



SISTEMA INTEGRADO DE GESTIÓN DE EXPEDIENTES MODULAR (SIGM)

FUNCIONALIDAD ANTIVIRUS
(MÓDULO TRAMITACIÓN ELECTRÓNICA)

SIGM v3



Administración Local Soluciones

Control de versiones

Versión	Fecha aprobación	Cambio producido	Autor
01	15-04-2009	Versión inicial.	IECISA
02	Octubre 2012	Versión actualizada para SIGM 3.0	IECISA

ÍNDICE

1	INTRODUCCIÓN	4
1.1	VISIÓN GENERAL DEL SISTEMA.....	4
1.2	FINALIDAD DEL DOCUMENTO	4
1.3	DEFINICIONES Y ABREVIATURAS.....	4
2	FUNCIONAMIENTO	5
3	CONFIGURACIÓN DE LA FUNCIONALIDAD ANTIVIRUS	6
4	CREACIÓN DE UN CONECTOR CON ANTIVIRUS.....	8

1 Introducción

1.1 Visión general del sistema

AL SIGM es la plataforma de Tramitación Electrónica del MINETUR, solución integral para la tramitación electrónica de los procedimientos administrativos, que fomenta la interoperabilidad entre administraciones mediante su adaptación a estándares de comunicación así como la reutilización de recursos e información pública.

1.2 Finalidad del documento

Este documento tiene por objeto explicar la funcionalidad de Antivirus ofrecida por el módulo de Registro Telemático para escanear los documentos entregados por un ciudadano durante un trámite electrónico

1.3 Definiciones y Abreviaturas

A continuación se expone una tabla con los diferentes acrónimos y abreviaturas utilizados a lo largo del documento, con su correspondiente definición.

Acrónimo / Abreviatura	Definición
MINETUR	Ministerio de Industria, Energía y Turismo
IECISA	Informática El Corte Inglés S.A.
SIGM	Sistema Integrado de Gestión de Expedientes Modular
AL	Administración Local

2 Funcionamiento

La funcionalidad de Antivirus permite que los documentos presentados por registro telemático sean chequeados a través de una clase conectora encargada de comunicarse con un software antivirus para detectar si un documento se encuentra infectado.

En caso de que dicha funcionalidad se encuentre activada cuando un trámite telemático es presentado, en el XML que contiene la información del registro se recoge como ha sido el resultado del chequeo del documento:

El documento no presenta infección:

```
<Documento>
....
  <Antivirus>OK</Antivirus>
</Documento>
```

El documento se encuentra infectado:

```
<Documento>
....
  <Antivirus>VIRUS</Antivirus>
</Documento>
```

Aquellos **documentos** en los que se detecte **infección no serán almacenados** en el sistema.

El funcionamiento anterior no es configurable, el sistema siempre trabajará del mismo modo ante la aparición de un documento con virus. Sin embargo es posible indicar si se debe de iniciar un expediente en el módulo de tramitación de expedientes aún cuando algún documento tenga virus. Esto es posible desde el fichero struts-config.xml

```
<set-property property="iniciarExpedienteConVirus" value="true" />
```

3 Configuración de la funcionalidad Antivirus

La clase conectora que se utiliza para comunicarse con el software antivirus se configura en el fichero SIGEM_spring.xml que se encuentra dentro del fichero sigem_core.jar utilizado por las aplicaciones de Registro Telemático:

- SIGEM_RegistroTelematicoWeb.war
- SIGEM_RegistroTelematicoWS.war

La configuración por defecto mantiene la funcionalidad antivirus desactivada:

```
<!-- ***** -->
<!-- ***** SERVICIO DE ANTIVIRUS ***** -->
<!-- ***** -->
<alias name="&ANTIVIRUS;.&SIGEM;.&API;"
alias="ANTIVIRUS_SERVICE_DEFAULT_IMPL"/>

<bean abstract="true" id="&ANTIVIRUS;"
class="ieci.tecdoc.sgm.core.services.antivirus.ServicioAntivirus">
</bean>

<bean id="&ANTIVIRUS;.&SIGEM;.&API;"
class="ieci.tecdoc.sgm.antivirus.NoAntivirus" parent="&ANTIVIRUS;" lazy-
init="true">
  <property name="rutaAntivirus" value=""/>
  <property name="parametros" value=""/>
  <property name="rutaTemporal" value=""/>
</bean>
```

La activación de la funcionalidad de antivirus supone la implementación de una clase conectora y la modificación del XML anterior.

Con la distribución de SIGEM se suministra un conector de ejemplo para comunicarse con el software McAfee bajo un entorno Windows.

A continuación se muestra la configuración necesaria para utilizar el conector implementado en la clase **ieci.tecdoc.sgm.antivirus.mcafee.AntivirusMcAfee**

```
<!-- ***** -->
<!-- ***** SERVICIO DE ANTIVIRUS ***** -->
<!-- ***** -->
<alias name="&ANTIVIRUS;.&SIGEM;.&API;"
alias="ANTIVIRUS_SERVICE_DEFAULT_IMPL"/>

<bean abstract="true" id="&ANTIVIRUS;"
class="ieci.tecdoc.sgm.core.services.antivirus.ServicioAntivirus">
</bean>

<bean id="&ANTIVIRUS;.&SIGEM;.&API;"
class="ieci.tecdoc.sgm.antivirus.mcafee.AntivirusMcAfee"
parent="&ANTIVIRUS;" lazy-init="true">
  <property name="rutaAntivirus" value="c:/Archivos de Programa/McAfee/VirusScan
Enterprise/scan32.exe"/>
  <property name="parametros" value="/UINONE /ARCHIVE"/>
  <property name="rutaTemporal" value="c:/Compartido/Antivirus Prueba"/>
</bean>
```

Los conectores tienen tres parámetros de configuración:

- rutaAntivirus: Ruta hacia el fichero ejecutable del software antivirus.
- parametros: Parámetros a pasar al fichero ejecutable para indicarle que debe escáner un fichero en línea de comandos.
- rutaTemporal: Carpeta temporal

4 Creación de un conector con Antivirus

La implementación de una clase conectora de antivirus debe extender la clase *ieci.tecdoc.sgm.antivirus.Antivirus* e implementar el interface *ieci.tecdoc.sgm.core.services.antivirus.ServicioAntivirus*.

Un ejemplo sería:

```
public class AntivirusMcAfee extends Antivirus implements ServicioAntivirus {
    protected boolean comprobarFichero(String rutaFichero) throws AntivirusException {
        File f = new File(rutaFichero);
        if (f.exists()) {
            f = null;
            //preparación de la linea de comandos
            cmd = new String[3];
            cmd[0] = getRutaAnivirus();
            cmd[1] = getParametros();
            cmd[2] = "\"" + rutaFichero + "\"";
            //ejecución de la linea de comandos:retorna 0 si la ejecución ha sido
correcta(tenga virus o no)
            int retorno = ejecucionAntivirus();
            switch (retorno) {
                case 0:
                    //si el fichero existe es q no lo movio o borro=> NO TIENE VIRUS
                    f = new File(rutaFichero);
                    if (f.exists())
                        return true;
                    else
                        return false;
                default:
                    throw new
AntivirusException(AntivirusException.EXC_EJECUCION_NO_VALIDA);
            }
        } else {
            throw new
AntivirusException(AntivirusException.EXC_FICHERO_NO_ENCONTRADO);
        }
    }
}
```

El método *comprobarFichero* deberá retornar un booleano con valor TRUE si el fichero no tiene virus y valor FALSE en caso contrario.

Durante la ejecución del método es posible lanzar una excepción *ieci.tecdoc.sgm.core.services.antivirus.AntivirusException*, en ese caso se considera



que el chequeo del fichero no ha podido realizarse y el sistema trabajará como si la funcionalidad de antivirus estuviese desactivada.