
agesic
agencia de gobierno electrónico
y sociedad de la información

1 Índice

2 Versiones

Fecha	Versión	Autor	Descripción
26/01/16	1	Pablo Di Leva Siage	Versión inicial del documento.
05/02/16	1.1	Pablo Di Leva Siage	Actualización para Centos 7.
18/02/16	1.2	Pablo Di Leva Siage	Agregados capít. Sphinx y Tomcat.
26/02/16	1.3	Pablo Di Leva Siage	Actualizaciones en general.
11/03/16	1.4	Pablo Di Leva Siage	Actualizaciones (SimpleSAML, PHP).
01/05/16	1.5	Pablo Di Leva Siage	Actualizaciones. Se agrega anexo.
10/05/16	1.6	Pablo Di Leva Siage	Actualizaciones instalación de PHP.
19/05/16	1.7	Pablo Di Leva Siage	Agregado capítulo de PDI.
08/06/16	1.8	Pablo Di Leva Siage	Actualizaciones y capítulo de integraciones.
09/06/16	1.9	Pablo Di Leva Siage	Cambio de orden migraciones e imp de datos.
20/09/16	1.10	Pablo Di Leva Siage	Se agrega información acerca de cron para conciliación de pagos y se agregan constantes a configurar de CDA, Pasarela de pagos y PDI.
01/11/16	1.11	Pablo Di Leva Siage	Se agrega comando Cron de reencolamiento de trazas, nueva librería Redis de PHP y recomendación de link simbólico para directorio uploads en caso de utilizar balanceador.
11/11/16	1.12	Pablo Di Leva Siage	Se agrega información para configuración de autenticación LDAP.

3 Procedimiento de instalación

Este documento provee los pasos necesarios para realizar la instalación de la aplicación AGESIC SIMPLE BPM.

3.1 Sobre la aplicación

La aplicación se ejecuta sobre un servidor web Apache versión 2.4.x y almacena sus datos en una base de datos relacional MySQL versión 5.x. Es necesario instalar también el interprete de lenguaje PHP en su versión 5.5.x. Para este manual utilizaremos el sistema operativo Gnu/Linux Centos 7.

3.2 Instalación de Apache

Para la instalación de Apache debemos tomar en cuenta en que sistema operativo estamos trabajando.

Centos 7

1. Ejecutamos en la terminal:

   $ yum install httpd

   $ systemctl start httpd.service

   $ systemctl enable httpd.service

   $ vi /etc/httpd/conf/httpd.conf

   Cambiamos dentro de **"/var/www/html"** el valor de **"AllowOverride None"** por **"AllowOverride All"**.

   Agregamos al final del documento:

   $ LoadModule rewrite_module modules/mod_rewrite.so

   Agregamos la linea:

   TraceEnable off

1. Habilitamos las conexiones externas a Apache:

   $ setsebool httpd_can_network_connect=1

   $ setsebool httpd_can_network_connect_db=1

   $ firewall-cmd --permanent -—add-port=80/tcp

   $ firewall-cmd --permanent --add-port=443/tcp

   $ systemctl restart httpd.service

2. Ubicamos la aplicación en /var/www/html o en donde se requiera.

NOTA 1: Asegurarse de que la raíz de la aplicación contiene el archivo **.htaccess** para su correcto funcionamiento.

3.3 Instalación de balanceador de carga

A continuación instalaremos un balanceador de carga en Apache utilizando **mod_proxy_balancer**.

Centos 7

3. Editamos el archivo de configuración de Apache ubicado en **/etc/httpd/conf/httpd.conf** y cargamos los siguientes módulos:

   LoadModule proxy_balancer_module modules/mod_proxy_balancer.so

   LoadModule proxy_module modules/mod_proxy.so

   LoadModule proxy_http_module modules/mod_proxy_http.so

   LoadModule status_module modules/mod_status.so

1. Agregamos las siguientes lineas a los VirtualHosts de cada servidor que queramos utilizar:

   RewriteEngine On

```
RewriteRule .* -
[CO=BALANCEID:balancer.http{numero_servidor}:.
{dominio.com.uy}]
```

1. Reiniciamos el servidor Apache.
2. Creamos el VirtualHost del balanceador de carga:

```
<VirtualHost *:80>
  ServerName {dominio.com.uy}
  DocumentRoot {ruta_aplicación}
  ProxyRequests Off
  <Proxy *>
    Allow from all
  </Proxy>
  ProxyPass /balancer-manager !
  ProxyPass / balancer://mycluster/
stickysession=BALANCEID nofailover=On
  ProxyPassReverse / http://{numero_servidor}.
{dominio.com.uy}/
  <Proxy balancer://mycluster>
    BalancerMember http://http{numero_servidor}.
{dominio.com.uy} route=http1
    ProxySet lbmethod=byrequests
  </Proxy>
  <Location /balancer-manager>
    SetHandler balancer-manager
    Allow from all
  </Location>
</VirtualHost>
```

3. Reniciamos el servidor Apache.

3.4 Instalación de MySQL

Al igual que sucede con la instalación de Apache, para la instalación del motor de base de datos MySQL debemos tomar en cuenta nuestro sistema operativo.

Centos 7

1. Ejecutamos en la terminal:

```
$ yum install mariadb-server mariadb mysql-devel
$ systemctl start mariadb.service
$ systemctl enable mariadb.service
$ mysql_secure_installation
```

Configuramos lo siguiente:

1. Ingresamos nueva contraseña de root para MySQL.

2. Removemos usuarios anónimos.
3. Actualizamos los nuevos privilegios.

NOTA: Quitar el modo "**STRICT_TRANS_TABLES**" en el archivo **/etc/my.cnf** en caso de que se encuentre habilitado, ya que puede generar inconsistencias en la aplicación.

3.5 Crear la base de datos para la aplicación

Luego de haber instalado el motor de base de datos MySQL será necesario crear la base de datos para la aplicación.

Centos 7

1. Ejecutamos en la terminal:

```
$ mysql -u root -p
   mysql> CREATE DATABASE simple_bpm;
   mysql> CREATE USER 'nombre_usuario'@'localhost'
   IDENTIFIED BY 'nueva_contrasena';
   mysql> GRANT ALL PRIVILEGES ON simple_bpm.* TO
   'nombre_usuario'@'localhost';
   mysql> FLUSH PRIVILEGES;
   mysql> USE simple_bpm;
   mysql> SOURCE <ruta_a_aplicación>/sql/estructura.sql
$ vi
   <ruta_a_aplicación>/application/config/database.php
```

   Editamos los datos de conexión con MySQL.

2. Luego desde la terminal, en la raíz de la aplicación realizamos las migraciones e importamos los datos basicos necesarios:

```
$ php index.php migration migrate
$ mysql -u root -p
mysql> use simple_bpm;
mysql> SOURCE <ruta_a_aplicación>/sql/datos.sql
mysql> SOURCE <ruta_a_aplicación>/sql/bloques.sql
```

3.6 Replicación de la base de datos

El sistema de respaldo de MySQL permite configurar un ambiente de replicación, en la cual existen dos servidores MySQL, uno de los cuales se llama "**Maestro**" y es el utilizado por la aplicación, y otro "**Esclavo**", que es una copia idéntica del primero. Para configurar dicho entorno debemos hacer lo siguiente:

Servidor Maestro

1. Ingresamos al cliente de MySQL via consola y ejecutamos:

```
mysql> CREATE USER {usuario_replicacion}@{IP_esclavo}
   IDENTIFIED BY '{contraseña}';
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO
   {usuario_replicacion}@{IP_esclavo};
```

1. Editamos el archivo de configuración de MySQL ubicado en **/etc/my.cnf** agregando:

```
server-id=1

log-bin

binlog-do-db={base_de_datos}
```

1. Reiniciamos el servidor de MySQL.
2. Desde la terminal generamos un respaldo completo de la base de datos:

```
$ mysqldump --master-data=2 -uroot -p --databases
   {base_de_datos} > dump.sql
```

Servidor Esclavo

1. Editamos el archivo de configuración de MySQL ubicado en **/etc/my.cnf** agregando:

```
server-id=2

replicate-do-db={base_de_datos}
```

1. Reiniciamos el servidor de MySQL.
2. Desde la terminal importamos el respaldo generado anteriormente desde el mater:

```
$ mysql -u root -p < dump.sql
```

3. Ingresamo al cliente de MySQL via terminal y cambiamos las coordenadas de replicación:

```
mysql> CHANGE MASTER TO MASTER_HOST={IP_master},
   MASTER_USER={usuario_replicacion},
   MASTER_PASSWORD={contraseña},
   MASTER_LOG_FILE={file}, MASTER_LOG_POS={pos}";
```

Los valores de **MASTER_LOG_FILE** y **MASTER_LOG_POS** se obtienen a partir del archivo del respaldo, buscando la sentencia **CHANGE MASTER TO**.

1. Iniciamos la replicación en el servidor esclavo:

```
mysql> START SLAVE
```

2. Comprobamos el funcionamiento de la replicación ejecutando:

```
mysql> SHOW PROCESSLIST
```

3.7 Respaldo y Recuperación

En esta sección se describe el procedimiento de respaldo y recuperación de la aplicación.

Respaldo

Existen dos formas de hacer un respaldo: total o parcial. A continuación se describen ambas.

Respaldo total

El respaldo total es el recomendado ya que es más seguro y hace la recuperación mucho más sencilla, aunque puede tomar más tiempo hacerlo y más espacio almacenarla. Consiste en lo siguiente;

- Hacer un respaldo de la base de datos completa (debe incluir todas las bases de datos existentes) para lo cual debe usarse la herramienta que se considere más apropiada para hacer respaldos de MySQL.
- Hacer un respaldo del servidor Apache completo, para lo cual debe hacerse un archivo comprimido (.zip, .tar.gz, .bzip o el formato que se prefiera) de la carpeta que contiene la aplicación (en CentOS sería **`/var/www/html`**). Se recomienda eliminar previamente el contenido de la carpeta logs (**`application/logs`**) ya que no serán útiles en caso de realizar una restauración. Nota importante: antes de hacer el respaldo del servidor es recomendable detenerlo por completo con el fin de asegurarse de que ningún archivo quede almacenado en forma parcial.

Respaldo parcial

El respaldo parcial no se recomienda si es viable hacer un respaldo total, aunque puede ser útil en los casos en los cuales se tiene más de una instalación similar con pequeñas variantes. Las cosas que deben respaldarse en este caso son las siguientes:

- La base de datos. Debe incluir todos las bases de datos existentes para lo cual debe usarse la herramienta que se considere más apropiada para hacer respaldos de MySQL.
- Directorio uploads completo ubicado en la raíz de la aplicación.
- Se recomienda también guardar una copia de los archivos que se mencionan a continuación; estos archivos no serán usados por sí mismos, pero pueden ser utilizados para abreviar el tiempo de recuperación en caso de un incidente (a los efectos de copiar los valores originales):
  - **`application/config/config.php`**
  - **`application/config/constants.php`**
  - **`application/config/database.pop`**
  - **`application/config/email.php`**
  - **`application/third_party/simplesaml/cert/*`**
  - **`application/third_party/simplesaml/authsources.php`**
  - **`application/third_party/simplesaml/config.php`**
  - **`application/third_party/simplesaml/metadata/saml20-idp-remote.php`**

Recuperación

La recuperación de la aplicación en caso de un incidente depende del mecanismo utilizado para hacer el respaldo (total o parcial), aunque en ambos casos debe comenzarse por restaurar el respaldo de la base de datos utilizano la herramienta que se considere más apropiada. A continuación se describe el procedimiento en cada caso.

Recuperación de un respaldo total

En el caso de que se desee recuperar un respaldo total debe realizarse lo siguiente:

1. Restaurar la base de datos utilizando el respaldo hecho anteriormente, para lo cual debe utilizarse la herramienta que se considere más apropiada.
2. Restaurar el servidor de aplicaciones respaldado anteriormente, para lo cual basta con descromprimir el zip hecho anteriormente en su ubicación original (en caso de utilizar otra ubicación diferente tal vez sea necesario modificar las rutas que se describen a lo largo del documento para reflejar este cambio).

Recuperación de un respaldo parcial

En el caso de que se desee recuperar un respaldo parcial debe realizarse lo siguiente:

1. Restaurar la base de datos utilizando el respaldo hecho anteriormente, para lo cual debe utilizarse la herramienta que se considere más apropiada.
2. Realizar el procedimiento de instalación, tal si fuera una instalación nueva, solo que donde se indique crear archivos nuevos (como certificados) o configuraciones deben utilizarse los archivos respaldados anteriormente.

3.8 Instalación de PHP

Luego de instalar el servidor web y el motor de base de datos procedemos con la instalación de PHP.

Centos 7

1. Ejecutamos en la terminal:

```
$ rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
$ rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
$ yum install php55w php55w-opcache php55w-mysql php55w-mcrypt php55w-xml php55w-mbstring php55w-openssl php55w-gd php55w-pecl-redis
$ vi /etc/php.ini
```

Cambiamos el valor de **“short_open_tag”** por **“On”.**

Descomentamos **“date.timezone”** y le agregamos el valor correspondiente a nuestra zona horaria.

Aumentamos los valores de los siguientes flags:

```
post_max_size = 40M
upload_max_filesize = 40M
$ systemctl restart httpd.service
```

Verificamos que la versión instalada es la correcta:

```
$ php -v
```

3.9 Instalación de Redis

A continuación se deberá instalar el servidor Redis.

Centos 7

1. Ejecutamos en la terminal:

```
$ yum install redis
$ systemctl start redis.service
$ systemctl enable redis.service
```

2. Verificamos que se encuentra activo y funcionando correctamente:

```
$ redis-cli ping
```

3.10 Instalación de Resque

A continuación se deberá instalar Resque para procesar las solicitudes de trazabilidad de la aplicación.

Centos 7

1. Editamos el archivo de configuración que se encuentra en **vendor/fresque/fresque.ini** y cambiamos el valor de "**user**" por el usuario de apache.
2. Luego ejecutamos en la terminal:

```
$ cd vendor/fresque
$ ./fresque start -q default
```

Es importante aclarar que luego de realizar cualquier cambio en la aplicación (configuración, codigo, etc.) debe reiniciarse Apache, Redis y Resque en ese orden.

3.11 Instalación de Conector PGE

A continuación se deberá instalar el conector para habilitar las consultas a webservices externos.

Para esta tarea debe consultarse el manual de instalación del Conector PGE.

3.12 Instalación de Apache Tomcat

A continuación se deberá instalar Apache Tomcat para ser utilizado por AgesicFirma posteriormente.

Centos 7

1. Ejecutamos en la terminal:

```
$ yum install tomcat
```

3.13 Instalación de XMLSIGNER PDI

A continuación se deberá configurar el componente para la funcionalidad de PDI.

1. Editamos el archivo de configuración "**config.properties**" ubicado en vendors/xmlsigner.jar. La variable "**keystore_directory**" debe tener como valor la ruta absoluta al directorio "**uploads/pdi**".
2. Editamos el archivo de configuracion ubicado en `application/config/constants.php` en donde debemos ingresar los siguientes datos:

- **JAR_FIRMA** – Ubicación absoluta del ejecutable **xmlsigner.jar** que se encuentra en vendors
- **UBICACION_CERTIFICADOS_PDI** – Mismo valor que el punto 1.

3.14 Instalación de AgesicFirma

A continuación se deberá instalar el componente AgesicFirma para la funcionalidad de firma de documentos de simple.

Centos 7

1. Movemos el directorio "**agesic_firma**" que se encuentra en **vendors** y colocarlo en la raíz de apache (al mismo nivel que Simple).
2. Movemos el componente **AgesicFirmaWS.war** en el directorio de tomcat.
3. Se deberá actualizar las rutas definidas en el archivo de configuración "**constants.php**" ubicado en config de Simple. Las constantes a actualizar son:

   WS_FIRMA_DOCUMENTOS (accedido por Simple desde servidor).

   WS_AGESIC_FIRMA (accedido por java desde cliente).

   WS_AGESIC_FIRMA_OK

   WS_AGESIC_FIRMA_CODEBASE (ubicación del directorio *agesic_firma* distribuido en *vendors*).
4. Se deberá crear un esquema en MySQL con el nombre "agesic_firma".
5. Dentro del mismo se deberá crear la siguiente tabla (en mayúsculas):

```
CREATE TABLE agesic_firma.DOCUMENTOS (
  id  int AUTO_INCREMENT,
  archivo longblob,
  certificate longblob,
  fecha_modif date,
  firma_valida boolean,
  firmado boolean,
  id_transaction character varying(255),
  tipo_firma character varying(255),
  CONSTRAINT documentos_pkey PRIMARY KEY (id))
```

6. Dentro del archivo **AgesicFirmaWS.war** en la ruta **META-INF/context.xml** se deberán cambiar los datos de acceso a la base de datos MySQL.
7. Dentro del archivo **AgesicFirmaWS.war** en la ruta **WEB-INF/classes/META-INF/persistence.xml** se deberán cambiar los datos de acceso a MySQL.

3.15 Instalación de Sphinx

A continuación se instalará Sphinx para el indexado de la base de datos MySQL.

Centos 7

1. Ejecutamos en la terminal:

   ```
   $ yum install sphinx
   ```

2. Se deberá reemplazar el archivo **sphinx.conf** alojado en **/etc/sphinx/sphinx.conf** por el que se encuentra en Simple dentro de **spinx/sphinx.conf.sample** y cambiar los datos de acceso a MySQL.

3. Se deberá mover el directorio **index** ubicado en Simple dentro de sphinx hacia **/etc/sphinx**.

4. Luego ejecutamos en la terminal:

   ```
   $ indexer --rotate --all

   $ searchd
   ```

5. Es necesario ejecutar estos comando de forma periódica para mantener los índices actualizados. Para esto se deberá crear un Cron de la siguiente forma:

   ```
   $ crontab -l

   Agregar las siguientes líneas:

   0 * * * * php /ruta/a/simple/index.php cron hourly

   0 0 * * * php /ruta/a/simple/index.php cron daily
   ```

3.16 Configuración de SimpleSAML

A continuación se procederá a configurar SimpleSAML para la autenticación con el servicio Coesys.

1. El primer paso será guardar la clave privada y el certificado requeridos por SimpleSAML (ambos con extensión .PEM) en el directorio ubicado en **application/third_party/simplesaml/cert**

2. Luego editamos el archivo ubicado en **application/third_party/simplesaml/config/authsources.php** cambiando el nombre de la clave privada y certificado nuevos que se encuentran dentro de la estructura titulada "**simplesaml**". Luego cambiamos el resto de los valores de acuerdo a los datos que tenemos de Coesys.

3. Una vez completado el paso anterior procedemos a editar el archivo ubicado en **application/third_party/simplesaml/metadata/saml20-idp-remote.php** en donde debemos modificar la estructura del IDP con los valores nuevos en los elementos "**Location**". Importante: el nombre de la estructura metadata debe ser igual al valor ingresado en el campo "**idp**" del archivo **authsources.php**.

4. Al finalizar editaremos el archivo de configuración **config.php** ubicado en **application/third_party/simplesaml/config**. Se deberán actualizar los siguientes parámetros de configuración con los valores correspondientes:

tempdir (directorio con permisos de escritura para archivos temporales)

timezone (zona horaria del servidor)

session.cookie.domain (dominio de la aplicación)

session.phpsession.savepath (directorio con permisos de escritura para archivos temporales)

3.17 Configuración general

A continuación se procederá a configurar parámetros generales de la aplicación.

1. Accedemos al archivo de configuración **config.php** ubicado en **application/config**. Se deberán actualizar los siguientes parámetros de configuración:

   $config['base_url'] (incluido el protocolo. Completar SOLO si NO se utilizan cuentas, de lo contrario dejar vacio).

   $config['cookie_secure'] (setear solo si la aplicación tiene HTTPS configurado).

   $config['cookie_domain'] (dominio comenzando con un punto, ej: .dominio.com).

   $config['https'] (si la aplicación debe accederse via HTTPS).

   $config['sphinx_host'] **(**host de Sphinx).

   $config['sphinx_port'] (puerto de Sphinx).

   $config['main_domain'] (dominio principal si se requiere utilizar cuentas).

2. Accedemos al archivo de configuración **constants.php** ubicado en **application/config**. Se deberán actualizar los siguientes parámetros de configuración:

   DIRECTORIO_SUBIDA_DOCUMENTOS (actualmente uploads/documentos).

   IMPORTANTE: En caso de estar configurado Simple detras de un balanceador, se recomienda crear un link simbólico en lugar del directorio **uploads**, para que apunte a un directorio único compartido por todos los nodos, ya que se podrian generar inconsistencias con documentos y otros archivos al no encontrarse disponibles en todos los nodos.

   WS_FIRMA_DOCUMENTOS (webapp de firma en tomcat – se accede internamente desde el servidor de Simple. Ejemplo: **SIMPLE.gub.uy:8080/AgesicFirmaWS/AgesicFirmaServer**).

   WS_AGESIC_FIRMA (WSDL de firma – se accede externamente. Ejemplo: **SIMPLE.gub.uy:8080/AgesicFirmaWS/AgesicFirma?wsdl**).

   WS_AGESIC_FIRMA_OK (ruta a Simple, se accede externamente. Ejemplo: **SIMPLE.gub.uy:80/etapas/confirmar_firma**).

   WS_AGESIC_FIRMA_CODEBASE (directorio con los componentes de firma, se accede externamente. Ejemplo: **SIMPLE.gub.uy:80/componentes_firma**).

DOMINIO (dominio de la aplicación).

3.18 Configuración de integraciones

Simple requiere determinados datos de servicios externos para su funcionamiento. A continuación se listan dichos servicios y los procedimientos para su integración:

CDA

1. Para permitir autenticación con este sistema, es necesario solicitar el alta a CDA con el nuevo dominio de la aplicación. No es necesario dar de alta CDA para cada cuenta de la instalación de la aplicación (subdominios), solo es necesario tener configurado CDA para el dominio principal.
2. Tanto la clave privada como el certificado deben almacenarse en el directorio **application/third_party/simplesaml/cert** en formato PEM.
3. Se deben configurar los archivos authsources y config contenidos en **application/third_party/simplesaml/config** con los nuevos datos provistos por CDA.
4. Se debe configurar el archivo saml20-idp-remote.php contenido en **application/third_party/simplesaml/metadata** con los nuevos valores provistos por CDA.
5. Se debe configurar dentro de application/config/constants.php la constante **JAR_VALIDACION** que hace referencia a la ruta en donde se encuentra el ejecutable **xmlsignaturevalidator.jar**. Este ejecutable se encuentra ubicado en **vendors** en las distribuciones de Simple. Tambien se debe ingresar la ruta al certificado público de CDA, dicha ruta se debe agregar en la constante **CERTIFICADO_CDA_PUBLICO**.

LDAP

1. Se deberá configurar el archivo **application/config/constants.php** con los siguientes datos:

TIPO_DE_AUTENTICACION debe contener el tipo de autenticación a utilizar, en este caso “LDAP”.

LDAP_HOST debe contener el host del servidor LDAP.

LDAP_PUERTO debe contener el puerto al cual conectar en el servidor LDAP.

LDAP_BASE_DN debe contener el Base DN correspondiente.

LDAP_ATTR debe contener el CN en caso de que se requiera.

LDAP_USER debe contener el usuario a utilizar para autenticar al servidor LDAP.

LDAP_PASS debe contener la contraseña a utilizar para autenticar al servidor LDAP.

LDAP_VERSION debe contener la versión LDAP utilizada en el servidor LDAP en caso de que se requiera.

Pasarela de Pagos

1. Para esta funcionalidad se requiere el alta del dominio en el proveedor del sistema de pasarela de pagos, al que se le deben enviar:
    1. IP pública de la red en donde se encuentra ubicada la aplicación.
    2. Cinco URLs (con el dominio principal) de las cuales, las primeras 4 deben poder ser accedidas desde el exterior ya que el sistema de pagos envía a Simple información acerca de los pagos realizados por los usuarios.

        SIMPLE.gub.uy/pagos/ok

        SIMPLE.GUB.UY/pagos/error

        SIMPLE.gub.uy/pagos/pendiente

        SIMPLE.GUB.UY/pagos/rechazado

        SIMPLE.GUB.UY/pagos/control
    3. Luego se debe ingresar en application/config/constants.php los siguientes datos:

        WS_PASARELA_PAGO (URL del servicio web de la pasarela de pagos para solicitar token).

        WS_PASARELA_PAGO_CONSULTA (URL del servicio web de la pasarela de pagos para realizar consultas de pagos).

        POST_PASARELA_PAGO (URL del gateway de la pasarela de pagos).

        UBICACION_CERTIFICADOS_PASARELA (ubicación de los certificados para la conexión con los servicios web de pasarela de pagos, por defecto dicha ubicación es **uploads/pasarela**).
1. Se deberá configurar un Cron para la ejecución diaria de un script de actualización de estados de los pagos iniciados por los usuarios. Se recomienda configurar dicho Cron para ser ejecutado cada día a las 02:00 am. Ejemplo del Cron:

    0 2 * * * php {RUTA_A_RAIZ_DE_SIMPLE}/index.php tasks/pagos conciliacion

    NOTA: Se debe tomar en cuenta que para la ejecución del script es requerido que se encuentre la pasarela de pagos correctamente configurada y el servidor con conectividad a Internet.

Trazabilidad

1. Para la integración con Trazabilidad, Simple requiere el conector (Conector PGE) el cual debe tener configurados los webservices Cabezal y Linea.

2. Una vez creados ambos servicios en el conector se deben ingresar las URLs de ambos servicios en las variables **WS_AGESIC_TRAZABLIDAD_CABEZAL** y **WS_AGESIC_TRAZABLIDAD_LINEA** en el archivo de configuración ubicado en application/config/constants.php
3. Se deberá configurar un Cron para la ejecución diaria de un script de reencolamiento de trazas que no pudieron enviarse por fallas de conectividad o de algún otro mal funcionamiento. Se recomienda configurar dicho Cron para ser ejecutado cada 15 minutos todos los dias de la semana. Ejemplo del Cron:

```
*/15 * * * * php {RUTA_A_RAIZ_DE_SIMPLE}/index.php
tasks/trazabilidad reencolar
```

3.19 Permisos de archivos y directorios

Para establecer los permisos de archivos y directorios de la aplicación debemos seguir los siguientes pasos:

1. Ejecutamos en la terminal:

   $ cd <ruta_a_aplicacion>

   $ systemctl restart httpd.service

1. Otorgar permiso de escritura a los siguientes directorios y archivos:

   application/logs

   application/third_party/simplesaml/log

   application/third_party/simplesaml/log/simplesamlphp.log

   application/libraries/tcpdf/cache

   uploads (y subdirectorios)

3.20 Configuración Multicuenta

Para la configuración de Simple con múltiples cuentas se deben realizar las siguientes acciones:

1. Editar el archivo de configuración ubicado en application/config/config.php:

   $config['base_url'] = '' (debe quedar vacío).

   $config['main_domain'] = '' (debe ir el dominio).

3.21 Accediendo por primera vez

Para ingresar por primera vez y probar el correcto funcionamiento de la aplicación podremos utilizar los usuarios de administración y de gestión por defecto.

NOTA: Luego del primer ingreso se deben cambiar las contraseñas por defecto de ambos usuarios. Mantener las contraseñas por defecto es inseguro.

A continuación se listan los usuarios por defecto para el Backend tanto para administración como para gestión:

Administrador (backend principal)

- URL: **http://<aplicacion>/admin**
- Usuario: **admin@admin.com**
- Contraseña: **123456**

Manager (backend de gestión de cuentas y usuarios de cuentas)

- URL: **http://<aplicacion>/manager**
- Usuario: **admin@admin.com**
- Contraseña: **123456**

Frontend

- URL: **http://<aplicacion>/**

3.22 Anexos

Se anexan los siguientes documentos al manual:

- Hardening CENTOS 7 (*Documento_Hardcentos7.pdf*).