs

Pra quem tem pouco espaço e quer armazenar por um curto período as mensagens, basta rodar o arquivo `admin/cron.php` de hora em hora ou diariamente. Você pode criar o arquivo `/etc/cron/lggr` contendo:

```
30 1 * * * www-data /usr/bin/php /var/www/lggr/admin/cron.php
```

Por padrão fica armazenado as entradas das últimas 4 semanas. Se não for o que você quer basta ajustar o arquivo `cron.php` para suas necessidades.

## Locales

Para usar o logger com traduções o servidor deve ter instalado o arquivo `locales` correto. O comando `locale -a` deve listar pelo menos `en_US.utf8` e `de_DE.utf8`, fique à vontade para adicionar suas traduções.