[INSTRUCTIVO]

Manual de implementación de servicios web de FirmaEC.

Versión 1.3

Elaborado por:
Alex John Camba Macas
Raquel Jenny Lojano Chavez

Revisado por:
Ing. Pablo Fernando Ordoñez Ordoñez Mg. Sc.

Enero 2022

Loja – Ecuador

HISTORIAL DE REVISIONES

Revisión	Fecha	Responsable	Descripción de la modificación
1.0	23/06/2021	Alex John Camba Macas	Versión inicial.
1.1	14/07/2021	Alex John Camba Macas	Se agregó los requisitos de hardware y software.
1.2	25/09/2021	Alex John Camba Macas	Se actualizó la versión de los siguientes componentes: • Postgresql. • Servidor de aplicaciones Wildfly.
1.3	07/01/2022	Raquel Jenny Lojano Chavez	Se aplica el formato institucional a todo el documento

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN

FirmaEC es un conjunto de aplicaciones que permiten firmar y verificar documentos electrónicos; también permite validar certificados digitales en archivo o token emitido por entidades certificadoras. Puede ser utilizado como un aplicativo de escritorio o como un componente web. El aplicativo FirmaEC web permite firmar documentos con extensión PDF desde un navegador web.

FirmaEC Web descentralizada

FirmaEC Web descentralizada está enfocado a utilizar los servicios web de firma electrónica en una infraestructura propia. El Módulo de Certificación Electrónica (MCE), genera un documento que requiere ser firmado, el cual es enviado a los servidores de FirmaEC instalados en la infraestructura propia, luego el usuario desde un navegador web inicia FirmaEC instalado en su escritorio y procede a firmar el documento. El documento firmado automáticamente es enviado en nuevo a los servidores de FirmaEC.

REQUISITOS

Se recomienda levantar una infraestructura con las siguientes características:

Requisitos de hardware

Tipo	Mínimo	Recomendado
Procesadores	2 núcleos de CPU	4 núcleos de CPU
Memoria (RAM)	2 GB	4 GB
Espacio de disco duro	10 GB	30 GB

Tabla 1. *Requisitos de Hardware*

Requisitos de software

Componente	Recomendado	Versión
Sistema Operativo	Centos	7, 8
Servidor de aplicaciones	Wildfly	20 o superior
Máquina Virtual de Java	OpenJDK	11
Base de Datos	PostgreSQL	12.6 y superior en la línea 11.x

Tabla 2. *Requisitos de Hardware*

Para que el servicio de firma funcione se requiere implementar los siguientes componentes:

Componente	Descripción	Link repositorio
firmadigital-api (3.001)	Este proyecto implementa los servicios REST para la aplicación de Firma Digital, la aplicación FirmaEC se comunicará con este API REST, la cual se comunicará con el Servicio Documentos (firmadigital-servicio).	https://github.com/Computacion-UNL/certificaciones/tree/main/firmadigital/firmadigital-api
firmadigital-servicio (3.0.0)	Es un servicio web para recibir documentos, para luego ser firmados del lado del cliente.	https://github.com/Computacion-UNL/certificaciones/tree/main/firmadigital/firmadigital-servicio
firmadigital-libreria (3.0.0)	Da soporte al proceso de firma y verificación de documentos, validación de certificados digitales, gestión de entidades certificadoras y administración de tokens.	https://github.com/Computacion-UNL/certificaciones/tree/main/firmadigital/firmadigital-libreria
certificacion-electronica-documentos (1.0.1)	El Servicio Documentos (firmadigital-servicio) devuelve el archivo firmado a este servicio web.	https://github.com/Computacion-UNL/certificaciones/tree/main/certificacion-electronica-documentos

Tabla 3. *Componentes de FirmaEC*

DESPLIEGUE

Configurar servidor de aplicaciones Wildfly

Paso 1: Para desplegar las aplicaciones se necesita un servidor de aplicaciones Java 11. Se debe descargar WildFly Application Server 20 (http://www.wildfly.org).

Paso 2: Crear un usuario para Wildfly.

```
sudo groupadd -r wildfly
sudo useradd -r -g wildfly -d  /opt/wildfly -s  /sbin/nologin wildfly
```

Imagen 1. *Crear usuario Wildfly*

Paso 3: Instalar WildFly

```
sudo wget https://download.jboss.org/wildfly/20.0.1.Final/wildfly-20.0.1.Final.tar.gz
sudo tar xvzf wildfly-20.0.1.Final.tar.gz -C /opt  &&
sudo ln -s /opt/wildfly-20.0.1.Final /opt/wildfly
```

Imagen 2. *Instalación Wildfly*

Asignamos como propietario al usuario wildfly a la caperta de /opt/wildfly:

```
sudo chown -RH wildfly:  /opt/wildfly
```

Imagen 3. *Asignación de propietario*

Paso 4: Configure system

```
sudo mkdir -p /etc/wildfly
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf   /etc/wildfly/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service
/etc/systemd/system/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh      /opt/wildfly/bin/
sudo sh -c 'chmod +x /opt/wildfly/bin/*.sh'
```

Imagen 4. *Configuración del sistema*

Paso 5: Inicie y habilite el servicio Wildfly

```
sudo systemctl daemon-reload
sudo systemctl start wildfly
sudo systemctl status wildfly
sudo systemctl enable wildfly
```

Imagen 5. *Inicio y habilitación de Wildfly*

Configuración de la base de datos con Postgresql 12

Paso 1: Se debe crear una base de datos llamada ***firmadigital*** con su respectivo usuario y contraseña.

```
sudo -i -u postgres
psql
# DROP DATABASE firmadigital;
CREATE USER firmadigital WITH PASSWORD 'firmadigital';
ALTER ROLE firmadigital WITH SUPERUSER;
CREATE DATABASE firmadigital WITH OWNER firmadigital;
GRANT ALL PRIVILEGES ON DATABASE firmadigital to firmadigital;
```

Imagen 6. *Creación de Base de Datos*

Paso 2: Se debe descargar el driver postgresql.jar para el servidor Wildfly:

```
wget https://jdbc.postgresql.org/download/postgresql-42.2.13.jar
```

Imagen 7. *Descarga de postgresql*

Paso 3: Luego, se debe ejecutar el servidor Wildfly y ejecutar lo siguiente:

```
/opt/wildfly*/bin/jboss-cli.sh -c
```

Imagen 8. *Ejecución del servidor Wildfly*

Después, se debe ejecutar los siguientes comandos:

```
module add --name=org.postgresql --
resources=/home/<user>/Descargas/postgresql-42.2.13.jar --
dependencies=javax.api,javax.transaction.api

/subsystem=datasources/jdbc-driver=postgresql:add(driver-
name=postgresql,driver-module-name=org.postgresql,driver-xa-datasource-class-
name=org.postgresql.xa.PGXADataSource)

data-source add --name=FirmaDigitalDS --jndi-name=java:/FirmaDigitalDS --driver-
name=postgresql  --connection-url=jdbc:postgresql://localhost:5432/firmadigital  --
user-name=firmadigital  --password=firmadigital  --valid-connection-checker-class-
name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnecti
onChecker                                    --exception-sorter-class-
name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSort
er
```

Imagen 9. *Ejecución de comandos*

Se crearán automáticamente las tablas necesarias en la base de datos “firmadigital” en caso de que no existan.

Paso 4: Posteriormente se necesita insertar un registro en la tabla "sistema" de dicha base de datos para registrar el servicio web al cual se devolverán los pdfs firmados:

```
sudo -i -u postgres
psql firmadigital
INSERT INTO sistema(id, url, apikey, apikeyrest, descripcion, nombre) VALUES (1,
'http://localhost:7776/recepcion/receiveDocument/saveSignedFile',
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx', 'xxxxxxxxxxxxxxxxxxxxx', 'Módulo de
certificación electrónica', 'mce');
```

Imagen 10. *Registro de servicio Web.*

Despliegue de los servicios web

Paso 1: Se debe descargar y compilar **firmadigital- librería** (Ver tabla 3) utilizando Apache Maven (http://maven.apache.org) con el siguiente comando:

```
mvn clean install
```

Imagen 11. *Compilación y empaquetamiento de firmadigital-libreria*

Paso 2: Se debe descargar y compilar **firmadigital-api, firmadigital-servicio** y **certificacion-electronica-documentos** (Ver tabla 3) utilizando Apache Maven con el siguiente comando:

```
mvn clean package
```

Imagen 12. *Compilación de firmadigital-api, firmadigital-servicio y certificacion-electrónica-documentos.*

Paso 3: Luego se debe copiar los archivos *.war* generados para los proyectos **firmadigital-api, firmadigital-servicio** y **certificacion-electronica-docs** en la carpeta */opt/wildfly*/standalone/deployments* del servidor de aplicaciones WildFly y ejecutar el comando:

```
/opt/wildfly*/bin/standalone.sh
```

Imagen 13*. Ejecución de comando de inicio*

Para probar rápidamente puede abrir este enlace (el número de puerto puede variar):
http://localhost:7776/api/fecha-hora

Instalación del aplicativo FirmaEC en el equipo del usuario final.

Para usar FirmaEC de escritorio, el usuario final debe descargar dicha aplicación desde el siguiente link: https://www.firmadigital.gob.ec/descargar-firmaec

Invocación de servicio de firma digital FirmaEC.

El proceso de firma digital se inicia desde el lado de un sistema requirente (por ejemplo, el MCE), que solicita se firme un documento, invocando al servicio web REST de firma previamente modificado, compilado y publicado en su infraestructura. Para ver mas detalles revise la imagen del siguiente enlace https://github.com/Computacion-UNL/certificaciones/blob/main/certificacion-

electronica-recursos/03_imagenes/memoria/Diagrama%20Secuencia%20-%20FirmaEC.png

Ejecución de la aplicación del lado del cliente.

Para esto se utiliza un protocolo especial denominado *firmaec://* mismo que será atendido por el software instalado en el equipo del usuario final; mediante un Protocol Handler en el sistema operativo, se configura para que todos los links en el navegador que inicien con el protocolo *firmaec://* sean abiertos con esta aplicación. Esto permite ejecutar lógica en forma externa al navegador web, pero invocada por un link desde la interfaz del Módulo de Certificación Electrónica.
Luego de que el usuario de clic en el link, la aplicación del lado del cliente solicita el certificado PKCS#12 en archivo o un token de firma digital y se procede con la firma.

Respuesta de FirmaEC al sistema requirente.

Una vez FirmaEC recibe el documento, este invoca automáticamente al servicio web REST denominado ***certificacion-electronica-docs*** registrado por el sistema requirente enviando de vuelta el documento firmado digitalmente. Dicho servicio web debe recibir la siguiente información por medio de una acción llamada por ejemplo "grabaArchivoFirmado" la cual está asociado a los siguientes parámetros:

Parámetro	Tipo	Descripción
cedula	String	Cédula de identidad del usuario
nombreDocumento	String	Nombre del documento firmado
archivo	base64Binary	Contenido del documento firmado, en formato Base 64.
nombreApellidoFirmante	String	Nombres y Apellidos del firmante

Tabla 4. *Parámetros de "grabaArchivoFirmado"*

Respuesta

Se envía **OK** si se recibió el documento y **ERROR** en caso de que, no se haya recibido el documento.

API REST

Definición de API REST

El servicio web del proyecto firmadigital-servicio contiene principalmente los siguientes endpoints:

Crear Documentos
POST http://localhost:7776/servicio/documentos

Descripción: Crea un documento para ser firmado por la aplicación del lado del cliente.

Tipo Parámetro	Parámetro	Tipo	Descripción
Header	X-API-KEY	String	API Key asignada al Módulo de Certificación Electrónica (no SHA256)
Body	sistema	String	Sistema al que pertenece el documento
	cedula	String	Cédula de identidad del usuario
	documentos		Array de documentos, con parámetros "nombre" y "documento"
	nombre	String	Nombre del documento a firmar
	documento	String	Contenido del documento a firmar, en formato Base 64.

Tabla 5. *Parámetros de "firmadigital-servicio"*

Respuestas esperadas

HTTP Code	Descripción
201	Operación exitosa
400	Error al decodificar JSON, no se cumple el schema.
403	Error al validar el API Key

Tabla 6*. Código de respuestas esperadas.*

Produces

application/text

Seguridad

Se debe incluir como Header el parámetro X-API-KEY con un API KEY provisto para el Módulo de Certificación Electrónica.

Aplicación del lado del cliente

Esta aplicación es invocada mediante un link desde la página web del Módulo de Certificación Electrónica, que tiene un protocolo especial, registrado en el sistema operativo del usuario. Este protocolo se presenta así:

```
<a href="firmaec://mce/firmar?token=XXXXX&tipo_certificado=2">Firmar</a>
```

Imagen 14. *Protocolo del Módulo de Certificación Electrónica.*

Este link debe ser construido del lado del servidor, y presentado en la interfaz de la aplicación web.

Especificación de parámetros.

Los parámetros se encuentran detallados a continuación:

Parámetro	Tipo	Obligatorio	Descripción
token	String	Si	JWT devuelto por el servicio web de FirmaEC
tipo_certificado	Integer	Si	**1** = Token **2** = Archivo **3** = Tarjeta inteligente (cédula)
llx	Integer	Si	En formato A4 Posición X (positionOnPageLowerLeftX)
lly	Integer	Si	En formato A4 Posición Y (positionOnPageLowerLeftY)
estampado	String	Si	**QR** = Información estampada en QR **information1** = Información personalizada **information2** = Información con estándar ETSI TS 102 778-6 V1.1.1
url	String	No	Hace referencia a la url del api rest de firma

Tabla 7. *Especificación de parámetros*

GLOSARIO DE TERMINOS

Término	Descripción
Centos	Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito.
Wildfly	Conocido antes como JBoss As, o solo JBoss, es un servidor Open Source de aplicaciones Java EE. Es útil para crear, implementar y hospedar aplicaciones y servicios Java. Además, maneja servlets, JSP, EJB y JMS.
OpenJDK	Es la versión libre de la plataforma de desarrollo Java bajo concepto de lenguaje orientado a objetos.
PostgreSQL	Es un sistema para gestionar bases de datos de muy alto nivel, completamente de software libre y con una licencia BSD, compatible con cualquier uso, ya sea personal o comercial.
Protocol Handler	Los controladores de protocolos basados en web permiten a las aplicaciones basadas en web participar en el proceso también.
REST	Define un conjunto de principios arquitectónicos por los que se pueden diseñar servicios Web que se centran en los recursos de un sistema, lo que incluye la forma en que los estados de los recursos se dirigen y transfieren a través de HTTP por un amplio rango de clientes que están escritos en diferentes lenguajes.
API KEY	Es un identificador que sirve como el medio de autenticación de un usuario para el uso de los servicios proporcionados por Reachcore, en los Web Services o en el API Rest.

***Tabla 8**. Glosario de términos*

Acción	Estudiante	Firma
Elaborado	Alex John Camba Macas	
	Raquel Jenny Lojano Chavez