

# Guía de Implementación - Validador DEIS SSO 2025 Mejorado

## Estructura de Archivos Nueva

```
validador-deis/
|
|— assets/
|   |— logo.png
|
|— config/
|   |— config.php      ★ NUEVO
|
|— includes/
|   |— functions.php   ★ NUEVO
|
|— Datos/              (archivos subidos)
|   |— .htaccess       ★ NUEVO
|
|— logs/               ★ NUEVO
|   |— .htaccess       ★ NUEVO
|
|— vendor/             (Composer)
|
|— index.php           ★ MEJORADO
|— upload.php          ★ MEJORADO
|— validar_excel.php   ★ MEJORADO
|— resultado_validacion.php ★ NUEVO (separado)
|— error_establecimiento.php ★ NUEVO (separado)
|— cleanup.php         ★ NUEVO
|— rules.json
|— establecimientos.txt
|— composer.json
|— .htaccess           ★ MEJORADO
```

## Pasos de Implementación

### 1. Backup del Sistema Actual

```
bash

# Crear backup completo
tar -czf backup-validador-$(date +%Y%m%d).tar.gz /ruta/al/validador/
```

## 2. Crear Nuevos Directorios

```
bash

cd /ruta/al/validador
mkdir -p config includes logs
chmod 755 config includes logs
chmod 777 Datos logs # Permisos de escritura
```

## 3. Archivos .htaccess de Protección

### Datos/.htaccess:

```
apache

# Denegar acceso directo a archivos subidos
Order Deny,Allow
Deny from all
```

### logs/.htaccess:

```
apache

# Proteger logs
Order Deny,Allow
Deny from all
```

## 4. Actualizar Archivos PHP

Reemplazar los archivos con las versiones mejoradas:

1. ☒ `index.php` - Con drag & drop
2. ☒ `upload.php` - Con validaciones de seguridad
3. ☒ `validar_excel.php` - Con mejor manejo de errores
4. ☒ Crear `resultado_validacion.php` - Vista separada
5. ☒ Crear `error_establecimiento.php` - Página de error
6. ☒ Crear `config/config.php` - Configuración central
7. ☒ Crear `includes/functions.php` - Funciones auxiliares
8. ☒ Crear `cleanup.php` - Mantenimiento

## 9. Actualizar `.htaccess` - Configuración Apache

### 5. Modificar Archivos Existentes

#### `upload.php`

Agregar al inicio:

```
php

<?php
define('APP_ACCESS', true);
require_once __DIR__ . '/config/config.php';
require_once __DIR__ . '/includes/functions.php';
```

#### `validar_excel.php`

Agregar al inicio:

```
php

<?php
define('APP_ACCESS', true);
require_once __DIR__ . '/config/config.php';
require_once __DIR__ . '/includes/functions.php';
```

### 6. Configurar Token de Seguridad

Editar `cleanup.php` y cambiar:

```
php

$expectedToken = 'TU_TOKEN_SECRETO_AQUI';
```

Por un token seguro generado:

```
bash

php -r "echo bin2hex(random_bytes(32));"
```

### 7. Configurar Cron para Limpieza

```
bash
```

```
# Editar crontab
```

```
crontab -e
```

```
# Agregar línea (ejecutar diariamente a las 2 AM)
```

```
0 2 * * * /usr/bin/php /ruta/completa/al/validador/cleanup.php >> /ruta/logs/cron.log 2>&1
```

```
# O con token desde web (menos recomendado)
```

```
0 2 * * * curl -s "https://tudominio.cl/validador/cleanup.php?token=TU_TOKEN_AQUI"
```

## 8. Verificar Permisos

```
bash
```

```
# Permisos correctos
```

```
chmod 644 *.php config/*.php includes/*.php
```

```
chmod 755 Datos logs
```

```
chmod 644 .htaccess
```

```
chmod 644 rules.json establecimientos.txt
```

```
# Usuario web debe ser dueño
```

```
chown -R www-data:www-data /ruta/al/validador/
```

```
# O según tu servidor
```

```
# chown -R apache:apache /ruta/al/validador/
```

## 9. Configurar PHP.ini

Verificar/ajustar en `php.ini` o `.user.ini`:

```
ini
```

```
upload_max_filesize = 50M
```

```
post_max_size = 55M
```

```
max_execution_time = 300
```

```
max_input_time = 300
```

```
memory_limit = 512M
```

```
file_uploads = On
```

## 10. Configurar .htaccess

Editar `.htaccess` y reemplazar `tudominio.cl` con tu dominio real.

## Lista de Verificación Post-Instalación

### Funcionalidad Básica

- ☐ Página principal carga correctamente
- ☐ Se puede seleccionar un archivo
- ☐ Drag & drop funciona
- ☐ Archivo se sube correctamente
- ☐ Validación se ejecuta sin errores
- ☐ Se muestran los resultados
- ☐ Exportación a Excel funciona
- ☐ Botón "Volver" funciona

### Seguridad

- ☐ Archivos .json no son accesibles directamente
- ☐ Directorio Datos/ no muestra listado
- ☐ Directorio logs/ no muestra listado
- ☐ Solo se permiten archivos .xlsm/.xlsx
- ☐ Validación MIME funciona
- ☐ Headers de seguridad están activos
- ☐ HTTPS está configurado (producción)

### Mantenimiento

- ☐ Logs se están generando correctamente
- ☐ Directorio logs/ es escribible
- ☐ Directorio Datos/ es escribible
- ☐ Script cleanup.php ejecuta correctamente
- ☐ Cron job está configurado
- ☐ Archivos antiguos se eliminan

### Rendimiento

- ☐ Archivos grandes (>10MB) se procesan
- ☐ No hay errores de timeout
- ☐ Memoria suficiente asignada
- ☐ Compresión gzip activa
- ☐ Cache de recursos estáticos funciona

## Error: "No se pudo crear el directorio"

```
bash

chmod 755 /ruta/al/validador
chown www-data:www-data /ruta/al/validador
```

## Error: "Permission denied" al subir archivo

```
bash

chmod 777 Datos
# O mejor, ajustar el usuario
chown www-data:www-data Datos
chmod 755 Datos
```

## Error: "Memory exhausted"

Aumentar en `php.ini`:

```
ini

memory_limit = 1024M
```

## Error: "Maximum execution time exceeded"

Aumentar en `php.ini`:

```
ini

max_execution_time = 600
```

## Los logs no se generan

```
bash
```

```
# Verificar permisos
```

```
chmod 755 logs
```

```
chown www-data:www-data logs
```

```
# Verificar en PHP
```

```
ini_set('log_errors', 1);
```

```
ini_set('error_log', '/ruta/completa/logs/php-errors.log');
```

## Archivos no se limpian automáticamente

```
bash
```

```
# Verificar cron
```

```
crontab -l
```

```
# Probar manualmente
```

```
php cleanup.php
```

```
# Verificar permisos
```

```
chmod 644 cleanup.php
```

## Monitoreo

### Ver logs en tiempo real

```
bash
```

```
tail -f logs/app-$(date +%Y-%m-%d).log
```

```
tail -f logs/cleanup-$(date +%Y-%m-%d).log
```

### Verificar espacio en disco

```
bash
```

```
df -h /ruta/al/validador
```

```
du -sh Datos/
```

```
du -sh logs/
```

### Contar archivos procesados

```
bash
```

## Actualización Futura

Para actualizar el sistema:

1. Crear backup
2. Descargar nuevas versiones
3. Reemplazar archivos PHP
4. Mantener `config.php` personalizado
5. Revisar `rules.json` por cambios
6. Probar en entorno de desarrollo primero
7. Desplegar en producción






## Soporte

En caso de problemas:

1. Revisar logs en `logs/app-YYYY-MM-DD.log`
2. Verificar permisos de archivos
3. Comprobar configuración PHP
4. Revisar `.htaccess` por conflictos
5. Verificar versión de PHP (mínimo 7.4)

## Métricas de Éxito

El sistema está funcionando correctamente si:

-  Validaciones se completan en < 30 segundos
-  No hay errores 500 en logs
-  Archivos antiguos se limpian automáticamente
-  Uso de disco se mantiene estable
-  Usuarios pueden subir y validar sin problemas



**¡Implementación completada!** 🎉

Tu sistema ahora cuenta con:

- ☒ Mejor seguridad
- ☒ Mejor rendimiento
- ☒ Mejor UX/UI
- ☒ Mantenimiento automático
- ☒ Código más limpio y mantenible