

Instalación y configuración de Mygestrans:

Web interna de usuarios

Versión: 4.1

0. Gestión de versiones.....	3
1. Información General del Servidor.....	4
2. Configuración de Red.....	4
2.1. Archivo /etc/hosts.....	4
2.2. Configuración de la Interfaz de Red (ens192).....	4
3. Configuración del Firewall (Firewalld).....	5
4. Configuración Inicial del Sistema.....	6
4.1. Configuración del Proxy en DNF.....	6
4.2. Registro del Servidor en Red Hat Subscription Management.....	7
4.3. Ejecución de Insights Client.....	7
5. Configuración de Almacenamiento (Volúmenes de Disco y NFS).....	8
5.1. Utilización del Espacio en Disco (df -h).....	8
5.2. Archivo /etc/fstab.....	9
6. Servidor Web Apache.....	10
6.1. Instalación de Apache.....	10
6.2. Archivo de Configuración del Host Virtual (myg-tms.dach800.dachser.com.conf).....	10
6.3. Gestión del Servicio Apache.....	19
7. Instalación y Configuración de PHP.....	20
7.1. Repositorio Remi y Paquetes PHP Instalados.....	20
7.2. Procedimiento de Instalación de PHP (con Gestión de alternatives).....	21
7.3. Archivos de Configuración de PHP.....	23
7.3.1. Configuración de /etc/opt/remi/php82/php.ini.....	23
7.3.2. Configuración de /etc/opt/remi/php82/php-fpm.d/www.conf.....	26
6.4. Archivos de Configuración de Apache para PHP.....	42
6.5. Gestión de Servicios PHP-FPM.....	43
6.6. Procedimiento para la Transición entre Versiones de PHP.....	43
8. Conectividad con Base de Datos SQL Server (ODBC / FreeTDS).....	45
8.1. Paquetes Necesarios.....	45
8.2. Archivos de Configuración.....	46
8.2.1. Contenido de /etc/freetds.conf.....	46
8.2.2. Contenido de /etc/odbc.ini.....	47
8.2.3. Contenido de /etc/odbcinst.ini.....	47
9. Configuración del Usuario admyG y Permisos.....	48
9.1. Creación del Usuario admyG.....	48
9.2. Permisos en el Document Root del Aplicativo Web.....	48
9.3. Permisos sudo para la Gestión de Servicios.....	48
9.4. Permisos de Lectura en los Registros de Apache.....	49
10. Instalación de Composer y Librerías PHP.....	50
10.1. Configuración del Proxy en yum.conf.....	50
10.2. Instalación de Composer.....	50
10.3. Instalación de Librerías PHP mediante Composer.....	51
11. Instalación de Open-VM-Tools.....	52
12. Instalación de CrowdStrike.....	52
13. Instalación del Agente Nagios NRPE.....	53
14. Script de Transferencia SFTP (sftp_out_myG.sh).....	54
14.1. Contenido del Script sftp_out_myG.sh.....	54
14.2. Permisos del Script.....	59
14.3. Directorio de Registros del Script.....	59

0. Gestión de versiones.

[illegible]

1. Información General del Servidor

El presente documento tiene como objetivo detallar la configuración e instalación del aplicativo web interno Mygestrans, desarrollado en PHP, sobre un servidor que opera con Red Hat 9.5.

- **Propósito:** La función principal de este servidor es alojar y gestionar la aplicación web interna Mygestrans, garantizando su operatividad.
- **Accesibilidad de Red:** La aplicación está diseñada exclusivamente para uso interno y no es accesible desde la red pública de Internet.
- **URL de Acceso:** El acceso a la aplicación se realiza a través de la siguiente dirección: <https://myg-tms.dach800.dachser.com>.
- **Sistema Operativo:** El servidor es una máquina virtual que ejecuta Red Hat Enterprise Linux 9.5.
- **Especificaciones de Hardware (Virtual):**
 - **Memoria RAM:** 8 GB
 - **vCPUs:** 4
- **Nombre de Host:** El nombre de host asignado al servidor es lx-mygestransapp02p.fro.int.

2. Configuración de Red

La configuración de red del servidor se detalla a continuación.

2.1. Archivo `/etc/hosts`

```
Unset
127.0.0.1    localhost
#::1        localhost
#192.168.5.104  gps.breogan-tte.com  mygestransweb01p
mygestransweb01p.dmz.int www.mygestransweb01p.dmz.int
```

2.2. Configuración de la Interfaz de Red (ens192)

```
Unset
GENERAL.DEVICE:          ens192
GENERAL.TYPE:            ethernet
GENERAL.HWADDR:          00:50:56:AD:59:05
GENERAL.MTU:              1500
GENERAL.STATE:            100 (connected)
```

```

GENERAL.CONNECTION:                ens192
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/ActiveConnection/2
WIRED-PROPERTIES.CARRIER:         on
IP4.ADDRESS[1]:                    10.100.16.220/23
IP4.GATEWAY:                       10.100.16.2
IP4.ROUTE[1]:                      dst = 10.100.16.0/23, nh =
0.0.0.0, mt = 100
IP4.ROUTE[2]:                      dst = 0.0.0.0/0, nh =
10.100.16.2, mt = 100
IP4.DNS[1]:                        10.100.20.103
IP4.DNS[2]:                        10.100.8.18
IP4.DNS[3]:                        10.100.8.19
IP4.SEARCHES[1]:                   des.int
IP4.SEARCHES[2]:                   fro.int
IP4.SEARCHES[3]:                   bac.int
IP4.SEARCHES[4]:                   dmz.int
IP4.SEARCHES[5]:                   svc.int
IP4.SEARCHES[6]:                   azkar.com
IP6.GATEWAY:                       --

GENERAL.DEVICE:                    lo
GENERAL.TYPE:                      loopback
GENERAL.HWADDR:                    00:00:00:00:00:00
GENERAL.MTU:                       65536
GENERAL.STATE:                     100 (connected (externally))
GENERAL.CONNECTION:                lo
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/ActiveConnection/1
IP4.ADDRESS[1]:                    127.0.0.1/8
IP4.GATEWAY:                       --
IP6.GATEWAY:                       --

```

3. Configuración del Firewall (Firewalld)

El firewall del servidor ha sido configurado para permitir el tráfico esencial para la operación de la aplicación.

Unset

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: dhcpv6-client http https ssh
  ports: 9001/tcp 3306/tcp 5666/tcp 9090/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Observación: El puerto `3306/tcp` se encuentra abierto para facilitar la conexión a la base de datos MySQL externa, la cual es un requisito para la funcionalidad de la aplicación.

4. Configuración Inicial del Sistema

4.1. Configuración del Proxy en DNF

Para permitir que el servidor acceda a los repositorios de Red Hat y otros recursos externos a través del proxy de la red, es necesario configurar la directiva de proxy en el archivo de configuración de DNF (`/etc/dnf/dnf.conf`).

Edite el archivo `dnf.conf` utilizando un editor de texto como `nano`:

Unset

```
sudo nano /etc/dnf/dnf.conf
```

Añada o modifique la siguiente línea en la sección `[main]`:

```
Unset
proxy=http://lx-reposrv01p.fro.int:8080
```

Guarde los cambios y cierre el archivo.

4.2. Registro del Servidor en Red Hat Subscription Management

Para obtener actualizaciones de software y asegurar la gestión centralizada del servidor por parte de Red Hat, es imprescindible registrarlo utilizando el Subscription Manager.

Registre el sistema con sus credenciales de cuenta de Red Hat:

```
Unset
sudo subscription-manager register --username=<tu_usuario_redhat>
--password=<tu_contraseña>
```

Reemplace `<tu_usuario_redhat>` y `<tu_contraseña>` con sus credenciales de Red Hat.

Si ya dispone de una suscripción asignada, puede adjuntarla automáticamente:

```
Unset
sudo subscription-manager attach --auto
```

Verifique el estado de la suscripción para confirmar el registro exitoso:

```
Unset
sudo subscription-manager status
```

4.3. Ejecución de Insights Client

Red Hat Insights es un servicio diseñado para el análisis continuo de plataformas y aplicaciones, con el fin de identificar posibles riesgos de seguridad y rendimiento. Para habilitar este servicio, instale y ejecute el cliente de Insights.

Instale el cliente de Insights:

```
Unset
sudo dnf install insights-client -y
```

Ejecute el cliente de Insights para generar el informe inicial y registrar el sistema en la consola de Insights:

```
Unset
sudo insights-client
```

Este comando recopilará la información necesaria del sistema y la transmitirá a Red Hat Insights. Tras la ejecución, debería aparecer un mensaje confirmando el registro del sistema.

5. Configuración de Almacenamiento (Volúmenes de Disco y NFS)

El servidor utiliza volúmenes lógicos (LVM) para la gestión del espacio en disco y cuenta con un montaje NFS para el almacenamiento de datos de la aplicación.

5.1. Utilización del Espacio en Disco (**df -h**)

```
Unset
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  4.0M        0   4.0M   0% /dev
tmpfs                     3.8G        0   3.8G   0% /dev/shm
tmpfs                    1.5G    9.1M   1.5G   1% /run
/dev/mapper/vg00-lvroot    40G    2.2G   38G    6% /
/dev/mapper/vg00-lvusr    8.0G    4.0G   4.1G   50% /usr
/dev/mapper/vg00-lvopt    50G    4.2G   46G    9% /opt
/dev/mapper/vg00-lvvar    30G    8.5G   22G   29% /var
/dev/mapper/vg00-lvtmp    2.0G     33M   2.0G    2% /tmp
/dev/sda1                 497M   408M    89M   83% /boot
```



```
nas-pro.bac.int:/MyGestrans 250G 206G 45G 83%
/var/www/myg-tms.dach800.dachser.com/hnas
tmpfs 766M 0 766M 0% /run/user/0
```

5.2. Archivo `/etc/fstab`

El archivo `/etc/fstab` describe la configuración de los sistemas de archivos montados, incluyendo el recurso NFS.

```
Unset
#
# /etc/fstab
# Created by anaconda on Wed Nov 18 11:13:40 2015
#
# Accessible filesystems, by reference, are maintained under
# '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for
# more info
#
/dev/mapper/vg00-lvroot / xfs defaults
0 0
UUID=a54967a2-15c2-412e-a2d1-957e81149ea8 /boot xfs
defaults 0 0
/dev/mapper/vg00-lvtmp /tmp xfs defaults
0 0
/dev/mapper/vg00-lvvar /var xfs defaults
0 0
/dev/mapper/vg00-lvopt /opt xfs defaults
0 0
/dev/mapper/vg00-lvusr /usr xfs defaults
0 0
/dev/mapper/vg00-lvswap swap swap defaults
0 0
```

```
nas-pro.bac.int:/MyGestrans
/var/www/myg-tms.dach800.dachser.com/hnas      nfs
defaults,auto,rw 0 0
```

6. Servidor Web Apache

El servidor web empleado en esta infraestructura es Apache HTTP Server.

6.1. Instalación de Apache

Para instalar Apache en Red Hat 9.5, se deben ejecutar los siguientes comandos:

```
Unset
sudo dnf update -y
sudo dnf install httpd httpd-tools mod_ssl -y
```

6.2. Archivo de Configuración del Host Virtual (myg-tms.dach800.dachser.com.conf)

La configuración específica del aplicativo web se encuentra en el archivo `/etc/httpd/conf.d/myg-tms.dach800.dachser.com.conf`.

Contenido del archivo:

```
Unset
#
# When we also provide SSL we have to listen to the
# the HTTPS port in addition.
#
Listen 443 https

##
##  SSL Global Context
```

```

##
## All SSL configuration in this context applies both to
## the main server and all SSL-enabled virtual hosts.
##

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program ('builtin' is a internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog exec:/usr/libexec/httpd-ssl-pass-dialog

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).
SSLSessionCache shmcb:/run/httpd/sslcache(512000)
SSLSessionCacheTimeout 300

# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the
# SSL library. The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough
# entropy
# is available. This means you then cannot use the /dev/random
# device
# because it would lead to very long connection times (as long
# as
# it requires to make more entropy available). But usually
# those
# platforms additionally provide a /dev/urandom device which
# doesn't
# block. So, if available, use this one instead. Read the
# mod_ssl User
# Manual for more details.
SSLRandomSeed startup file:/dev/urandom 256
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random 512

```

```
#SSLRandomSeed connect file:/dev/random 512
#SSLRandomSeed connect file:/dev/urandom 512

#
# Use "SSLCryptoDevice" to enable any supported hardware
# accelerators. Use "openssl engine -v" to list supported
# engine names. NOTE: If you enable an accelerator and the
# server does not start, consult the error logs and ensure
# your accelerator is functioning properly.
#
SSLCryptoDevice builtin
#SSLCryptoDevice ubsec

##
## SSL Virtual Host Context
##

<VirtualHost *:443>

# General setup for the virtual host, inherited from global
configuration
DocumentRoot "/var/www/myg-tms.dach800.dachser.com/"
#ServerName myg-tms.dach800.dachser.com:443
ServerName myg-tms-new.dachser.com:443

# Use separate log files for the SSL virtual host; note that
LogLevel
# is not inherited from httpd.conf.

ErrorLog
${APACHE_LOG_DIR}/myg-tms.dach800.dachser.com_ssl_error.log
CustomLog
${APACHE_LOG_DIR}/myg-tms.dach800.dachser.com_ssl_access.log
combined
```

```
#ErrorLog logs/ssl_error_log
#TransferLog logs/ssl_access_log
LogLevel warn

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Protocol support:
#   List the enable protocol levels with which clients will be able
#   to
#   connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA

#   Speed-optimized SSL Cipher configuration:
#   If speed is your main concern (on busy HTTPS servers e.g.),
#   you might want to force clients to specific, performance
#   optimized ciphers. In this case, prepend those ciphers
#   to the SSLCipherSuite list, and enable SSLHonorCipherOrder.
#   Caveat: by giving precedence to RC4-SHA and AES128-SHA
#   (as in the example below), most connections will no longer
#   have perfect forward secrecy - if the server's key is
#   compromised, captures of past or future traffic must be
#   considered compromised, too.
#SSLCipherSuite RC4-SHA:AES128-SHA:HIGH:MEDIUM:!aNULL:!MD5
#SSLHonorCipherOrder on

#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  A new
```

```
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/ssl/certs/myg-tms.dach800.dachser.com.cer

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile
/etc/ssl/certs/myg-tms.dach800.dachser.com.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
SSLCertificateChainFile /etc/ssl/certs/DACHSER-CA-1.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# SSLCACertificateFile /etc/ssl/certs/myg-tms.p7b

# Client Authentication (Type):
# Client certificate verification type and depth. Types are
# none, optional, require and optional_no_ca. Depth is a
# number which specifies how deeply to verify the certificate
# issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10

# Access Control:
# With SSLRequire you can do per-directory access control based
```

```

#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives. The syntax is a
#   mixture between C and Perl. See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#               and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#               and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"}
#               \
#               and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#               and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20      )
#               \
#               or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#       Translate the client X.509 into a Basic Authorisation.
#       This means that
#       the standard Auth/DBMAuth methods can be used for access
#       control. The
#       user name is the `one line' version of the client's X.509
#       certificate.
#       Note that no password is obtained from the user. Every
#       entry in the user
#       file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#       This exports two additional environment variables:
#       SSL_CLIENT_CERT and
#       SSL_SERVER_CERT. These contain the PEM-encoded certificates
#       of the
#       server (always existing) and the client (only existing when
#       client
#       authentication is used). This can be used to import the
#       certificates

```

```

#      into CGI scripts.
#      o StdEnvVars:
#          This exports the standard SSL/TLS related `SSL_`
environment variables.
#          Per default this exportation is switched off for
performance reasons,
#          because the extraction step is an expensive operation and
is usually
#          useless for serving static content. So one usually enables
the
#          exportation for CGI and SSI requests only.
#      o StrictRequire:
#          This denies access when "SSLRequireSSL" or "SSLRequire"
applied even
#          under a "Satisfy any" situation, i.e. when it applies
access is denied
#          and no other module can change it.
#      o OptRenegotiate:
#          This enables optimized SSL connection renegotiation
handling when SSL
#          directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<Files ~ "\.(cgi|shhtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>

<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

    <Directory /var/www/myg-tms.dach800.dachser.com>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        AddDefaultCharset UTF-8

```


</Directory>

```
# The safe and default but still SSL/TLS standard compliant
shutdown
# approach is that mod_ssl sends the close notify alert but
doesn't wait for
# the close notify alert from client. When you need a different
shutdown
# approach you can use one of the following variables:
# o ssl-unclean-shutdown:
# This forces an unclean shutdown when the connection is
closed, i.e. no
# SSL close notify alert is send or allowed to received.
This violates
# the SSL/TLS standard but is needed for some brain-dead
browsers. Use
# this when you receive I/O errors because of the standard
approach where
# mod_ssl sends the close notify alert.
# o ssl-accurate-shutdown:
# This forces an accurate shutdown when the connection is
closed, i.e. a
# SSL close notify alert is send and mod_ssl waits for the
close notify
# alert of the client. This is 100% SSL/TLS standard
compliant, but in
# practice often causes hanging connections with brain-dead
browsers. Use
# this only for browsers where you know that their SSL
implementation
# works correctly.
# Notice: Most problems of broken clients are also related to
the HTTP
```

```

# keep-alive facility, so you usually additionally want to
disable
# keep-alive for those clients, too. Use variable "nokeepalive"
for this.
# Similarly, one has to force some clients to use HTTP/1.0 to
workaround
# their broken HTTP/1.1 implementation. Use variables
"downgrade-1.0" and
# "force-response-1.0" for this.
BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Per-Server Logging:
# The home of a custom SSL log file. Use this when you want a
# compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

```

Ubicaciones Clave de Apache:

- **Document Root:** `/var/www/myg-tms.dach800.dachser.com/`
- **Certificado SSL:** `/etc/ssl/certs/myg-tms.dach800.dachser.com.cer`
- **Clave SSL:** `/etc/ssl/certs/myg-tms.dach800.dachser.com.key`
- **Cadena de Certificados SSL:** `/etc/ssl/certs/DACHSER-CA-1.crt`
- **Registros de Errores:**
`${APACHE_LOG_DIR}/myg-tms.dach800.dachser.com_ssl_error.log`
 (ubicación definida por la variable de entorno de Apache, comúnmente
`/var/log/httpd/`)
- **Registros de Acceso:**
`${APACHE_LOG_DIR}/myg-tms.dach800.dachser.com_ssl_access.log`
 (ubicación definida por la variable de entorno de Apache, comúnmente
`/var/log/httpd/`)

- **Registros de Solicitudes SSL:** `logs/ssl_request_log` (ubicación relativa a la directiva `ServerRoot` de Apache)

6.3. Gestión del Servicio Apache

- **Verificación de la Sintaxis de Configuración:**

Unset

```
sudo apachectl configtest
# 0
sudo httpd -t
```

- **Detención del Servicio Apache:**

Unset

```
sudo systemctl stop httpd
```

- **Inicio del Servicio Apache:**

Unset

```
sudo systemctl start httpd
```

- **Reinicio del Servicio Apache:**

Unset

```
sudo systemctl restart httpd
```

- **Recarga de la Configuración (sin reiniciar el servicio):**

Unset

```
sudo systemctl reload httpd
```

-

Habilitación del Servicio Apache para el Arranque del Sistema:

Unset

```
sudo systemctl enable httpd
```

7. Instalación y Configuración de PHP

El servidor dispone de dos versiones de PHP: PHP 8.2 (actualmente activa) y PHP 8.3 (designada para futura producción), gestionadas a través del sistema [alternatives](#).

7.1. Repositorio Remi y Paquetes PHP Instalados

Las versiones de PHP se instalan desde el repositorio Remi.

Paquetes PHP instalados actualmente:

Unset

```
oniguruma5php-6.9.10-1.el9.remi.x86_64
gd3php-2.3.3-9.el9.remi.x86_64
php82-runtime-8.2-5.el9.remi.x86_64
php82-php-common-8.2.28-1.el9.remi.x86_64
php82-php-pdo-8.2.28-1.el9.remi.x86_64
php82-php-mysqlnd-8.2.28-1.el9.remi.x86_64
php82-php-cli-8.2.28-1.el9.remi.x86_64
php82-php-fpm-8.2.28-1.el9.remi.x86_64
php82-php-gd-8.2.28-1.el9.remi.x86_64
php82-php-mbstring-8.2.28-1.el9.remi.x86_64
php82-php-openssl-8.2.28-1.el9.remi.x86_64
php82-php-xml-8.2.28-1.el9.remi.x86_64
php83-runtime-8.3-1.el9.remi.x86_64
php82-php-odbc-8.2.28-1.el9.remi.x86_64
```

```
php82-php-ldap-8.2.28-1.el9.remi.x86_64
php83-php-common-8.3.21-1.el9.remi.x86_64
php83-php-pdo-8.3.21-1.el9.remi.x86_64
php83-php-mysqlnd-8.3.21-1.el9.remi.x86_64
php83-php-odbc-8.3.21-1.el9.remi.x86_64
php83-php-cli-8.3.21-1.el9.remi.x86_64
php83-php-fpm-8.3.21-1.el9.remi.x86_64
php83-php-gd-8.3.21-1.el9.remi.x86_64
php83-php-ldap-8.3.21-1.el9.remi.x86_64
php83-php-mbstring-8.3.21-1.el9.remi.x86_64
php83-php-openssl-8.3.21-1.el9.remi.x86_64
php83-php-xml-8.3.21-1.el9.remi.x86_64
php82-php-gmp-8.2.28-1.el9.remi.x86_64
php83-php-gmp-8.3.21-1.el9.remi.x86_64
php82-php-pear-1.10.1-1.el9.remi.x86_64
php83-php-pear-1.10.1-1.el9.remi.x86_64
```

7.2. Procedimiento de Instalación de PHP (con Gestión de **alternatives**)

Para instalar PHP desde el repositorio Remi y configurar **alternatives**, siga los siguientes pasos:

1. **Habilitación del Repositorio Remi:**

Unset

```
sudo dnf install
https://rpms.remi.fedora.com/enterprise/remi-release-9.rpm -y
sudo dnf config-manager --set-enabled remi
```

2.

Instalación de los Paquetes PHP Requeridos (ej. PHP 8.2 y PHP 8.3):

Unset

Para PHP 8.2

```
sudo dnf install php82-php-fpm php82-php-cli php82-php-common  
php82-php-pdo php82-php-mysqlnd php82-php-gd php82-php-mbstring  
php82-php-openssl php82-php-xml php82-php-odbc php82-php-ldap  
php82-php-gmp php82-php-pear -y
```

Para PHP 8.3

```
sudo dnf install php83-php-fpm php83-php-cli php83-php-common  
php83-php-pdo php83-php-mysqlnd php83-php-gd php83-php-mbstring  
php83-php-openssl php83-php-xml php83-php-odbc php83-php-ldap  
php83-php-gmp php83-php-pear -y
```

3.

4.

(Se recomienda instalar todos los módulos PHP necesarios para la funcionalidad completa de la aplicación).

5. **Configuración de `alternatives` para PHP CLI:** Los paquetes del repositorio Remi suelen configurar automáticamente `alternatives` para el binario `php`. La configuración puede verificarse mediante:

Unset

```
sudo alternatives --config php
```

6.

7.

En caso de que las opciones no estén configuradas, pueden añadirse manualmente (ajustando las rutas si difieren de las predeterminadas):

Unset

```
sudo alternatives --install /usr/bin/php php /usr/bin/php82 82  
sudo alternatives --install /usr/bin/php php /usr/bin/php83 83  
# Posteriormente, seleccione la versión deseada:
```

```
sudo alternatives --config php
```

7.3. Archivos de Configuración de PHP

Cada versión de PHP mantiene su propia configuración en directorios separados:

- `/etc/opt/remi/php82/`
- `/etc/opt/remi/php83/`

Dentro de cada directorio, se encuentran los archivos `php.ini` y `php-fpm.conf`. Se asume que la configuración es idéntica entre las versiones para el correcto funcionamiento de la aplicación Mygestrans.

7.3.1. Configuración de `/etc/opt/remi/php82/php.ini`

El archivo `php.ini` es fundamental para el comportamiento de PHP. A continuación, se destacan directivas clave y sus valores actuales. Es importante señalar que estos valores pueden diferir de una instalación PHP estándar o por defecto, ya que han sido ajustados para el entorno específico de la aplicación.

- **`engine = On`**: Habilita el motor de scripting PHP.
- **`short_open_tag = Off`**: Deshabilita las etiquetas de apertura cortas (`<? ?>`), promoviendo el uso de `<?php ?>`.
- **`output_buffering = 4096`**: Habilita el almacenamiento en búfer de salida con un tamaño de 4KB, lo cual es una práctica común para optimizar el rendimiento en entornos de producción.
- **`zlib.output_compression = Off`**: Deshabilita la compresión de salida mediante la librería zlib.
- **`implicit_flush = Off`**: Deshabilita el vaciado implícito del búfer de salida después de cada bloque de salida, lo cual es una buena práctica para el rendimiento.
- **`serialize_precision = -1`**: Configura la precisión de serialización de números flotantes para preservar la fidelidad de los datos.
- **`zend.enable_gc = On`**: Habilita el recolector de basura para referencias circulares.
- **`zend.exception_ignore_args = On`**: En entornos de producción, se recomienda activar esta opción para evitar la exposición de información sensible en los rastreos de pila de excepciones.
- **`zend.exception_string_param_max_len = 0`**: En producción, se aconseja establecer este valor en 0 para reducir la salida de información sensible en los rastreos de pila.

- **expose_php = On**: Determina si PHP revela su firma en las cabeceras del servidor web.
- **max_execution_time = 30**: Establece el tiempo máximo de ejecución para cada script en segundos.
- **max_input_time = 60**: Define el tiempo máximo que un script puede dedicar al análisis de datos de solicitud.
- **memory_limit = 512M**: Limita la cantidad máxima de memoria que un script puede consumir.
- **error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT**: Configuración para el reporte de errores, excluyendo avisos de desuso y estándares de codificación.
- **display_errors = Off**: Deshabilita la visualización de errores en la salida web, lo cual es fundamental para la seguridad en entornos de producción.
- **display_startup_errors = Off**: Deshabilita la visualización de errores ocurridos durante el inicio de PHP.
- **log_errors = On**: Habilita el registro de errores en un archivo de log.
- **ignore_repeated_errors = Off**: No ignora mensajes repetidos.
- **report_memleaks = On**: Reporta fugas de memoria (aplicable solo en compilaciones de depuración).
- **variables_order = "GPCS"**: Define el orden de registro de las variables superglobales (GET, POST, COOKIE, SERVER).
- **request_order = "GP"**: Define el orden en que los datos superglobales (GET, POST) se registran en el array REQUEST.
- **register_argc_argv = Off**: Deshabilita el registro de `$argv` y `$argc` para scripts web (activado por defecto para CLI).
- **auto_globals_jit = On**: Crea variables globales "Just In Time" para optimizar el rendimiento.
- **post_max_size = 8M**: Establece el tamaño máximo de los datos POST que PHP aceptará.
- **file_uploads = On**: Permite la subida de archivos HTTP.
- **upload_max_filesize = 2M**: Define el tamaño máximo permitido para archivos subidos.
- **max_file_uploads = 20**: Establece el número máximo de archivos que pueden ser subidos en una única solicitud.
- **allow_url_fopen = On**: Permite el tratamiento de URLs como archivos.
- **allow_url_include = Off**: Deshabilita la inclusión de URLs como archivos por motivos de seguridad.
- **default_socket_timeout = 60**: Define el tiempo de espera predeterminado para flujos basados en sockets.
- **sendmail_path = /usr/sbin/sendmail -t -i**: Especifica la ruta al binario de sendmail para la función `mail()`.

- **mail.add_x_header = Off**: Deshabilita la adición de la cabecera X-PHP-Originating-Script.
- **odbc.allow_persistent = On**: Permite conexiones persistentes ODBC.
- **odbc.check_persistent = On**: Verifica la validez de las conexiones persistentes antes de su reutilización.
- **odbc.max_persistent = -1**: Establece el número máximo de enlaces persistentes ODBC (ilimitado).
- **odbc.max_links = -1**: Establece el número máximo de enlaces ODBC (persistentes + no persistentes, ilimitado).
- **odbc.defaultlrl = 4096**: Configura el manejo de campos LONG (número de bytes a variables).
- **odbc.defaultbinmode = 1**: Configura el manejo de datos binarios (retornar tal cual).
- **mysqli.max_persistent = -1**: Establece el número máximo de enlaces persistentes MySQLi (ilimitado).
- **mysqli.allow_persistent = On**: Permite conexiones persistentes MySQLi.
- **mysqli.max_links = -1**: Establece el número máximo de enlaces MySQLi (ilimitado).
- **mysqli.default_port = 3306**: Define el puerto por defecto para `mysqli_connect()`.
- **mysqli.reconnect = Off**: Deshabilita la reconexión automática.
- **mysqlnd.collect_statistics = On**: Habilita la recolección de estadísticas generales por mysqlnd.
- **pcre.jit = 0**: Deshabilita la compilación JIT (Just-In-Time) para expresiones regulares PCRE.
- **session.save_handler = files**: Especifica el manejador para almacenar y recuperar datos de sesión.
- **session.use_strict_mode = 0**: Deshabilita el modo de sesión estricto.
- **session.use_cookies = 1**: Habilita el uso de cookies para la gestión de sesiones.
- **session.use_only_cookies = 1**: Fuerza a PHP a utilizar únicamente cookies para el ID de sesión.
- **session.name = PHPSESSID**: Define el nombre de la sesión (utilizado como nombre de cookie).
- **session.auto_start = 0**: Evita el inicio automático de la sesión al comienzo de la solicitud.
- **session.cookie_lifetime = 0**: Establece la vida útil de la cookie de sesión (hasta el cierre del navegador).
- **session.cookie_path = /**: Define la ruta para la cual la cookie es válida.
- **session.gc_probability = 1**: Establece la probabilidad de que el proceso de "recolección de basura" se inicie en cada inicialización de sesión.

- **session.gc_divisor = 1000**: Divisor para la probabilidad de recolección de basura.
- **session.gc_maxlifetime = 1440**: Los datos de sesión se consideran "basura" después de este número de segundos.
- **session.cache_limiter = nocache**: Determina los aspectos del almacenamiento en caché HTTP.
- **session.cache_expire = 180**: Los documentos de sesión expiran después de N minutos.
- **session.use_trans_sid = 0**: Deshabilita el soporte de SID transparente.
- **session.sid_length = 26**: Longitud del ID de sesión.
- **session.sid_bits_per_character = 5**: Bits almacenados en cada carácter al convertir datos hash binarios.
- **zend.assertions = -1**: Las aserciones se encuentran deshabilitadas en entornos de producción.
- **soap.wsdl_cache_enabled = 1**: Habilita la función de caché de WSDL.
- **soap.wsdl_cache_dir = "/tmp"**: Directorio donde la extensión SOAP almacenará los archivos de caché.
- **soap.wsdl_cache_ttl = 86400**: Tiempo de vida (TTL) para los archivos de caché WSDL (24 horas).
- **soap.wsdl_cache_limit = 5**: Límite de tamaño del caché (número máximo de archivos WSDL a cachear).
- **ldap.max_links = -1**: Número máximo de enlaces LDAP abiertos (ilimitado).

7.3.2. Configuración de `/etc/opt/remi/php82/php-fpm.d/www.conf`

El archivo `www.conf` dentro del directorio de configuración de PHP-FPM para la versión 8.2 ha sido modificado para dirigir los errores de PHP al log de errores de Apache. La línea `php_admin_value[error_log]` ha sido comentada para lograr este objetivo.

Unset

```
; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be
replaced by the
; pool name ('www' here)
[www]

; Per pool prefix
; It only applies on the following directives:
; - 'access.log'
```

```

; - 'slowlog'
; - 'listen' (unixsocket)
; - 'chroot'
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (or @php_fpm_prefix@) applies
instead.
; Note: This directive can also be relative to the global prefix.
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the
default user's group
;       will be used.
; RPM: apache user chosen to provide access to the same
directories as httpd
user = apache
; RPM: Keep a group allowed to write in log dir.
group = apache

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'   - to listen on a TCP socket to a
specific IPv4 address on
;                           a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a
specific IPv6 address on
;                           a specific port;
;   'port'                - to listen on a TCP socket to all
addresses
;                           (IPv6 and IPv4-mapped) on a specific
port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.

```

```
listen = /var/opt/remi/php82/run/php-fpm/www.sock

; Set listen(2) backlog.
; Default Value: 511
;listen.backlog = 511

; Set permissions for unix socket, if one is used. In Linux,
read/write
; permissions must be set in order to allow connections from a
web server.
; Default Values: user and group are set as the running user
;                  mode is set to 0660
;listen.owner = nobody
;listen.group = nobody
;listen.mode = 0660

; When POSIX Access Control Lists are supported you can set them
using
; these options, value is a comma separated list of user/group
names.
; When set, listen.owner and listen.group are ignored
listen.acl_users = apache
;listen.acl_groups =

; List of addresses (IPv4/IPv6) of FastCGI clients which are
allowed to connect.
; Equivalent to the FCGI_WEB_SERVER_ADDRS environment variable in
the original
; PHP FCGI (5.2.2+). Makes sense only with a tcp listening
socket. Each address
; must be separated by a comma. If this value is left blank,
connections will be
; accepted from any ip address.
; Default Value: any
listen.allowed_clients = 127.0.0.1
```

```

; Specify the nice(2) priority to apply to the pool processes
(only if set)
; The value can vary from -19 (highest priority) to 20 (lower
priority)
; Note: - It will only work if the FPM master process is launched
as root
;         - The pool processes will inherit the master process
priority
;         unless it specified otherwise
; Default Value: no set
; process.priority = -19

; Set the process dumpable flag (PR_SET_DUMPABLE prctl) even if
the process user
; or group is different than the master process user. It allows
to create process
; core dump and ptrace the process for the pool user.
; Default Value: no
; process.dumpable = yes

; Choose how the process manager will control the number of child
processes.
; Possible Values:
;   static - a fixed number (pm.max_children) of child
processes;
;   dynamic - the number of child processes are set dynamically
based on the
;             following directives. With this process management,
there will be
;             always at least 1 children.
;             pm.max_children      - the maximum number of
children that can
;                                     be alive at the same time.
;             pm.start_servers    - the number of children
created on startup.

```

```

;          pm.min_spare_servers - the minimum number of
children in 'idle'
;          state (waiting to process).
If the number
;          of 'idle' processes is less
than this
;          number then some children
will be created.
;          pm.max_spare_servers - the maximum number of
children in 'idle'
;          state (waiting to process).
If the number
;          of 'idle' processes is
greater than this
;          number then some children
will be killed.
;  ondemand - no children are created at startup. Children will
be forked when
;          new requests will connect. The following parameter
are used:
;          pm.max_children          - the maximum number of
children that
;          can be alive at the
same time.
;          pm.process_idle_timeout  - The number of seconds
after which
;          an idle process will be
killed.
; Note: This value is mandatory.
pm = dynamic

; The number of child processes to be created when pm is set to
'static' and the
; maximum number of child processes when pm is set to 'dynamic'
or served. Equivalent to the ApacheMaxClients directive with
mpm_prefork.

```

```
; Equivalent to the PHP_FCGI_CHILDREN environment variable in the
original PHP
; CGI. The below defaults are based on a server without much
resources. Don't
; forget to tweak pm.* to fit your needs.
; Note: Used when pm is set to 'static', 'dynamic' or 'ondemand'
; Note: This value is mandatory.
pm.max_children = 50
```

```
; The number of child processes created on startup.
; Note: Used only when pm is set to 'dynamic'
; Default Value: min_spare_servers + (max_spare_servers -
min_spare_servers) / 2
pm.start_servers = 5
```

```
; The desired minimum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.min_spare_servers = 5
```

```
; The desired maximum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.max_spare_servers = 35
```

```
; The number of seconds after which an idle process will be
killed.
; Note: Used only when pm is set to 'ondemand'
; Default Value: 10s
;pm.process_idle_timeout = 10s;
```

```
; The number of requests each child process should execute before
respawning.
; This can be useful to work around memory leaks in 3rd party
libraries. For
```

```

; endless request processing specify '0'. Equivalent to
PHP_FCGI_MAX_REQUESTS.
; Default Value: 0
;pm.max_requests = 500

; The URI to view the FPM status page. If this value is not set,
no URI will be
; recognized as a status page. It shows the following
information:
;   pool                - the name of the pool;
;   process manager      - static, dynamic or ondemand;
;   start time           - the date and time FPM has started;
;   start since          - number of seconds since FPM has
started;
;   accepted conn        - the number of request accepted by the
pool;
;   listen queue         - the number of request in the queue of
pending
;                           connections (see backlog in
listen(2));
;   max listen queue     - the maximum number of requests in the
queue
;                           of pending connections since FPM has
started;
;   listen queue len     - the size of the socket queue of
pending connections;
;   idle processes       - the number of idle processes;
;   active processes     - the number of active processes;
;   total processes      - the number of idle + active processes;
;   max active processes - the maximum number of active processes
since FPM
;                           has started;
;   max children reached - number of times, the process limit has
been reached,
;                           when pm tries to start more children
(work only for

```



```
;                                     pm 'dynamic' and 'ondemand');
; Value are updated in real time.
; Example output:
;   pool:                               www
;   process manager:                     static
;   start time:                           01/Jul/2011:17:53:49 +0200
;   start since:                           62636
;   accepted conn:                         190460
;   listen queue:                           0
;   max listen queue:                       1
;   listen queue len:                       42
;   idle processes:                         4
;   active processes:                       11
;   total processes:                        15
;   max active processes:                   12
;   max children reached:                   0
;
; By default the status page output is formatted as text/plain.
Passing either
; 'html', 'xml' or 'json' in the query string will return the
corresponding
; output syntax. Example:
;   http://www.foo.bar/status
;   http://www.foo.bar/status?json
;   http://www.foo.bar/status?html
;   http://www.foo.bar/status?xml
;
; By default the status page only outputs short status. Passing
'full' in the
; query string will also return status for each pool process.
; Example:
;   http://www.foo.bar/status?full
;   http://www.foo.bar/status?json&full
;   http://www.foo.bar/status?html&full
;   http://www.foo.bar/status?xml&full
; The Full status returns for each process:
```

; pid	- the PID of the process;
; state	- the state of the process (Idle,
Running, ...);	
; start time	- the date and time the process has
started;	
; start since	- the number of seconds since the
process has started;	
; requests	- the number of requests the process has
served;	
; request duration	- the duration in μ s of the requests;
; request method	- the request method (GET, POST, ...);
; request URI:	- the request URI with the query
string;	
; content length:	- the content length of the request
(only with POST);	
; user:	- the user (PHP_AUTH_USER) (or '-' if
not set);	
; script:	- the main script called (or '-' if not
set);	
; last request cpu	- the %cpu the last request consumed
;	it's always 0 if the process is not in
Idle state	
;	because CPU calculation is done when
the request	
;	processing has terminated;
; last request memory	- the max amount of memory the last
request consumed	
;	it's always 0 if the process is not in
Idle state	
;	because memory calculation is done
when the request	
;	processing has terminated;
; If the process is in Idle state, then informations are related	
to the	
; last request the process has served. Otherwise informations are	
related to	

```

; the current request being served.
; Example output:
; *****
; pid:                31330
; state:              Running
; start time:         01/Jul/2011:17:53:49 +0200
; start since:        63087
; requests:           12808
; request duration:   1250261
; request method:     GET
; request URI:        /test_mem.php?N=10000
; content length:     0
; user:               -
; script:             /home/fat/web/docs/php/test_mem.php
; last request cpu:    0.00
; last request memory: 0
;
; Note: There is a real-time FPM status monitoring sample web
page available
;       It's available in: @EXPANDED_DATADIR@/fpm/status.html
;
; Note: The value must start with a leading slash (/). The value
can be
;       anything, but it may not be a good idea to use the .php
extension or it
;       may conflict with a real PHP file.
; Default Value: not set
;pm.status_path = /status

; The ping URI to call the monitoring page of FPM. If this value
is not set, no
; URI will be recognized as a ping page. This could be used to
test from outside
; that FPM is alive and responding, or to
; - create a graph of FPM availability (rrd or such);

```

```
; - remove a server from a group if it is not responding (load
balancing);
; - trigger alerts for the operating team (24/7).
; Note: The value must start with a leading slash (/). The value
can be
;     anything, but it may not be a good idea to use the .php
extension or it
;     may conflict with a real PHP file.
; Default Value: not set
;ping.path = /ping

; This directive may be used to customize the response of a ping
request. The
; response is formatted as text/plain with a 200 response code.
; Default Value: pong
;ping.response = pong

; The access log file
; Default: not set
;access.log = log/$pool.access.log

; The access log format.
; The following syntax is allowed
;   %: the '%' character
;   %C: %CPU used by the request
;       it can accept the following format:
;       - %{user}C for user CPU only
;       - %{system}C for system CPU only
;       - %{total}C  for user + system CPU (default)
;   %d: time taken to serve the request
;       it can accept the following format:
;       - %{seconds}d (default)
;       - %{milliseconds}d
;       - %{mili}d
;       - %{microseconds}d
;       - %{micro}d
```

```
; %e: an environment variable (same as $_ENV or $_SERVER)
;     it must be associated with embraces to specify the name of
the env
;     variable. Some examples:
;     - server specifics like: %{REQUEST_METHOD}e or
%{SERVER_PROTOCOL}e
;     - HTTP headers like: %{HTTP_HOST}e or %{HTTP_USER_AGENT}e
; %f: script filename
; %l: content-length of the request (for POST request only)
; %m: request method
; %M: peak of memory allocated by PHP
;     it can accept the following format:
;     - %{bytes}M (default)
;     - %{kilobytes}M
;     - %{kilo}M
;     - %{megabytes}M
;     - %{mega}M
; %n: pool name
; %o: output header
;     it must be associated with embraces to specify the name of
the header:
;     - %{Content-Type}o
;     - %{X-Powered-By}o
;     - %{Transfert-Encoding}o
;     - ....
; %p: PID of the child that serviced the request
; %P: PID of the parent of the child that serviced the request
; %q: the query string
; %Q: the '?' character if query string exists
; %r: the request URI (without the query string, see %q and %Q)
; %R: remote IP address
; %s: status (response code)
; %t: server time the request was received
;     it can accept a strftime(3) format:
;     %d/%b/%Y:%H:%M:%S %z (default)
```

```

;      The strftime(3) format must be encapsuled in a
%{<strftime_format>}t tag
;      e.g. for a ISO8601 formatted timestring, use:
%{%Y-%m-%dT%H:%M:%S%z}t
; %T: time the log has been written (the request has finished)
;      it can accept a strftime(3) format:
;      %d/%b/%Y:%H:%M:%S %z (default)
;      The strftime(3) format must be encapsuled in a
%{<strftime_format>}t tag
;      e.g. for a ISO8601 formatted timestring, use:
%{%Y-%m-%dT%H:%M:%S%z}t
; %u: remote user
;
; Default: "%R - %u %t \"%m %r\" %s"
;access.format = "%R - %u %t \"%m %r%Q%q\" %s %f %{mili}d
%{kilo}M %C%"

; The log file for slow requests
; Default Value: not set
; Note: slowlog is mandatory if request_slowlog_timeout is set
slowlog = /var/opt/remi/php82/log/php-fpm/www-slow.log

; The timeout for serving a single request after which a PHP
backtrace will be
; dumped to the 'slowlog' file. A value of '0s' means 'off'.
; Available units: s(econds)(default), m(inutes), h(ours), or
d(ays)
; Default Value: 0
;request_slowlog_timeout = 0

; Depth of slow log stack trace.
; Default Value: 20
;request_slowlog_trace_depth = 20

; The timeout for serving a single request after which the worker
process will

```

```
; be killed. This option should be used when the
'max_execution_time' ini option
; does not stop script execution for some reason. A value of '0'
means 'off'.
; Available units: s(econds)(default), m(inutes), h(ours), or
d(ays)
; Default Value: 0
;request_terminate_timeout = 0

; Set open file descriptor rlimit.
; Default Value: system defined value
;rlimit_files = 1024

; Set max core size rlimit.
; Possible Values: 'unlimited' or an integer greater or equal to
0
; Default Value: system defined value
;rlimit_core = 0

; Chroot to this directory at the start. This value must be
defined as an
; absolute path. When this value is not set, chroot is not used.
; Note: you can prefix with '$prefix' to chroot to the pool
prefix or one
; of its subdirectories. If the pool prefix is not set, the
global prefix
; will be used instead.
; Note: chrooting is a great security feature and should be used
whenever
; possible. However, all PHP paths will be relative to the
chroot
; (error_log, sessions.save_path, ...).
; Default Value: not set
;chroot =

; Chdir to this directory at the start.
```

```
; Note: relative path can be used.
; Default Value: current directory or / when chroot
;chdir = /var/www

; Redirect worker stdout and stderr into main error log. If not
set, stdout and
; stderr will be redirected to /dev/null according to FastCGI
specs.
; Note: on highloaded environment, this can cause some delay in
the page
; process time (several ms).
; Default Value: no
;catch_workers_output = yes

; Clear environment in FPM workers
; Prevents arbitrary environment variables from reaching FPM
worker processes
; by clearing the environment in workers before env vars
specified in this
; pool configuration are added.
; Setting to "no" will make all environment variables available
to PHP code
; via getenv(), $_ENV and $_SERVER.
; Default Value: yes
;clear_env = no

; Limits the extensions of the main script FPM will allow to
parse. This can
; prevent configuration mistakes on the web server side. You
should only limit
; FPM to .php extensions to prevent malicious users to use other
extensions to
; execute php code.
; Note: set an empty value to allow all extensions.
; Default Value: .php
;security.limit_extensions = .php .php3 .php4 .php5 .php7
```



```
; Pass environment variables like LD_LIBRARY_PATH. All $VARIABLEs
are taken from
; the current environment.
; Default Value: clean env
;env[HOSTNAME] = $HOSTNAME
;env[PATH] = /usr/local/bin:/usr/bin:/bin
;env[TMP] = /tmp
;env[TMPDIR] = /tmp
;env[TEMP] = /tmp
```

```
; Additional php.ini defines, specific to this pool of workers.
These settings
; overwrite the values previously defined in the php.ini. The
directives are the
; same as the PHP SAPI:
;   php_value/php_flag           - you can set classic ini
defines which can
;                               be overwritten from PHP call
'ini_set'.
;   php_admin_value/php_admin_flag - these directives won't be
overwritten by
;                               PHP call 'ini_set'
; For php_*flag, valid values are on, off, 1, 0, true, false, yes
or no.
```

```
; Defining 'extension' will load the corresponding shared
extension from
; extension_dir. Defining 'disable_functions' or
'disable_classes' will not
; overwrite previously defined php.ini values, but will append
the new value
; instead.
```

```
; Note: path INI options can be relative and will be expanded
with the prefix
```

```

; (pool, global or @prefix@)

; Default Value: nothing is defined by default except the values
in php.ini and
;                specified at startup with the -d argument
;php_admin_value[sendmail_path] = /usr/sbin/sendmail -t -i -f
www@my.domain.com
php_flag[display_errors] = off
;php_admin_value[error_log] =
/var/opt/remi/php82/log/php-fpm/www-error.log
php_admin_flag[log_errors] = on
;php_admin_value[memory_limit] = 128M

; Set the following data paths to directories owned by the FPM
process user.
;
; Do not change the ownership of existing system directories, if
the process
; user does not have write permission, create dedicated
directories for this
; purpose.
;
; See warning about choosing the location of these directories on
your system
; at http://php.net/session.save-path
php_value[session.save_handler] = files
php_value[session.save_path]    =
/var/opt/remi/php82/lib/php/session
php_value[soap.wsdl_cache_dir]   =
/var/opt/remi/php82/lib/php/wsdlcache
;php_value[opcache.file_cache]   =
/var/opt/remi/php82/lib/php/opcache

```

6.4. Archivos de Configuración de Apache para PHP

En el directorio `/etc/httpd/conf.d/`, existen dos archivos relevantes para la integración de PHP con Apache:

- `/etc/httpd/conf.d/php82-php.conf` (activo cuando PHP 8.2 está en uso)
- `/etc/httpd/conf.d/php83-php.conf.inactive` (inactivo, se activa al cambiar a PHP 8.3)

6.5. Gestión de Servicios PHP-FPM

Los servicios FPM para cada versión deben estar habilitados para el inicio automático del sistema:

- **Para PHP 8.2:**

Unset

```
sudo systemctl enable php82-php-fpm.service
sudo systemctl start php82-php-fpm.service
```

-

- **Para PHP 8.3:**

Unset

```
sudo systemctl enable php83-php-fpm.service
sudo systemctl start php83-php-fpm.service
```

6.6. Procedimiento para la Transición entre Versiones de PHP

Para cambiar la versión activa de PHP de 8.2 a 8.3 (o viceversa, ajustando los nombres de los archivos correspondientes), se deben seguir los siguientes pasos:

1. **Detener el Servicio Apache:**

Unset

```
sudo systemctl stop httpd
```

2.

Detener y Deshabilitar el Servicio FPM de la Versión Actual (ej. PHP 8.2):

Unset

```
sudo systemctl stop php82-php-fpm.service
sudo systemctl disable php82-php-fpm.service
```

3.

Renombrar los Archivos de Configuración de Apache para PHP:

- Para desactivar la versión actual y activar la nueva:

Unset

```
sudo mv /etc/httpd/conf.d/php82-php.conf
/etc/httpd/conf.d/php82-php.conf.inactive
sudo mv /etc/httpd/conf.d/php83-php.conf.inactive
/etc/httpd/conf.d/php83-php.conf
```

○

○

(El proceso inverso aplica para la transición de 8.3 a 8.2)

4. **Seleccionar la Versión de PHP Deseada mediante **alternatives**:**

Unset

```
sudo alternatives --config php
```

5.

6.

Seleccione el número correspondiente a la versión de PHP deseada (ej. PHP 8.3).

7. **Iniciar y Habilitar el Servicio FPM de la Nueva Versión (ej. PHP 8.3):**

Unset

```
sudo systemctl start php83-php-fpm.service  
sudo systemctl enable php83-php-fpm.service
```

8.

Iniciar el Servicio Apache:

Unset

```
sudo systemctl start httpd
```

8. Conectividad con Base de Datos SQL Server (ODBC / FreeTDS)

La aplicación establece conexión con una base de datos SQL Server 2008 R2 externa a través de ODBC y FreeTDS.

8.1. Paquetes Necesarios

Se deben instalar los siguientes paquetes para asegurar la conectividad ODBC y FreeTDS:

Unset

```
sudo dnf install unixODBC freetds freetds-libs -y
```

Paquetes instalados actualmente:

Unset

```
unixODBC-2.3.11-1.rh.x86_64
```

```
freetds-libs-1.4.23-1.el9.x86_64
freetds-1.4.23-1.el9.x86_64
```

8.2. Archivos de Configuración

8.2.1. Contenido de `/etc/freetds.conf`

```
Unset
#
# This file is installed by FreeTDS if no file by the same
# name is found in the installation directory.
#
# For information about the layout of this file and its settings,
# see the freetds.conf manpage "man freetds.conf".

# Global settings are overridden by those in a database
# server specific section
[global]
    # TDS protocol version
    tds version = auto

    # Whether to write a TDSDUMP file for diagnostic purposes
    # (setting this to /tmp is insecure on a multi-user
system)
;    dump file = /tmp/freetds.log
;    debug flags = 0xffff

    # Command and connection timeouts
;    timeout = 10
;    connect timeout = 10

    # To reduce data sent from server for BLOBs (like TEXT or
;    # IMAGE) try setting 'text size' to a reasonable limit
    # openssl ciphers = HIGH:!SSLv2:!aNULL:-DH
```

```
# A typical Sybase server
[egServer50]
    host = symachine.domain.com
    port = 5000
    tds version = 5.0

# A typical Microsoft server
[egServer73]
    host = ntmachine.domain.com
    port = 1433
    tds version = 7.3

# Satelium
[SatdataTDS]
    host            = 178.60.208.138
    port            = 1433
    database        = satdataweb
    timeout         = 600
    connect timeout = 15
```

8.2.2. Contenido de `/etc/odbc.ini`

```
Unset
[SatdataTDS]
Driver            = FreeTDS
Description       = SatdataTDS
Servername        = SatdataTDS
```

8.2.3. Contenido de `/etc/odbcinst.ini`

```
Unset
[FreeTDS]
```

```
Description="FreeTDS Driver"  
Driver=/usr/lib64/libtdsodbc.so  
UsageCount=1
```

9. Configuración del Usuario **admyG** y Permisos

Se procederá a la creación del usuario **admyG** y a la asignación de los permisos necesarios para su operación.

9.1. Creación del Usuario **admyG**

```
Unset  
sudo useradd admyG  
sudo passwd admyG # Se solicitará la introducción de una  
contraseña
```

9.2. Permisos en el Document Root del Aplicativo Web

El usuario **admyG** requiere permisos de lectura, escritura y ejecución sobre el directorio raíz del aplicativo web. La configuración se realiza como sigue:

```
Unset  
sudo chown -R admyG:apache /var/www/myg-tms.dach800.dachser.com/  
sudo chmod -R 775 /var/www/myg-tms.dach800.dachser.com/
```

9.3. Permisos **sudo** para la Gestión de Servicios

Para que el usuario **admyG** pueda administrar los servicios de Apache y PHP-FPM, es necesario otorgarle permisos específicos en el archivo **sudoers**.

Edite el archivo `/etc/sudoers` (se recomienda encarecidamente utilizar `visudo` para evitar errores de sintaxis):

```
Unset
sudo visudo
```

Añada las siguientes líneas al final del archivo:

```
Unset
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl stop httpd
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl start httpd
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl restart httpd
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl reload httpd
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl stop
php*-php-fpm.service
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl start
php*-php-fpm.service
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl restart
php*-php-fpm.service
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl enable
php*-php-fpm.service
admyG ALL=(root) NOPASSWD: /usr/bin/systemctl disable
php*-php-fpm.service
```

Esto permitirá al usuario `admyG` ejecutar los comandos `systemctl` especificados con privilegios de `root` sin requerir una contraseña.

9.4. Permisos de Lectura en los Registros de Apache

Para que el usuario `admyG` pueda acceder a los registros de Apache, se le otorgarán permisos de lectura:

Unset

```
sudo setfacl -m u:admyG:r /var/log/httpd  
sudo setfacl -m u:admyG:r /var/log/httpd/*
```

10. Instalación de Composer y Librerías PHP

Esta sección aborda la instalación de Composer y las librerías PHP esenciales para el aplicativo.

10.1. Configuración del Proxy en `yum.conf`

En caso de que se utilice un servidor proxy para las descargas de paquetes, es imperativo configurar esta directiva en el archivo `/etc/yum.conf`:

Unset

```
[main]  
gpgcheck=1  
installonly_limit=3  
clean_requirements_on_remove=True  
best=True  
skip_if_unavailable=False  
exclude=  
proxy=http://lx-reposrv01p.fro.int:8080
```

10.2. Instalación de Composer

La instalación de Composer requiere la configuración temporal de variables de entorno de proxy para la sesión actual:

Unset

```
export http_proxy="http://lx-reposrv01p.fro.int:8080"  
export https_proxy="http://lx-reposrv01p.fro.int:8080"
```

```
# Descarga del instalador de Composer
php -r "copy('https://getcomposer.org/installer',
'composer-setup.php');"

# Verificación del instalador (opcional, pero recomendado)
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ffce03186d2349af556795c8ab449cd0b5cfd079f65733d9d7d5
48e6ab02a1c4229bb7fb3b06d9e449d4') { echo 'Installer verified'; }
else { echo 'Installer corrupt'; unlink('composer-setup.php'); }
echo PHP_EOL;"

# Instalación global de Composer
sudo php composer-setup.php --install-dir=/usr/local/bin
--filename=composer

# Eliminación del instalador
php -r "unlink('composer-setup.php');"

# Verificación de la instalación
composer --version
```

10.3. Instalación de Librerías PHP mediante Composer

Las librerías requeridas por la aplicación se instalarán utilizando Composer. Las librerías identificadas son:

- **laminas/laminas-escaper**: Proporciona funcionalidades para el escape seguro de HTML, atributos HTML, JavaScript, CSS y URLs.
- **phpoffice/phpword**: Una biblioteca PHP para la lectura y escritura de documentos de procesamiento de texto (OOXML, ODF, RTF, HTML, PDF).
- **sop/asn1**: Una biblioteca PHP para la codificación y decodificación X.690 ASN.1 DER.

Para proceder con la instalación, navegue al directorio raíz de su proyecto PHP (donde se encuentra el archivo **composer.json** de su aplicación) y ejecute:

Unset

```
cd /var/www/myg-tms.dach800.dachser.com/ # 0 la ruta de su
proyecto
composer install # Este comando instalará las dependencias
listadas en composer.json
# Si se requiere añadir una librería específica:
composer require laminas/laminas-escaper phpooffice/phpword
sop/asn1
```

11. Instalación de Open-VM-Tools

Este paquete es crucial para optimizar la interacción de la máquina virtual con el entorno de virtualización subyacente.

Paquete instalado actualmente:

Unset

```
open-vm-tools-12.4.0-2.el9.x86_64
```

Para su instalación (en caso de no estar presente):

Unset

```
sudo dnf install open-vm-tools -y
sudo systemctl enable vmtoolsd
sudo systemctl start vmtoolsd
```

12. Instalación de CrowdStrike

CrowdStrike es una solución de seguridad que se implementa mediante un paquete RPM proporcionado por el departamento de seguridad informática y un script que facilita su inclusión

en el tenant correspondiente. El paquete RPM (`falcon-sensor_RHEL9.rpm`) debe ser obtenido del equipo de seguridad.

Una vez que el archivo RPM se encuentre en el servidor, ejecute el siguiente script:

```
Unset
#!/bin/bash
yum install falcon-sensor_RHEL9.rpm -y # Asegúrese de que el RPM
esté en la misma carpeta o especifique la ruta completa
/opt/CrowdStrike/falconctl -s -f
--cid="BB79B68158C945B8B2FE547F816A1CA5-48"
--provisioning-token="F89D34CE"

systemctl start falcon-sensor

echo "#-----#"
echo "Falcon Sensor installed"
echo "#-----#"

systemctl status falcon-sensor.service
```

13. Instalación del Agente Nagios NRPE

Para permitir la monitorización del servidor por parte de Nagios, es indispensable instalar el agente NRPE.

```
Unset
sudo dnf install epel-release -y # Requerido si el repositorio
EPEL no está habilitado
sudo dnf install nrpe nagios-plugins-all -y
sudo systemctl enable nrpe
sudo systemctl start nrpe
```

Posterior a la instalación, será necesario configurar el archivo `/etc/nagios/nrpe.cfg` para autorizar las conexiones desde el servidor Nagios.

14. Script de Transferencia SFTP (`sftp_out_myG.sh`)

Se describe a continuación un script de utilidad para la transferencia de archivos mediante SFTP. Este script se localiza en `/var/www/scripts/sftp_out_myG.sh` y genera un archivo de registro en `/var/www/scripts/logs/pruebaSFTP.log`. La propiedad de este script recae en el usuario `admyG`.

14.1. Contenido del Script `sftp_out_myG.sh`

Unset

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

```
#####  
#####
```

```
#Variables del script
```

```
#=====
```

```
#. $CAP_MODSH/modenv.sh
```

```
#$1 nombre de fichero a enviar
```

```
#$2 directorio donde se encuentra el fichero
```

```
#$3 nombre fichero de log
```

```
#$4 ruta backup
```

```
#$5 ruta en el ftp
```

```
#$6 usuario
```

```
#$7 password
```

```
#$8 servidor(IP)
```

```
#$9 ruta backup en ftp destino
```

```
#$10 codigo cliente
```

```
#$11 para email
```

```
#$12 cc email
```

```
#$13 cco email
```

```
#$14 asunto email
```

```
#$15 fichero asociado
```

```
export DIA=`date +%d%m%Y`
```

```
export HORA=`date +%H%M`
```

```

export DIALOG=`date +%Y%m%d`

export ERROR="error"

FICHERO_MOVE=$1
DIR_FICH=$2
FICH_LOG=$3
DIR_FTP=$4
USU_FTP="$5"
PSW_FTP="$6"
DIR_IP=$7

#####
#####

echo "Inicio de la ejecución del script de transferencia `date` "
echo
"=====
===="
echo "Variables de FTP: $FICHERO_MOVE $DIR_FICH $FICH_LOG
$DIR_BCK $DIR_FTP $USU_FTP $PSW_FTP $DIR_IP $DIR_BKF $FICASO"
cd $DIR_FICH
echo "estoy en el siguiente directorio: $DIR_FICH"

#Se borran primero los ficheros vacios
#=====
echo "Se borran los ficheros vacios"

find * -type f -size 0c -name "*" ! -name ".*" -exec rm {} \;

#Se obtienen los ficheros a enviar
#=====
FICHEROS=`ls $DIR_FICH | grep $FICHERO_MOVE`
PUNTO=.

```

```

#Comprobamos que hay ficheros para enviar por ftp
#=====

        #if [ "$FICHEROS"="" ]
        #if test -f $DIR_FICH/$FICHERO_MOVE*
hayfich=`ls -A $DIR_FICH/*$FICHERO_MOVE* | wc -l`
        if [ "$hayfich" -gt "0" ]
        then

                echo "Se van a pasar el/los ficheros $FICHEROS al
directorio del cliente"

#Se obtienen los ficheros a recibir
#=====
expect -c > $FICH_LOG << EOF "
spawn sftp -v $USU_FTP@$DIR_IP
expect \"password:\"
send \"\$PSW_FTP\r\"
expect \"sftp>\"
send \"bye\r\"
interact "
EOF

        (grep "Authentication succeeded" $FICH_LOG)

        if [ $? = 0 ]
        then

                echo "TEST OK"

echo "Se comprueba directorio"
expect -c > $FICH_LOG << EOF "
spawn sftp -v $USU_FTP@$DIR_IP
expect \"password:\"
send \"\$PSW_FTP\r\"

```



```
expect \"sftp>\"
send \"cd $DIR_FTP\r\"
expect \"sftp>\"
send \"bye\r\"
interact \"
EOF
```

```
        (grep "Couldn't canonicalize: No such file or directory"
$FICH_LOG || grep "Couldn't stat remote file: No such file or
directory" $FICH_LOG)
```

```
        if [ $? = 0 ]
        then
cat $FICH_LOG
        echo "No existe ruta en FTP: $DIR_FTP en FTP $DIR_IP"

                exit 0
        else
```

```
#Comienza la ejecucion del ftp
```

```
#=====
```

```
echo "Primera opcion del if sin ruta backup, antes del for
FICHEROS"
```

```
        for i in $FICHEROS
        do
```

```
mv $i $PUNTO$i
echo "Se renombra $i con el nombre $PUNTO$i"
expect -c > $FICH_LOG << EOF "
spawn sftp -v $USU_FTP@$DIR_IP
expect \"password:\"
send \"$PSW_FTP\r\"
expect \"sftp>\"
send \"cd $DIR_FTP \r\"
expect \"sftp>\"
send \"put $PUNTO$i \r\"
```

```
expect \"sftp>\"
send \"rename $PUNTO$i $i \r\"
expect \"sftp>\"
send \"bye\r\"
interact \"
EOF
```

```
mv $PUNTO$i $i
```

```
#0587265 - Se controla si hay error a la hora de dejar el fichero
(grep "Connection reset by peer" $FICH_LOG)
```

```
    if [ $? = 0 ]
    then
cat $FICH_LOG
    echo "Not connected. Error a la hora de enviar el fichero $i
al FTP $DIR_IP"
```

```
FICH="'$i' "
```

```
    fi
#0587265 - Fin
```

```
done
```

```
    fi
fi
#Comprobamos que el login y la transmision se ha hecho
correctamente, si no, no se movera ningun fichero del directorio.
#=====
=====
echo "Comprobamos"
cat $FICH_LOG
(grep "Failure" $FICH_LOG || grep "Connection closed"
$FICH_LOG || grep "Connection reset by peer" $FICH_LOG )
else
```

```

            echo "El directorio $DIR_FICH no contiene
ficheros para mandar por ftp"
        fi
    echo "Fin comprobacion"

    echo
    "=====
=====
echo "Finalizacion de la ejecución del script de transferencia
`date` "
    echo
    "=====
=====

```

14.2. Permisos del Script

Asegúrese de que el script posea permisos de ejecución y que **admyG** sea su propietario:

```

Unset
sudo chmod +x /var/www/scripts/sftp_out_myG.sh
sudo chown admyG:admyG /var/www/scripts/sftp_out_myG.sh

```

14.3. Directorio de Registros del Script

Cree el directorio destinado a los registros y asegúrese de que **admyG** disponga de permisos de escritura en él:

```

Unset
sudo mkdir -p /var/www/scripts/logs
sudo chown admyG:admyG /var/www/scripts/logs

```