**Manual de instalação, configuração e utilização do LattesService**

**Autor: Rivaldo Sampaio de Oliveira Júnior**

**Mestrando do Programa de Pós-Graduação em   
Tecnologia, Gestão e Saúde Ocular da   
Universidade Federal de São Paulo  
  
São Paulo, 19 de dezembro de 2020**

Sumário

[1. INTRODUÇÃO 1](#_Toc59282532)

[2. LattesService 1](#_Toc59282533)

[2.1. Pré-Requisitos 1](#_Toc59282534)

[3. INSTALAÇÃO E CONFIGURAÇÃO 2](#_Toc59282535)

[3.1. Instalação das dependências das bibliotecas *PyGraphviz* e *PyTidyLib* 2](#_Toc59282536)

[3.2. Instalação do *PIP* 3](#_Toc59282537)

[3.3. Instalação do *Virtualenv* 3](#_Toc59282538)

[3.4. Instalação das bibliotecas *Python* utilizadas pelo *scriptLattes* e *LattesService* 3](#_Toc59282539)

[3.5. Configuração do *scriptLattes* 3](#_Toc59282540)

[3.6. Configuração do *LattesService* 4](#_Toc59282541)

[3.6.1. Segurança na execução do *LattesService* 4](#_Toc59282542)

[3.6.2. Bancos de dados 4](#_Toc59282543)

[3.6.3. Regionalização 5](#_Toc59282544)

[3.6.4. Cache de dados no cliente 5](#_Toc59282545)

[*3.6.5.* Execução do *scriptLattes* 5](#_Toc59282546)

[3.6.6. Automatização do *LattesService* 6](#_Toc59282547)

[3.7. Criação das tabelas no PostgresSQL 7](#_Toc59282548)

[3.8. Geração dos arquivos estáticos 8](#_Toc59282549)

[3.9. Verificação do funcionamento do Django 8](#_Toc59282550)

[3.10. Modificação das permissões dos arquivos 8](#_Toc59282551)

[3.11. Exemplo de configuração do servidor HTTP 8](#_Toc59282552)

[4. ATUALIZAÇÃO DA VERSÃO DO *scriptLattes* 11](#_Toc59282553)

[5. UTILIZAÇÃO DO CLIENTE HTTP 12](#_Toc59282554)

# INTRODUÇÃO

Este manutal tem por objetivo orientar o processo de implantação do *LattesService* em ambiente *Ubuntu®* com o servidor *HTTP* *Apache 2*®.

O *LattesService* é um *software* desenvolvido em linguagem *Python®* e *framework* *Django®* com o propósito de disponibilizar, como *webservice*, dados de produção científica obtidos da Plataforma Lattes por meio do *scriptLattes*.

O *scriptLattes,* principal peça que compõe o LattesService*,* é um software que extrai e compila os dados de produção científica de currículos da Plataforma Lattes, a partir de uma lista pré-definida de pesquisadores. Desta forma, a partir dos arquivos *HTML* gerados pelo *scriptLattes*, é possível visualizar os dados de forma agrupada, o que proporciona uma visão da produção de todo o grupo de pesquisadores. Foi desenvolvido em linguagem *Python* pelos pesquisadores Jesús P. Mena-Chalco e Roberto M. Cesar-Jr em 2005 e, por ser de código aberto, vem sendo aprimorado colaborativamente por diversos pesquisadores que buscam disponibilizar novas funcionalidades.

# LattesService

Uma vez implantado, o *LattesService* automatiza todo o processo de obtenção e disponibilização de dados de produção cientifica oriundos da Plataforma Lattes. Para isso, foi organizado em 3 módulos (ou *apps*, terminologia utilizada pela comunidade *Django®*): **institucional**, **curriculo** e **api**.

O módulo **institucional** é responsável por acessar o banco de dados da instituição para obter a lista de orientadores e gerar o arquivo que é utilizado pelo *scriptLattes* para buscar os currículos na Plataforma Lattes.

O módulo **curriculo** executa o *scriptLattes*, acessa os dados em memória e os grava no banco de dados.

Tanto o módulo **institucional** quanto o módulo **currículo** são executados automaticamente a partir de um agendamento prévio utilizando o *cron* do *Ubuntu*. O agendamento é configurado por meio do próprio *LattesService*.

O módulo **api** disponibiliza 2 *endpoints* para que os *websites* dos Programas de Pós-Graduação possam obter e consumir os dados de produção científica armazenados no banco de dados do *LattesService*.

## Pré-Requisitos

O *LattesService* foi testado em ambiente operacional com os seguintes recursos:

* *Sistema Operacional Ubuntu® Server 16.04.6 LTS (Xenial Xerus) 64 bits;*
* *Servidor HTTP Apache 2.4.18* e a biblioteca *libapache2-mod-wsgi;*
* *PostgreSQL 10;*
* *Python 2.7* (já é instalado por padrão no *Ubuntu* *Server 16.04*) e as bibliotecas:
  + *PIP;*
  + *Virtualenv;*
  + dependências das bibliotecas *PyGraphviz* e *PyTidyLib*: *python-dev*, *graphviz*, *libgraphviz-dev*, *pkg-config, libtidy-dev;*
  + demais bibliotecas *Python* listadas no arquivo requirements.txt que acompanha o *LattesService*.

Neste manual, não será abordado o processo de instalação e configuração do Ubuntu, Apache e PostgreSQL, pois envolve questões de segurança e desempenho que somente profissionais já capacitados e experientes terão.

Em relação ao *Apache*, é necessária a instalação do pacote *libapache2-mod-wsgi* para que possa se comunicar com as aplicações do *Django*. Um exemplo de como configurar o *Apache* para servir *Django* com *WGSI* será demonstrado em 3.10

No *PostgreSQL* deverá ser criado um usuário e um banco de dados que serão usados pelo *LattesService* a fim de armazenar os dados extraídos do Currículo Lattes.

# INSTALAÇÃO E CONFIGURAÇÃO

Clone ou faça o download do LattesService a partir do repositório <https://github.com/rivaldo-so-junior/LattesService>.

Transfira os arquivos para um diretório de sua preferência.

Durante os testes e elaboração deste manual, os arquivos foram transferidos de um computador com Windows para a máquina virtual Ubuntu. Foi utilizado o comando **scp** para realizar a transferência do LattesService para o diretório **home** do usuário logado no Ubuntu. Em seguida, os arquivos foram movidos para o diretório **/var/www/**. Portanto, nas instruções a seguir será utilizado o diretório **/var/www/**.

Em seguida, proceda com as instalações e configurações.

## Instalação das dependências das bibliotecas *PyGraphviz* e *PyTidyLib*

As bibliotecas *PyGraphviz* e *PyTidyLib* são utilizadas pelo *scriptLattes*. Estão listadas no arquivo requirements.txt do *LattesService* e serão instaladas posteriormente. No entanto, é necessário que alguns pacotes sejam instalados antes: *python-dev*, *graphviz*, *libgraphviz-dev*, *pkg-config e libtidy-dev.* Para instalar esses pacotes execute o comando a seguir:

$ sudo apt-get install python-dev graphviz libgraphviz-dev pkg-config libtidy-dev

## Instalação do *PIP*

O *PIP* é um gerenciador de pacotes *Python*. É utilizado para fazer a instalação de bibliotecas *Python* e suas dependências. Portanto, será necessário para que se possa fazer a instalação de todas as bibliotecas *Python* utilizadas pelo *scriptLattes* e *LattesService*. Para instalá-lo execute o comando a seguir:

$ sudo apt-get install python-pip

Em seguida, faça o upgrade do PIP executando o comando a seguir:

$ pip install --upgrade pip

## Instalação do *Virtualenv*

O *Virtualenv* permitirá criar um ambiente que irá isolar as bibliotecas *Python* utilizadas pelo *LattesService*. Desta forma, o *LattesService* pode ser instalado em um servidor que já esteja rodando ou que venha a rodar outras aplicações *Python*, pois não haverá conflito de versões das bibliotecas *Python* utilizadas pelo *LattesService* e as demais aplicações. Para instalar o Virtualenv execute o comando a seguir:

$ pip install virtualenv

Após a instalação, acesse o diretório raiz do *LattesService* e crie o *Virtualenv:*

:/var/www/LattesService$ virtualenv venv

Nesse exemplo, o *Virtualenv* foi criado com o nome de **venv**.

## Instalação das bibliotecas *Python* utilizadas pelo *scriptLattes* e *LattesService*

Antes de iniciar a instalação, ative o *Virtualenv* criado anteriormente. Para isso, execute o comando a seguir a partir do diretório raiz do *LattesService*:

:/var/www/LattesService$ source venv/bin/activate

Após ativação, o nome do *Virtualenv* passa a aparecer entre parênteses no início da linha de comando do terminal. Com o *Virtualenv* ativado, execute o comando a seguir para instalar todas as bibliotecas *Python* utilizadas pelo *scriptLattes* e *LattesService*:

(venv) usuario@host:/var/www/LattesService$ pip install -r requirements.txt

O *PIP* irá instalar, dentro de **venv**, todas as bibliotecas *Python* listadas no arquivo requirements.txt, presente na raiz do *LattesService*, de acordo com a versão já especificada para cada uma delas.

## Configuração do *scriptLattes*

O *scriptLattes* possui um arquivo que permite configurar alguns parâmetros que irão guiar sua execução. Para fins de utilização com o *LattesService*, a configuração de alguns parâmetros pode ser indispensável, opcional, irrelevante ou não recomendada.

O parâmetro indispensável é **global-arquivo\_de\_entrada**. Deve especificar o caminho absoluto para acesso a lista de orientadores que terão o Currículo Lattes baixado e processado pelo *scriptLattes* para extrair a produção intelectual.

Exemplo: /var/www/LattesService/scriptLattes/scriptLattes.list

Acerca da configuração dos demais parâmetros, consulte o arquivo de configuração para maiores informações.

O arquivo de configuração encontra-se em scriptLattes/scriptLattes.conf. Você pode alterar a localização e o nome do arquivo, mas será necessário indicar essas alterações posteriormente no arquivo de configurações do *LattesService*.

## Configuração do *LattesService*

As aplicações que utilizam o *Django* possuem um arquivo onde são definidas uma série de configurações para personalizar e assegurar seu funcionamento. No *LattesService* este arquivo está localizado em lattes\_service/settings.py. Nele foram acrescentadas também algumas configurações para possibilitar a execução do *scriptLattes*. A seguir são apresentadas as configurações que requerem atenção.

### Segurança na execução do *LattesService*

* SECRET\_KEY: essa configuração não pode ficar em branco porque impedirá a execução do *Django*. A configuração utilizada em ambiente de produção não deve ser compartilhada com o ambiente de desenvolvimento. Para gerar uma nova chave utilize o seguinte comando com o Virtualenv ativado:

$ python -c ‘from django.core.management.utils import get\_random\_secret\_key; print(get\_random\_secret\_key())’

* DEBUG: a essa configuração deverá ser atribuído o valor *False*. Somente em ambiente de desenvolvimento poderá ser ativada (atribuindo o valor *True*), pois exibe uma séria de informações sensíveis sobre a aplicação.
* ALLOWED\_HOSTS: informe o endereço *IP* público e o nome do *host* e domínio do servidor que está hospedando o *LattesService*.

Exemplo:

ALLOWED\_HOSTS = [‘186.202.161.38’, ‘oftalmounifesp.com.br’]

### Bancos de dados

Há dois bancos de dados utilizados pelo *LattesService*: ***default*** e **institucional**. Configure, no mínimo, o banco de dados *default*, pois será utilizado para armazenar os dados extraídos dos Currículos Lattes. Já o banco de dados institucional, quando configurado, permitirá o acesso à lista de orientadores da sua instituição. Veja a seguir um exemplo de como configurar os bancos de dados:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'lattes\_service',

'USER': 'NOME\_DO\_USUARIO\_NO\_BANCO\_DE\_DADOS',

'PASSWORD': 'SENHA\_DO\_USUARIO',

'HOST': 'LOCALHOST',

'PORT': '5432',

},

'institucional': {

'ENGINE': 'django.db.backends.oracle',

'NAME': 'NOME\_DO\_BANCO\_DADOS',

'USER': 'NOME\_DO\_USUARIO\_NO\_BANCO\_DE\_DADOS',

'PASSWORD': 'SENHA\_DO\_USUARIO',

'HOST': 'ENDERECO\_DO\_BANCO\_DE\_DADOS',

'PORT': 'NUMERO\_DA\_PORTA\_ONDE\_O\_SERVICO\_ESTA\_RODANDO',

}

}

O parâmetro *ENGINE* determina o SGBD (Sistema de Gerenciamento de Banco de Dados) que será usado. O banco de dados **institucional** deverá ser configurado conforme o SGBD utilizado em sua instituição. Além disso, deverão ser instaladas as bibliotecas *Python* pertinentes. Para o banco de dados ***default*** a biblioteca necessária já está inclusa no arquivo requirements.txt.

Para maiores informações, consulte a documentação do *Django* em <https://docs.djangoproject.com/en/1.11/ref/databases/>

### Regionalização

O LattesService utiliza por padrão o *time zone* America/Sao\_Paulo para lidar com datas e horas. Caso sua instituição esteja em uma região onde o horário padrão não é o mesmo de São Paulo, consulte esta lista de *time zones* <https://en.wikipedia.org/wiki/List_of_tz_database_time_zones> e atualize a configuração conforme o exemplo a seguir:

TIME\_ZONE: ‘America/Sao\_Paulo’

### Cache de dados no cliente

A fim de otimizar o uso do servidor que hospedará o *LattesService* e diminuir o tráfego de rede, há duas configurações que irão impor à aplicação cliente a manutenção de um cache (cópia) dos dados. Quando for detectado que o cliente já solicitou os dados e ainda não houve uma nova extração dos Currículos Lattes, o *LattesService* irá retornar um arquivo sem dados e com o *status* *HTTP* 204 (sem conteúdo). Veja a seguir as duas configurações:

* FORCAR\_CACHE\_NO\_CLIENTE: quando definida como TRUE irá obrigar o cliente a fazer o cache, mas caso seja definida como FALSE irá atender a todas as solicitações do cliente.
* PRAZO\_DO\_CACHE: este prazo (definido em número de dias) possibilitará ao cliente evitar novas solicitações ao LattesService e fazer uso do cache até que expire.

### Execução do *scriptLattes*

Estas configurações permitem personalizar a execução do *scritpLattes* pelo *LattesService*:

* COMPILAR\_PRODUCAO: quando definida com *True* irá aumentar consideravelmente o tempo de processamento, mas removerá as duplicidades encontradas entre as produções intelectuais. É útil quando se deseja exibir toda a produção de um PPG ou da instituição, embora não tenha sido implementada uma funcionalidade que faça essa exibição, pois não era o objetivo do projeto que concebeu o *LattesService*. Para fins de entrega dos dados da produção intelectual do orientador, basta manter essa configuração com *False*;
* ARQUIVO\_DE\_CONFIGURACAO\_DIR: deve indicar o local onde estará armazenado o arquivo de configuração do *scriptLattes.* O padrão é: os.path.join(BASE\_DIR, ‘scriptLattes’). Caso queira mudar, recomenda-se mudar apenas o diretório ‘scriptLattes’, mantendo o novo diretório dentro do diretório base (BASE\_DIR) do *LattesService*;
* ARQUIVO\_DE\_CONFIGURACAO\_SCRIPT\_LATTES: deve indicar o nome e extensão do arquivo de configuração do *scriptLattes*. O padrão é: os.path.join(ARQUIVO\_DE\_CONFIGURACAO\_DIR, ‘scriptLattes.config’). Observe que esta configuração depende da anterior, portanto, certifique-se de que o arquivo de configuração seja armazenado dentro do diretório definido por ARQUIVO\_DE\_CONFIGURACAO\_DIR. Caso queira, é possível mudar o nome do arquivo.
* LISTA\_ORIENTADORES\_SCRIPT\_LATTES: deve indicar o nome e extensão do arquivo de entrada do *scripLattes*, ou seja, o arquivo que contém a lista de orientadores. O LattesService utilizará essa configuração para criar o arquivo de entrada. O padrão é: os.path.join(ARQUIVO\_DE\_CONFIGURACAO\_DIR, ‘scriptLattes.list’). Observe que, assim como o arquivo de configuração, o arquivo de entrada depende da configuração ARQUIVO\_DE\_CONFIGURACAO\_DIR. Caso queira, é possível mudar o nome do arquivo.

### Automatização do *LattesService*

A automatização do LattesService é realizada por meio do “django\_crontab”, um aplicativo que permite configurar tarefas a serem executadas de forma recorrente pelo cron do Ubuntu.

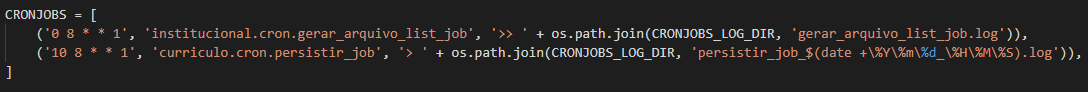
Cada tarefa do “django\_crontab” é chamada de job e deve ser adicionada ao dicionário CRONJOBS.

Um job do LattesService possui três parâmetros:

* intervalo de tempo no formato do cron (<https://help.ubuntu.com/community/CronHowto>);
* caminho, no formato Python, para o método ou função que será executada;
* símbolo de redirecionamento de saída (>> ou >) e caminho para o arquivo de log que armazenará os dados da execução.

Apenas o primeiro parâmetro precisa ser configurado para ajustar a execução conforme a necessidade da IES.

A seguir, é demonstrado um exemplo de configuração:



O primeiro *job* executará a função gerar\_arquivo\_list\_job(), que está no módulo institucional.cron, todas as segundas-feiras (1) às 8h00 da manhã e acrescentará (>>) os dados da execução no arquivo **logs/cron/gerar\_arquivo\_list\_job.log**.

Em seguida, o segundo job executará a função persistir\_job(), que está no módulo curriculo.cron, todas as segundas-feiras (1) às 8h10 da manhã e armazenará (>) dos dados da execução no arquivo **logs/cron/persistir\_job\_20201217\_081001.log** (o nome do arquivo conterá a data e a hora que o job começou a ser executado, neste exemplo o início foi em 17/12/2020 às 08h10m01s).

Após configurar os jobs, salve o arquivo lattes\_service/settings.py e execute o comando a seguir no terminal, com o Virtualenv ativado:

$ python manage.py crontab add

## Criação das tabelas no PostgresSQL

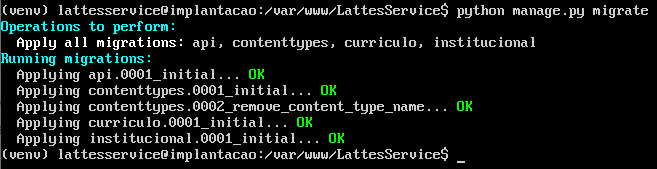
Após realizar as configurações, o próximo passo é criar as tabelas no banco de dados do LattesService.

O Django possui um utilitário de linha de comando que cria todas as tabelas no banco de dados. O utilitário busca as definições de dados em arquivos que ficam armazenados em diretórios chamados *migrations*. Todos os apps no Django possuem, por padrão, um diretório *migrations*.

Para criar as tabelas no banco de dados execute o seguinte comando no terminal, com o Virtualenv ativado:

$ python manage.py migrate

O progresso será exibido à medida que as tabelas forem criadas, conforme a figura a seguir.



## Geração dos arquivos estáticos

Para que o servidor HTTP possa encontrar e disponibilizar os conteúdos estáticos (HTML, CSS, imagens, ícones, javascripts) do Django, execute o seguinte comando no terminal, com o Virtualenv ativado:

$ python manage.py collectstatic

## Verificação do funcionamento do Django

Para verificar se o Django está funcionando corretamente, execute o comando a seguir:

$ python manage.py check

Se estiver funcionando corretamente será exibida a seguinte mensagem:

System check identified no issues (0 silenced).

## Modificação das permissões dos arquivos

O servidor Apache só conseguirá acessar os arquivos do LattesService se eles estiverem com as permissões adequadas. Altere as permissões de arquivo com seguintes comandos:

:/var/www$ sudo chown -R :www-data LattesService/

:/var/www$ sudo chmod -R 755 LattesService/

:/var/www$ sudo chmod 765 LattesService/logs/lattes\_service.log

## Exemplo de configuração do servidor HTTP

A seguir será mostrado um exemplo básico de configuração do servidor HTTP para servir as páginas do LattesService a partir de um virtual host.

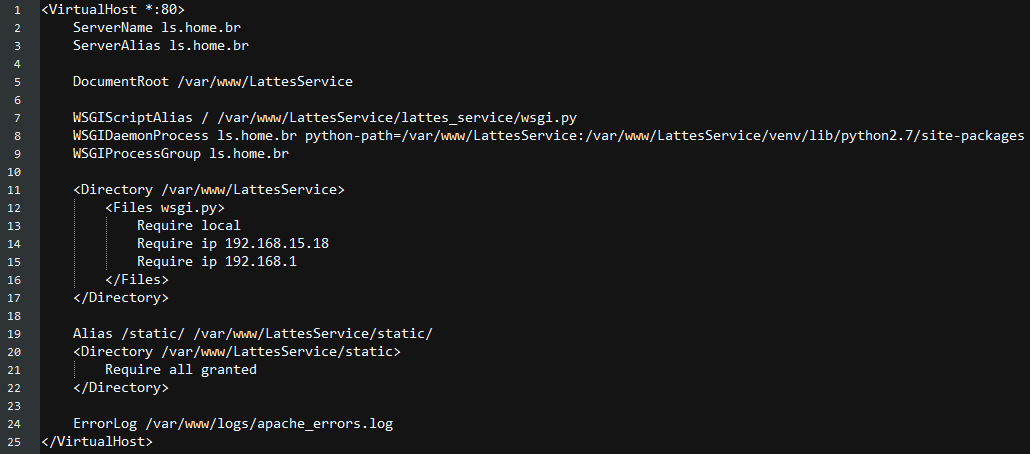
Execute o seguinte comando:

$ sudo nano /etc/apache2/sites-available/ls.conf

Explicando as partes do comando:

* nano – é editor de texto que será usado para escrever as configurações;
* /etc/apache2/sites-available/ - é o diretório onde são armazenados os arquivos de configuração de *virtual host*.
* ls.conf – é o nome do arquivo que será criado para guardar as configurações do virtual host. Neste exemplo, foi dado o nome do subdomínio ao arquivo, **ls**. A extensão do arquivo deve ser **.conf**.

O arquivo ls.conf terá as seguintes configurações



Explicando as partes da configuração:

* linha 1 - Este virtual host receberá todas as requisições que chegarem na porta 80;
* linhas 2 e 3 – nome e alias atribuídos ao servidor;
* linha 5 - diretório onde está hospedado o LattesService;
* linha 7 – indica a raiz do diretório como um alias para o arquivo wsgi.py do LattesService;
* linha 8:
  + ls.home.br – nome que permitirá identificar o processo;
  + python-path – diretório onde está o LattesService e diretório onde o Virtualenv armazena as bibliotecas Python do LattesService;
* linha 9: nome do grupo do processo;
* linhas 11 a 17 – por meio desta configuração é possível restringir o acesso ao LattesService, pois nela são definidas as permissões de execução do arquivo wsgi.py quando:
  + linha 13 –a requisição partir do computador onde o LattesService está hospedado;
  + linha 14 – a requisição partir de um dispositivo com o IP 192.168.15.18;
  + linha 15 – a requisição partir de um dispositivo da rede 192.168.1;
* linha 19 – define um alias para o diretório static do LattesService;
* linhas 20 a 22 – define as permissões de acesso ao diretório static;
* linha 24 – indica o caminho para salvar o log de erros do servidor Apache. É necessário criar o diretório.

Crie o diretório logs/ conforme informado na linha 24 e altere as permissões de acesso.

:/var/www$ sudo mkdir logs/

:/var/www$ sudo chown www-data:www-data logs/

:/var/www$ sudo chmod 755 logs/

Após configurar o virtual host e salvar o arquivo, ative-o executando o seguinte comando:

$ sudo a2ensite ls.conf

Não é necessário informar o caminho para o arquivo de configuração. Informe apenas o nome e a extensão.

Caso não haja erros na configuração será exibida a mensagem:

Enabling site ls.

To activate the new configuration, you need to run:

service apache2 reload

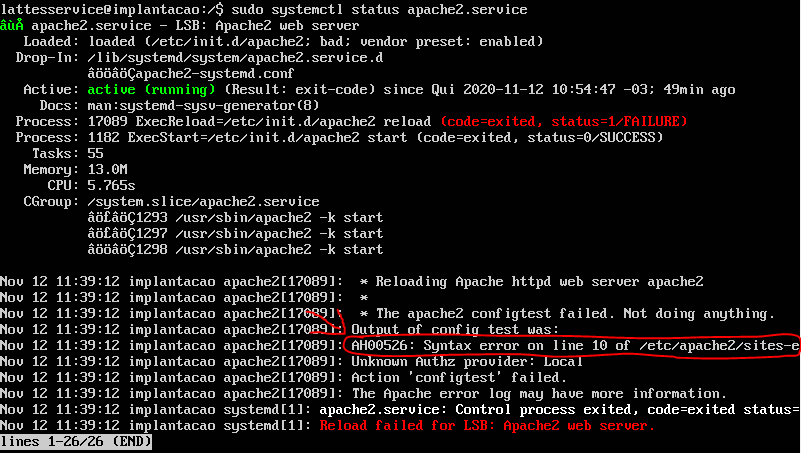
Reinicie o servidor Apache com o comando:

$ sudo service apache2 reload

Caso seja apresentada uma mensagem de erro, execute o comando a seguir:

$ sudo systemctl status apache2.service

Serão exibidas as informações sobre o erro. No exemplo a seguir, o erro foi detectado na linha 10 do arquivo ls.conf.



Corrija o erro e reinicie o Apache, conforme orientado anteriormente.

# ATUALIZAÇÃO DA VERSÃO DO *scriptLattes*

Algumas alterações foram realizadas no scriptLattes para que o LattesService pudesse executá-lo corretamente e com bom desempenho. Por isso, ao realizar a atualização do scriptLattes modifique os seguintes trechos de código:

* Arquivo scriptLattes/util.py

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* Arquivo scriptLattes/grupo.py

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

# UTILIZAÇÃO DO CLIENTE HTTP

Junto com o LattesService disponibilizamos um modelo de cliente de *web service* desenvolvido em PHP.

O download pode ser realizado em <https://github.com/rivaldo-so-junior/LattesServiceClient>

O cliente possui quatro classes que auxiliarão na obtenção e configuração dos dados para exibição no *website* do PPG:

* classe Cache - cria e utiliza caches;
* classe LattesServiceAPI – realiza a requisição de dados ao LattesService;
* classe ProducaoCientifica – organiza os dados de produção científica obtidos com a classe LattesServiceAPI;
* classe GeradorDeHTML – monta o arquivo HTML que será exibido pelo website do PPG e que também é guardado em cache.

A seguir, é mostrado um exemplo de utilização das classes para implementação no website do PPG.

