

Mejoras Implementadas en Validador DEIS SSO 2025



Resumen de Cambios

1. upload.php - Mejoras de Seguridad y Validación

Seguridad:

- ☒ Validación de método HTTP (solo POST)
- ☒ Validación de tipo MIME del archivo
- ☒ Protección contra sobrescritura con timestamp
- ☒ Sanitización de nombres de archivo con `basename()`
- ☒ Límite de tamaño de archivo (50MB)
- ☒ Validación de extensiones permitidas

Manejo de Errores:

- ☒ Mensajes de error descriptivos
- ☒ Códigos HTTP apropiados
- ☒ Validación de todos los posibles errores de carga

2. validar_excel.php - Mejoras de Rendimiento y Mantenibilidad

Optimización:

- ☒ Aumento de límite de memoria (512M)
- ☒ Timeout extendido (300 segundos)
- ☒ Separación de lógica y vista
- ☒ Manejo de errores con try-catch
- ☒ Logging de errores

Funcionalidad:

- ☒ Función `comparar()` con tolerancia para flotantes
- ☒ Mejor manejo de encodings (UTF-8)
- ☒ Función `generarExcel()` separada y reutilizable
- ☒ Validación de existencia de hojas antes de procesar
- ☒ Formato mejorado de valores numéricos (2 decimales)

Código más Limpio:

- ☒ Documentación con PHPDoc
- ☒ Constantes para configuración
- ☒ Separación de responsabilidades
- ☒ Uso de `match()` para casos (PHP 8+)

3. resultado_validacion.php - Nueva Vista Separada

Mejoras de UX:

- ☒ Tarjetas de resumen con estadísticas
- ☒ Contadores por tipo de severidad
- ☒ Iconos SVG integrados
- ☒ Colores diferenciados por fila según severidad
- ☒ Uso de <code> para expresiones
- ☒ Diseño responsive mejorado

Información:

- ☒ Muestra serie del archivo
- ☒ Resumen visual de incidencias
- ☒ Mensaje claro cuando no hay errores

4. index.php - Interfaz Mejorada

UX/UI:

- ☒ Drag & Drop funcional
- ☒ Vista previa de archivo seleccionado
- ☒ Indicador de tamaño de archivo
- ☒ Animaciones suaves
- ☒ Feedback visual en tiempo real
- ☒ Zona de drop con efectos hover

Funcionalidad:

- ☒ Validación de extensión en cliente
- ☒ Barra de progreso animada
- ☒ Información contextual sobre formato de archivo
- ☒ Botón con estado loading

5. error_establecimiento.php - Página de Error Dedicada

Mejoras:

- ☒ Diseño consistente con el resto del sistema
- ☒ Información clara del error
- ☒ Iconos y badges visuales
- ☒ Sugerencias para el usuario

Configuración Adicional Recomendada

PHP.ini (Servidor de Producción)



ini

```
; Límites de carga

upload_max_filesize = 50M
post_max_size = 55M
max_execution_time = 300
memory_limit = 512M


; Seguridad

file_uploads = On
allow_url_fopen = Off
allow_url_include = Off
expose_php = Off


; Errores (en producción)

display_errors = Off
log_errors = On
error_log = /var/log/php/error.log
```

.htaccess (Apache)



apache

Proteger archivos sensibles

<Files "*.json">

Order Allow,Deny

Deny from all

</Files>

<Files "*.txt">

Order Allow,Deny

Deny from all

</Files>

Proteger directorio de datos

<Directory "Datos">

Options -Indexes

Order Allow,Deny

Deny from all

</Directory>

Habilitar compresión

<IfModule mod_deflate.c>

AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css application/javascript

</IfModule>

Cache de recursos estáticos

<IfModule mod_expires.c>

ExpiresActive On

ExpiresByType image/png "access plus 1 year"

ExpiresByType text/css "access plus 1 month"

ExpiresByType application/javascript "access plus 1 month"

</IfModule>

Estructura de Archivos Recomendada



validador-deis/

```
|
|— assets/
|   └─ logo.png
|
|— Datos/      # Archivos subidos (proteger)
|   └─ .htaccess # Deny from all
|
|— vendor/      # Composer dependencies
|
|— config/      # Nuevo (recomendado)
|   └─ config.php  # Configuración centralizada
|   └─ establecimientos.php # Convertir .txt a .php
|
|— includes/    # Nuevo (recomendado)
|   └─ functions.php # Funciones reutilizables
|   └─ validation.php # Lógica de validación
|
|— logs/        # Nuevo (recomendado)
|   └─ .htaccess  # Deny from all
|
|— index.php
|— upload.php
|— validar_excel.php
|— resultado_validacion.php
|— error_establecimiento.php
|— rules.json
|— establecimientos.txt
|— composer.json
|— .htaccess
```

Próximas Mejoras Sugeridas

Corto Plazo:

1. **Sistema de logs** - Registrar todas las validaciones
2. **Base de datos** - Almacenar histórico de validaciones
3. **Autenticación** - Sistema de login por establecimiento
4. **API REST** - Permitir validaciones programáticas

Mediano Plazo:

- 1. **Dashboard** - Panel con estadísticas y gráficos
- 2. **Notificaciones** - Email cuando se detectan errores críticos
- 3. **Comparación** - Comparar archivos de diferentes meses
- 4. **Exportar PDF** - Además de Excel

Largo Plazo:

- 1. **Machine Learning** - Detectar patrones anómalos
- 2. **Validación en tiempo real** - WebSocket para progreso
- 3. **Multi-tenant** - Soporte para múltiples servicios de salud
- 4. **App móvil** - Versión para dispositivos móviles



Consideraciones de Seguridad

Importantes:

- 1. ☒ **Nunca** almacenar archivos en carpetas públicas sin validación
- 2. ☒ **Siempre** validar tipo MIME además de extensión
- 3. ☒ **Limitar** tamaño de archivos
- 4. ☒ **Sanitizar** nombres de archivos
- 5. ☒ **Usar** prepared statements si se implementa BD
- 6. ☒ **Implementar** CSRF tokens en formularios
- 7. ☒ **Habilitar** HTTPS en producción
- 8. ☒ **Rotar** logs periódicamente

Checklist de Producción:

- ☐ Configurar HTTPS/SSL
- ☐ Implementar rate limiting
- ☐ Configurar backups automáticos
- ☐ Monitoreo de errores (ej: Sentry)
- ☐ Implementar CSRF protection
- ☐ Configurar headers de seguridad
- ☐ Auditoría de dependencias (composer audit)
- ☐ Documentar procesos de deploy



Testing

Tests Recomendados:



php

// Ejemplo de test unitario para comparar()

```
function testComparar() {  
    assert(comparar(10, 10, '==') === true);  
    assert(comparar(10.0001, 10, '==') === true); // tolerancia  
    assert(comparar(5, 10, '<') === true);  
    assert(comparar(15, 10, '>') === true);  
}
```

// Test de carga de archivo

```
function testUpload() {  
    // Simular carga de archivo válido  
    // Verificar que se guarda correctamente  
    // Verificar que se rechaza archivo inválido  
}
```



Mantenimiento

Tareas Periódicas:

1. **Limpiar archivos antiguos** del directorio Datos/ (>30 días)
2. **Rotar logs** mensuales
3. **Actualizar dependencias** de Composer
4. **Revisar reglas** de validación
5. **Actualizar lista** de establecimientos

Script de Limpieza (cron):



bash

```
#!/bin/bash
```

```
# Ejecutar mensualmente: 0 0 1 * *
```

```
find /path/to/Datos -type f -mtime +30 -delete
```

```
find /path/to/logs -type f -mtime +90 -delete
```



Contribución

Para mantener la calidad del código:

1. Seguir PSR-12 para estilo de código PHP
2. Documentar funciones con PHPDoc
3. Probar cambios localmente antes de deploy

Versión: 2.0
Última actualización: Octubre 2025
Desarrollado para: DEIS Servicio Salud Osorno