@Firma :: Componente Miniapplet de firma electrónica

Dirección General de Política Digital

Consejería de Hacienda y Administración Pública

Sevilla, 24 de Junio de 2016

ÍNDICE

Introducción

Actividad reciente (1)

02/10/2015 Sesión técnica de formación @firma y HCV. Guía de migración hacia la herramienta centralizada de verificación desde sistemas previos.

20/11/2015 Sesión técnica sobre la nueva versión de la herramienta Portafirmas 3.0

28/12/2015 Visor de documentos y expedientes electrónicos ENI. *En la Herramienta Centralizada de Verificación se ha implantado una nueva funcionalidad de visor de documentos y expedientes electrónicos conformados de acuerdo con la Norma Técnica de Interoperabilidad de Documento Electrónico y la Norma Técnica de Interoperabilidad de Expediente Electrónico*.

Introducción

Actividad reciente (2)

20/01/2016 Nueva versión del componente afirma-enidocs-ws. *Se ha actualizado la versión del componente "afirma-enidocs-ws" a la versión 1.11. Este componente permite la comunicación entre la Herramienta Centralizada de Validación y la base de datos de custodia de @firma 5 (implantaciones locales)*.

25/01/2016 Portafirmas documento de definición de requisitos. *En el apartado correspondiente a la herramienta Portafirmas se ha publicado el documento de definición de requisitos de la próxima versión (3.1)*

03/02/2016 Sesión técnica de formación NTIs de documento y expediente electrónico y la adecuación a las mismas.

Introducción

Actividad reciente (3)

08/03/2016 Versión 3.0.1 de Port@firmas

05/04/2016 Sesión técnica formación nueva versión Miniapplet, Autofirma.

06/04/2016 Presentación de la herramienta de validación y generación de expedientes ENI en la reunión de jefes y responsables de servicios y unidades TIC de la Junta de Andalucía

11/04/2016 Publicación de la versión 1.4 de Miniapplet y Autofirma

16/06/2016 Publicación de la versión 1.4JAv02 de Miniapplet y Autofirma

Introducción

Ley 18/2011, de 5 de julio, reguladora del uso de las tecnologías de la información y la comunicación en la Administración de Justicia

Artículo 37.4: "Los expedientes y demás actuaciones que deban ser remitidos por otras Administraciones y organismos públicos deberán realizarse en todo caso por vía telemática a través de la correspondiente sede judicial electrónica. El expediente administrativo electrónico habrá de cumplir los requisitos previstos en la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos, y deberá remitirse debidamente foliado mediante un índice electrónico que permita la debida localización y consulta de los documentos incorporados".

1 de enero de 2017

Introducción

Situación objetivo de las aplicaciones (1)

Utilizar la implantación corporativa de la plataforma @firma.

Cumplir las guías y directrices especificadas en el apartado correspondiente a la plataforma en la web de soporte de administración electrónica.

Gestionar los metadatos mínimos obligatorios de las NTIs de documento y expediente electrónico.

Introducción

Situación objetivo de las aplicaciones (2)

Para la práctica de la verificación, mediante un código generado electrónicamente, de documentos firmados electrónicamente, utilizar la Herramienta Centralizada de Verificación (aportando a la misma también la información sobre los metadatos mínimos obligatorios de los documentos electrónicos).

Introducción

Situación objetivo de las aplicaciones (3)

Utilizar herramienta Portafirmas en su última versión (actualmente 3.0.1) en modalidad de creación de peticiones de firma por referencia.

Capacidad de exportar documentos electrónicos y expedientes electrónicos conformes a las NTIs.

Miniapplet de firma 1.4.JAv02

- ¿Qué es el miniapplet?
- Diferencias con el Cliente de firma
- Prestaciones / funcionalidad
- Comparativa con la versión liberada por MINHAP
- ¿Cómo se integra?
- Diferencia con versiones anteriores
- Matriz de compatibilidad escritorio (APPLET)

Miniapplet de firma 1.4.JAv02

¿Qué es el Miniapplet de firma?

El MiniApplet @firma es una herramienta de firma electrónica que funciona en forma de applet de Java integrado en una página Web mediante JavaScript.

- Muy similar al **Cliente de firma** tradicional (puede considerarse un cliente de firma “ligero”).
- De fácil integración (requiere menos dependencias).
- Compatible con **firma móvil (Android, iOS) y AutoFirma.**
- Compatible con la firma por protocolo en navegadores que no soportan Java.
- **Centralizado**: Mínimos cambios en las aplicaciones para adoptar una nueva versión. La integración centralizada es opcional, se sigue distribuyendo como el cliente de firma.
- Cliente más ligero y de carga más rápida.

Miniapplet de firma 1.4.JAv02

Diferencias con el Cliente de firma

	Cliente Firma	Miniapplet
Recursos	Cliente de firma completo, solicitado a través de las **descargas privadas**.	Instanciados del repositorio central: - miniapplet.js - miniapplet-full.jar (no se referencia directamente) Se da la opción de ser descargado.
API	139 métodos	14 métodos
Formatos de firma	Catálogo **completo**	Catálogo reducido: **CAdES**, **XAdES**, **PAdES**, **FacturaE**
Soporte firma móvil	NO	SÍ (Servicios de firma móvil centralizados)
Compatible con Autofirma	NO	Sí (a partir de la versión 1.3)

Miniapplet de firma 1.4.JAv02

Prestaciones / funcionalidad

- El MiniApplet @firma proporciona **únicamente** funcionalidades de **firma electrónica** (incluyendo multifirmas) sobre un conjunto determinado de formatos de firma
- Incluye un conjunto muy reducido de métodos auxiliares
- Compatibilidad integrada con DNIe, incluyendo el driver Java, lo cual permite no tener que instalar el driver del DNI Electrónico (sólo para entornos de escritorio)
- No permite sobres digitales o cifrados simétricos

Miniapplet de firma 1.4.JAv02

Prestaciones / funcionalidad

- Formatos de firma soportados

Formatos	Variantes	Extras
CAdES	BES y EPES	Implícita y Explícita
XAdES	BES y EPES	Enveloped, Enveloping, Externally Detached e Internally Detached
PAdES	BES	No admite ficheros adjuntos o empotrados en el PDF
FacturaE		Versión 3.1 del estándar

(*) Sobre todos estos formatos de Firma hay restricciones y características propias que pueden consultarse en el Manual del Integrador.

Miniapplet de firma 1.4.JAv02

Prestaciones / funcionalidad

- Formatos de firma No soportados

Formatos	Recomendaciones
CMS / PKCS#7	Utilizar CAdES
XMLDSig	Utilizar XAdES
ODF	No utilizar
OOXML	No utilizar

- Eliminados los formatos menos usados (ODF, OOXML) y no recomendados (CMS, XMLDSig).

(*) Si se necesita algún formato no soportado deberá utilizar el **Cliente de firma** tradicional.

Miniapplet de firma 1.4.JAv02

Prestaciones / funcionalidad

- Mecanismo automático de detección de almacenes de certificados (según el navegador y el S.O)

Navegador	S.O.	Almacén de certificados
Firefox	Cualquiera	NSS + PKCS#11 + SmartCard + DNIe + HSM , etc.
Otros (Internet Explorer, Opera, Chrome, Safari)	Windows	CAPI (Microsoft CryptoAPI)
	Mac OS X	Llavero de Mac OS X
	GNU/Linux	NSS (Network Security Services)

- Ejemplos de carga del miniapplet con el almacén por defecto:

```
...
<script type="text/javascript">
MiniApplet.cargarMiniApplet(codeBase);
</script>
...
```

Miniapplet de firma 1.4.JAv02

Prestaciones / funcionalidad

- Posibilidad de forzar manualmente el uso de un almacén determinado

Formados de almacenes soportados	Parámetro
PKCS#12 / PFX	KEYSTORE_PKCS12
PKCS#11	KEYSTORE_PKCS11
CAPI (Microsoft CryptoAPI)	KEYSTORE_WINDOWS
Llavero de Mac OS X	KEYSTORE_APPLE
NSS (Network Security Services)	KEYSTORE_FIREFOX

- Ejemplos de carga del miniapplet con un almacén determinado:

```
…
<script type="text/javascript">
MiniApplet.cargarMiniApplet(codeBase, KEYSTORE_WINDOWS);
MiniApplet.cargarMiniApplet(codeBase, KEYSTORE_FIREFOX);
</script>
…
```

Miniapplet de firma 1.4.JAv02

Comparativa con la versión liberada por MINHAP

- **Miniapplet 1.4JAv02 vs Miniapplet 1.4 (MINHAP)**

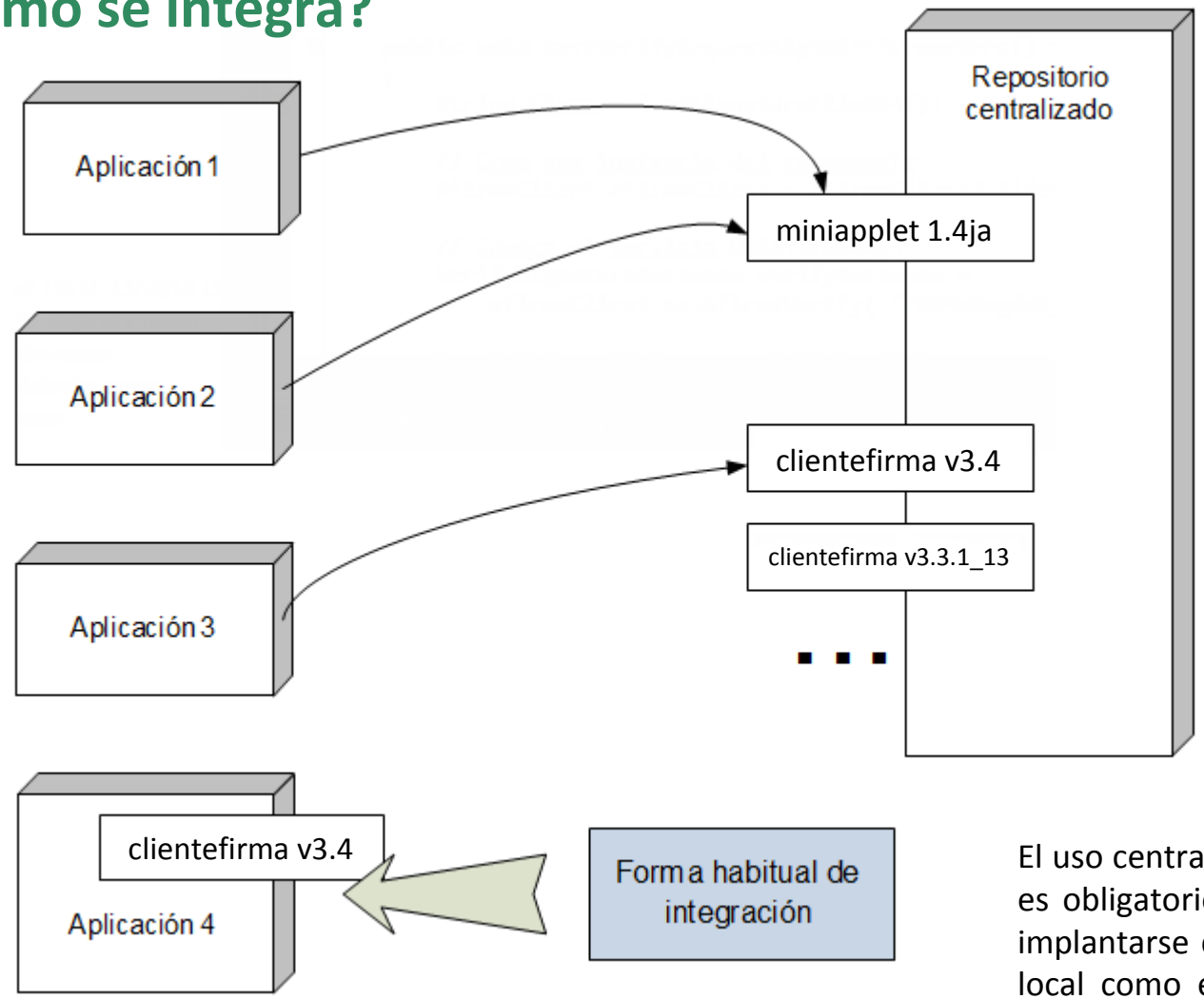
- Se ha creado una nueva rama de desarrollo de miniapplet
- La versión 1.4JAv02 es una versión mejorada de la versión liberada por MINHAP y la segunda revisión de la Junta de Andalucía

- Cambios realizados sobre la versión 1.4 de MINHAP

  - Se corrige un BUG que impedía operaciones de cofirma y contrafirma sobre firmas electrónicas con sello de tiempo (firmas CAdES).
  - Se incluye el atributo firmado ContentHint en las contrafirmas AdES-EPES, necesario para su validación contra @firma.
  - Se ha firmado el componente con un certificado de la Junta de Andalucía.

Miniapplet de firma 1.4.JAv02

¿Cómo se integra?



El uso centralizado no es obligatorio, podría implantarse de forma local como el cliente de firma

Miniapplet de firma 1.4.JAv02

¿Cómo se integra?

```
<html>
  <head>

    ...

    <script type="text/javascript"
src="https://ws024.juntadeandalucia.es/afirma-validator-miniapplet-1_4/miniapplet.js"></script>

    ...

  </head>

  ...

</html>
```

Miniapplet de firma 1.4.JAv02

¿Cómo se integra?

- Carga del miniapplet

```
...

<script type="text/javascript">
MiniApplet.cargarMiniApplet("https://ws024.juntadeandalucia.es/afirma-validator-miniapplet-1_4/");
</script>

...
```

NOTA: *La carga del miniapplet es asíncrona. Es tarea del integrador asegurar la carga del componente antes de la invocación de los métodos de firma electrónica.*

Miniapplet de firma 1.4.JAv02

¿Cómo se integra?

- Ejemplo de firma, cofirma y contrafirma

```
...
var params = "format=XAdES Detached\nmode=implicit";

function firmaExito(signatureB64){
    MiniApplet.saveDataToFile(signatureB64, "Guardar firma", null, null, null);
}
function firmaError(errorType, errorMessage){
    alert("Type: " + errorType + "\nMessage: " + errorMessage);
}

//Firma
MiniApplet.sign(data, "SHA1withRSA", "XADES", params, firmaExito, firmaError);
// Cofirma
MiniApplet.coSign(signature, data, "SHA1withRSA", "XADES", params, firmaExito, firmaError);
// Contrafirma
MiniApplet.counterSign(signature, "SHA1withRSA", "XADES", params, firmaExito, firmaError);
...
```

Miniapplet de firma 1.4.JAv02

Diferencia con versiones anteriores

	v1.4.JA v02	v1.4JA v01	v1.3	v1.2	v1.1.5	v1.1.4
Compatible con firma móvil	Sí	Sí	Sí	Sí (*)	Sí (*)	Sí (*)
Compatibilidad con Autofirma	Sí	Sí	Sí	No	No	No
Requiere servidor intermedio (autofirma)	No	No	Sí	N/A	N/A	N/A

NOTA (*): *La últimas versiones del cliente móvil de firma son incompatibles con versiones del miniapplet anteriores a la versión v1.3.*

Miniapplet de firma 1.4.JAV02

Matriz de compatibilidad escritorio (APPLET)

	JRE 1.6 (1.6.0_45)						
	Firefox	I. Explorer				Chrome	Safari
S.O.	46	8	9	10	11	51	9
Windows XP SP3	NOK (3)	OK	N/S	N/S	N/S	N/S (1)	N/S
Windows 7	NOK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 8.1	NOK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 10	NOK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Ubuntu 14.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Ubuntu 16.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Guadalinex v9	N/D	N/S	N/S	N/S	N/S	N/S (2)	N/S
Mac OS X 10.11 El Capitán	N/S (4)	N/S	N/S	N/S	N/S	N/S (1)	N/S (4)

(1) Desde Chrome 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://www.java.com/es/download/faq/chrome.xml

(2) Desde Chromium 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://code.google.com/p/chromium/issues/detail?id=375909

(3) La consola Java devuelve el siguiente error al firmar con las últimas versiones de Mozilla Firefox: **Se ha pedido alias a un almacén no inicializado.**

(4) No existe Java 6 oficial para la última versión de Mac OS X 10.11 El Capitán.

Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente.

Miniapplet de firma 1.4.JAv02

Matriz de compatibilidad escritorio (APPLET)

	JRE 1.7 (1.7.0_80)						
	Firefox	I. Explorer				Chrome	Safari
S.O.	46	8	9	10	11	51	9
Windows XP SP3	OK	OK	N/S	N/S	N/S	N/S (1)	N/S
Windows 7	OK	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 8.1	OK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 10	OK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Ubuntu 14.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Ubuntu 16.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Guadalinex v9	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Mac OS X 10.11 El Capitán	OK	N/S	N/S	N/S	N/S	N/S (1)	OK (4)

(1) Desde Chrome 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://www.java.com/es/download/faq/chrome.xml

(2) Desde Chromium 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://code.google.com/p/chromium/issues/detail?id=375909

(3) Se ha podido firmar con Firefox 46, pero se han detectado problemas de foco en la ventana de selección de certificados.

(4) Existe un bug reconocido en Safari al ejecutar módulos Java en modo seguro. Para poder firmar sin problemas es necesario configurar la ejecución de código Java desde Safari en modo no seguro para los sitios que cargan el miniapplet. (Safari->Preferencias->Seguridad->Ajustes de módulo (Java)->Ejecutar en modo no seguro (Sitio del miniapplet)). https://bugs.openjdk.java.net/browse/JDK-8148430

Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente.

Miniapplet de firma 1.4.JAv02

Matriz de compatibilidad escritorio (APPLET)

	JRE 1.8 (1.8.0_77)						
	Firefox	I. Explorer				Chrome	Safari
S.O.	46	8	9	10	11	51	9
Windows XP SP3	OK	OK	N/S	N/S	N/S	N/S (1)	N/S
Windows 7	OK	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 8.1	OK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Windows 10	OK (3)	N/D	N/D	N/D	OK	N/S (1)	N/S
Ubuntu 14.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Ubuntu 16.04	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Guadalinex v9	OK	N/D	N/D	N/D	N/D	N/S (2)	N/S
Mac OS X 10.11 El Capitán	OK	N/S	N/S	N/S	N/S	N/S (1)	OK (4)

(1) Desde Chrome 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://www.java.com/es/download/faq/chrome.xml

(2) Desde Chromium 43 en adelante se ha retirado el soporte al plugin de Java, por lo que no resulta posible la ejecución de applets en este navegador. Más información en el siguiente enlace: https://code.google.com/p/chromium/issues/detail?id=375909

(3) Se ha podido firmar con Firefox 46, pero se han detectado problemas de foco en la ventana de selección de certificados.

(4) Existe un bug reconocido en Safari al ejecutar módulos Java en modo seguro. Para poder firmar sin problemas es necesario configurar la ejecución de código Java desde Safari en modo no seguro para los sitios que cargan el miniapplet. (Safari->Preferencias->Seguridad->Ajustes de módulo (Java)->Ejecutar en modo no seguro (Sitio del miniapplet)). https://bugs.openjdk.java.net/browse/JDK-8148430

Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente.

Autofirma 1.4.2.JAv02

- ¿Qué es autofirma?
- Funcionalidades y limitaciones
- Requisitos de instalación
- Esquema de funcionamiento
- ¿Cómo se integra?
- Matriz de compatibilidad Autofirma

Autofirma 1.4.2.JAv02

¿Qué es autofirma?

- **¿Qué es Autofirma?**

  Es una **aplicación de escritorio**, **independiente del navegador web,** que permite a los usuarios la realización de firmas electrónicas **en navegadores que no soportan la ejecución de applets Java**.

- **¿Por qué es útil Autofirma? ¿Aporta algo nuevo?**
  - El 1 de septiembre de 2015 Google eliminó el soporte de complementos NPAPI en Google Chrome.
  - Oracle ha anunciado que dejará de dar soporte a la tecnología Java Applets en septiembre de 2016.
  - Mozilla Firefox ha anunciado el fin del soporte de complementos NPAPI a finales de 2016.

AUTOFIRMA es la única alternativa "independiente del navegador" para la realización de firmas electrónicas en entornos de escritorio.

Autofirma 1.4.2.JAv02

Funcionalidades y limitaciones

- **Funcionalidades principales de Autofirma**
  - Puede utilizarse como aplicación de escritorio para la firma local de ficheros (PadES, CadES, XadES)
  - Compatible con factura electrónica (FacturaE)
  - No requiere de servidor intermedio en entornos de escritorio (habilita la comunicación con el navegador a través de un SOCKET SSL local)
  - La nueva versión 1.4.2.JAv02 ya implementa la funcionalidad de firma masiva.
  - Está disponible para Windows, Linux y Mac OS X.

- **Limitaciones de la versión actual**
  - Java es un requisito para la ejecución de Autofirma, aunque en las versiones para Windows y Mac OS X ya incorpora una versión de JRE 1.8. En la versión de Linux será necesario instalar una JRE 1.8 compatible (Oracle u OpenJRE).
  - Requiere disponer de la aplicación previamente instalada y actualizada (Permisos de administración)

Autofirma 1.4.2.JAv02

Requisitos de instalación

- **Instalación de Autofirma**

  - Los instalables para los diferentes S.O. están disponibles en la siguiente URL:
    https://ws024.juntadeandalucia.es/clienteafirma/autofirma/autofirma.html

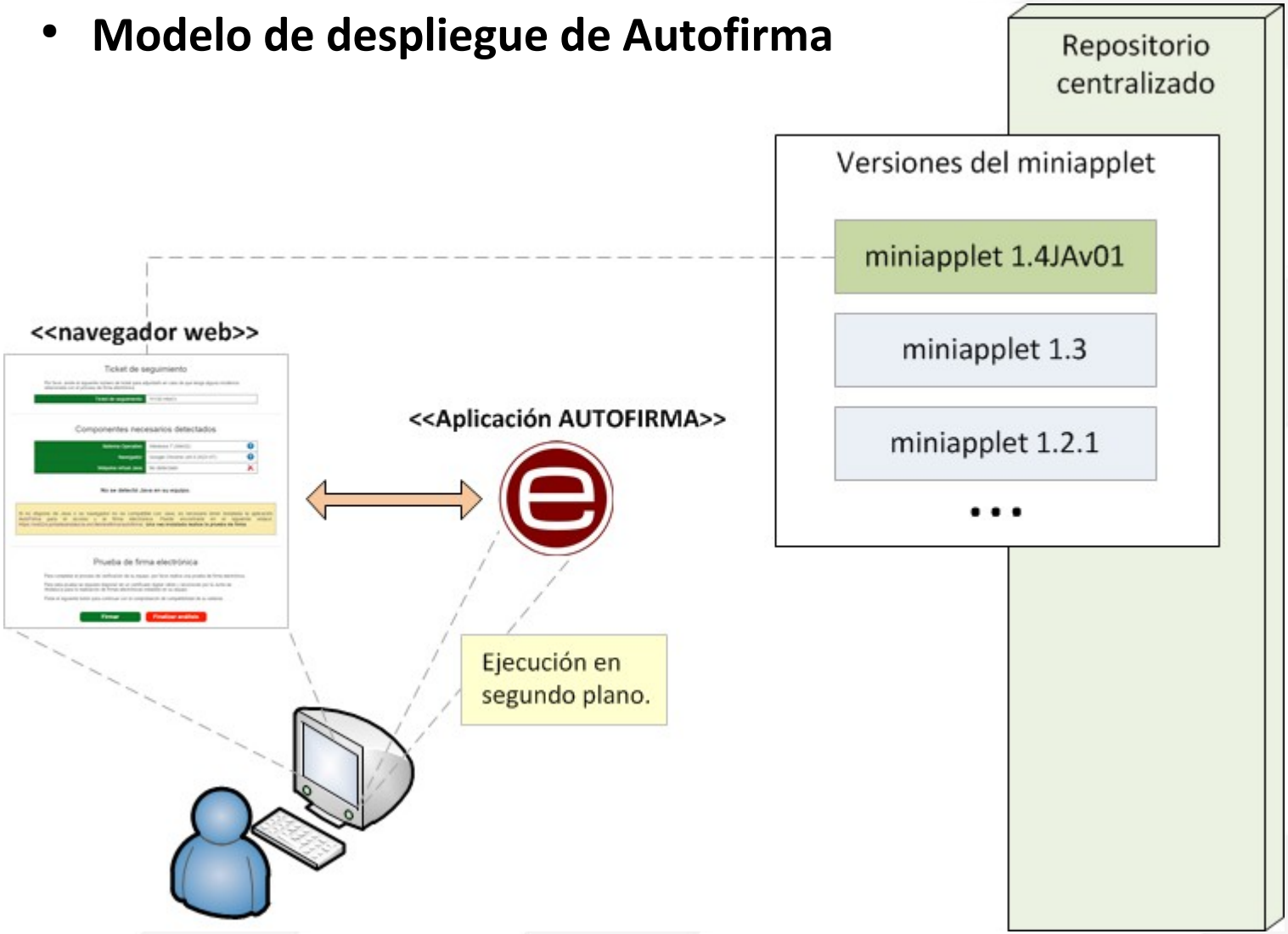
  - El instalador requiere que el usuario posea **permisos de administración** para poder instalar correctamente la aplicación y el certificado de confianza generado.

  - **Actualizaciones automáticas** (Autofirma puede detectar nuevas versiones en la URL anteriormente indicada e informar al usuario convenientemente). El proceso de actualización es manual.

Autofirma 1.4.2.JAv02

Esquema de funcionamiento

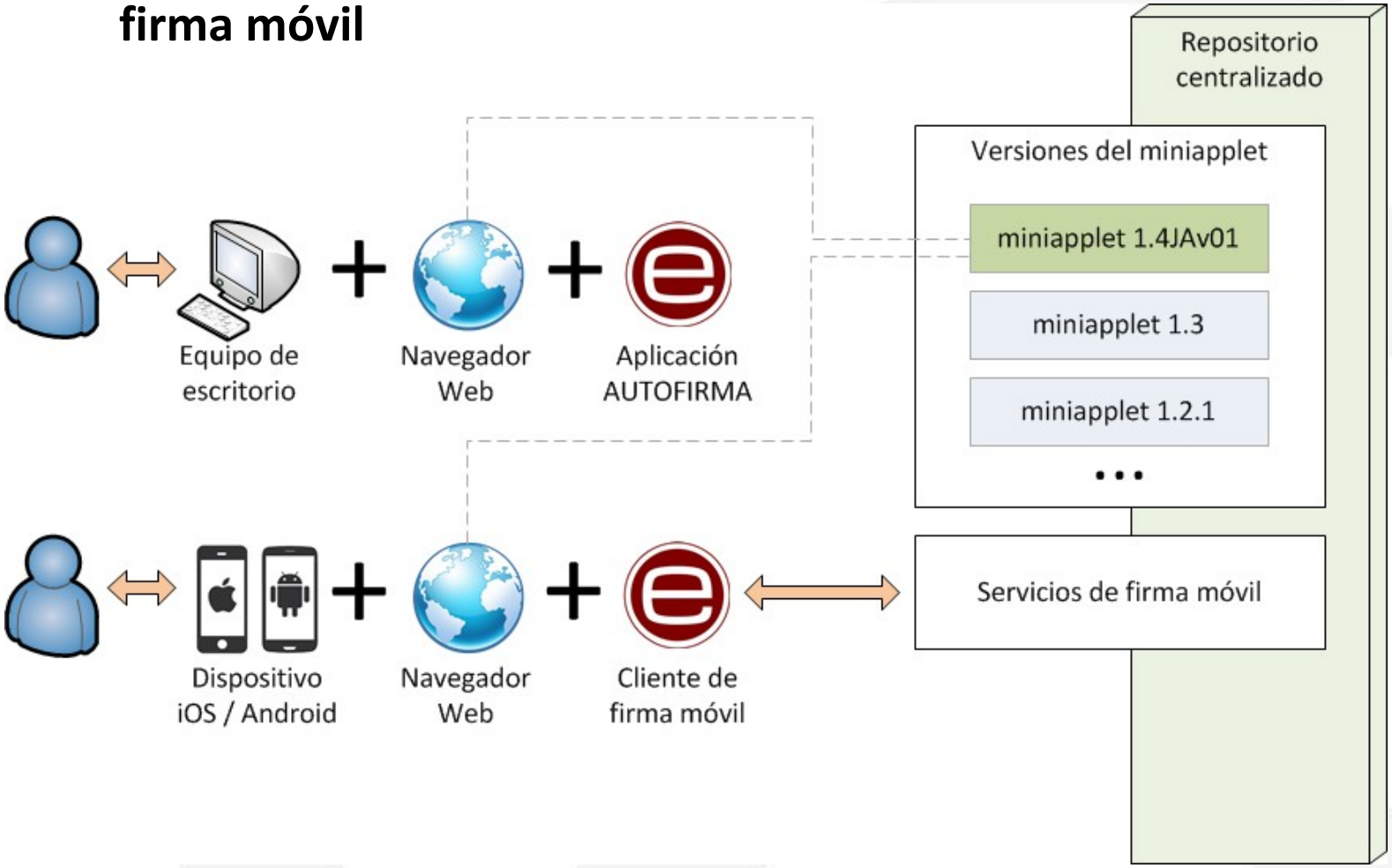
- **Modelo de despliegue de Autofirma**



Autofirma 1.4.2.JAv02

Esquema de funcionamiento

- **Modelo de despliegue de Autofirma y firma móvil**



Autofirma 1.4.2.JAv02

Esquema de funcionamiento

- **Funcionamiento de la nueva versión del miniapplet**

```
...

<script type="text/javascript">
MiniApplet.cargarMiniApplet("https://ws024.juntadeandalucia.es/afirma-validator-miniapplet-1_4/");
</script>

...
```

1) **¿Es un dispositivo móvil o un navegador sin soporte JAVA?** → Aplicación externa
2) **¿Está JAVA habilitado?** →
3) **En cualquier otro caso** → Aplicación externa

Autofirma 1.4.2.JAv02

Esquema de funcionamiento

- **Funcionamiento de la nueva versión del miniapplet**
  - **3 modos de ejecución:**
    - Firma mediante **Applet** (sólo en entornos de escritorio, con Java instalado, y un navegador compatible con applets Java)
    - Firma mediante **Cliente móvil** (sólo en dispositivos iOS y Android). **Requiere obligatoriamente de los servicios de firma móvil.**
    - Firma mediante **Autofirma** (en PCs que utilicen un navegador que no esté cargando el plugin Java para applets). **No necesita los servicios de firma móvil.**

Autofirma 1.4.2.JAv02

¿Cómo se integra?

- Integración del miniapplet, compatible con Firma móvil, Autofirma y Applet.

```
...
<script type="text/javascript">

// URL del miniapplet (ubicación de miniapplet-full_1_4_JAv02.jar y miniapplet.js)
var urlmapplet = "https://ws024.juntadeandalucia.es/afirma-validator-miniapplet-1_4/";

// Carga del miniapplet. Se autodetecta el entorno de ejecución (escritorio o móvil)
MiniApplet.cargarMiniApplet(urlmapplet);

// Fuerza el uso de los servicios de firma móvil para entornos de escritorio. Por defecto, el
// miniapplet tratará de utilizar tecnología de SOCKET SSL para el intercambio de información.
MiniApplet.setForceWSMode(false);

// Fuerza el uso de la invocación por protocolo "afirma://" para ejecutar Autofirma, siendo el valor por defecto. Se
//utiliza en contra de una funcionalidad experimental para la carga de Autofirma mediante Java WebStart (JNLP).
MiniApplet.setForceAFirma(true);

// Servicios intermedios. Adicionalmente, para la firma móvil será necesario indicar el parámetro extra serverUrl
// para habilitar la firma trifásica. ("serverUrl=" + urlmapplet + "sign/TriPhaseSignatureService\n").
MiniApplet.setServlets(urlmStorageServiceapplet + "sign/StorageService", urlmapplet + "sign/RetrieveService");

</script>
...
```

Autofirma 1.4.2.JAv02

Integración multimodo I: Definición

- Se ha definido un nuevo método de firma ***multiModeSign*** en el JavaScript miniapplet.js, qué servirá para realizar cualquier tipo de firma, ya sea simple o masiva y utilizando Applet, Autofirma o Móvil.
- En el caso de la firma móvil la firma masiva no está soportada, de forma que el método multiModeSign solo funcionará correctamente con *arrayLength = 1*.

```
// JUNTA DE ANDALUCÍA: Método multiModeSign
// Método centralizado para la firma simple y masiva, utilizando Autofirma, Miniapplet o Móvil. El miniapplet tiene
// preferencia

// Parámetros:
// operationArray: Define cada operación a realizar ('sign, 'cosign' o 'countersign').
// dataArray: Define los datos a firmar, pudiendo ser datos o hashes ('sign'), o firmas ('cosign' y 'countersign').
// originalDataArray: Define los datos originales, solo útil para la operación 'cosign'. Es necesario rellenar con cadena
// vacía ("") en los casos de 'sign' y 'countersign'.
// arrayLength: Número entero que define el tamaño de las tablas, de forma que las tablas deben coincidir con este
// tamaño.
// commonSignAlgorithm: Algoritmo de firma común (Obligatorio:'SHA1withRSA', 'SHA256withRSA', 'SHA512withRS).
// commonSignFormat: Formato de firma común (Obligatorio: 'AUTO', 'CAdES', 'XAdES', 'PAdES', …).
// commonSignParams: Parámetros extra comunes para todas las operaciones, separados por '\n'.
// successCallback: Función JavaScript de retorno SIN errores (Recomendado: successCallback(signatureB64,
// certificateB64)).
// errorCallback: Función JavaScript de retorno CON errores (Recomendado: errorCallback(type, message)).

multiModeSign(operationArray, dataArray, originalDataArray, arrayLength, commonSignAlgorithm,
commonSignFormat, commonSignParams, successCallback, errorCallback);
```

Autofirma 1.4.2.JAv02

Integración multimodo II: Callbacks

- El método de integración multimodo devuelve los resultados a través de funciones de callback, qué debe implementar el integrador tal y como ya se hacía con los métodos de firma nativos del script miniapplet.js
- A la hora de implementar las funciones de callback hay que tener en cuenta que cuando usamos miniapplet, el callback será invocado una vez por cada operación y cuando usamos Autofirma se invocará una sola vez para todas las operaciones.
- El método de callback se invoca con dos parámetros que contienen los resultados, la firma (signatureB64) y la clave pública del firmante (caertificateB64), aunque este último parámetro no se recibirá con los cliente móviles.
- En el ejemplo siguiente, tenemos en cuenta dos campos hidden para almacenar los resultados de las firmas y el certificado ('result' y 'signerCert' respectivamente). El resultado final será el mismo para Miniapplet y Autofirma, obteniendo en el campo 'result' las firmas en base64 separadas por el carácter ':'

```
function multiModeResultCallback(signatureB64, certificateB64) {
		document.getElementById('signerCert').value = certificateB64;
		if(document.getElementById('result').value == "") {
			document.getElementById('result').value = signatureB64;
		} else {
			document.getElementById('result').value += ":" + signatureB64;
		}
}
```

Autofirma 1.4.2.JAv02

Integración multimodo III: Ejemplo de firma (simple)

```
function firmar() {
		// Inicialización de los parámetros de entrada para el método de firma multimodo.
		var operationArray = [];
		var dataArray = [];
		var originalDataArray = [];
		var arrayLength = 1;
		var commonSignAlgorithm = "SHA256withRSA";
		var commonSignFormat = "CAdES";
		var commonSignParams = "precalculatedHashAlgortihm=SHA-256\n";
		var successCallback = showMassiveResultCallback;
		var errorCallback = showErrorCallback;

		// Asignación de valores para las operaciones del método de firma multimodo.
		operationArray[0] = "sign";
		dataArray[0] = "<HASH-256-DATA-B64>";
		originalDataArray[0] = "";

		MiniApplet.multiModeSign(operationArray, dataArray, originalDataArray,
		arrayLength, commonSignAlgorithm, commonSignFormat, commonSignParams,
		multiModeResultCallback, errorCallback);
}
```

Autofirma 1.4.2.JAv02

Integración multimodo IV: Ejemplo de firma (masiva)

```
function firmar() {
        // Inicialización de los parámetros de entrada para el método de firma multimodo.
        var operationArray = [];
        var dataArray = [];
        var originalDataArray = [];
        var arrayLength = 2;
        var commonSignAlgorithm = "SHA256withRSA";
        var commonSignFormat = "CAdES";
        var commonSignParams = "precalculatedHashAlgortihm=SHA-256\n";
        var successCallback = showMassiveResultCallback;
        var errorCallback = showErrorCallback;

        // Asignación de valores para las operaciones del método de firma multimodo.
        operationArray[0] = "sign";
        operationArray[1] = "sign";
        dataArray[0] = "<HASH-256-DATA1-B64>";
        dataArray[1] = "<HASH-256-DATA2-B64>";
        originalDataArray[0] = "";
        originalDataArray[1] = "";

        MiniApplet.multiModeSign(operationArray, dataArray, originalDataArray, arrayLength,
        commonSignAlgorithm, commonSignFormat, commonSignParams,
        multiModeResultCallback, errorCallback);
}
```

Autofirma 1.4.2.JAv02

Integración multimodo V: Ejemplo de cofirma (masiva)

```
function firmar() {
        // Inicialización de los parámetros de entrada para el método de firma multimodo.
        var operationArray = [];
        var dataArray = [];
        var originalDataArray = [];
        var arrayLength = 2;
        var commonSignAlgorithm = "SHA256withRSA";
        var commonSignFormat = "CAdES";
        var commonSignParams = "";
        var successCallback = showMassiveResultCallback;
        var errorCallback = showErrorCallback;

        // Asignación de valores para las operaciones del método de firma multimodo.
        operationArray[0] = "cosign";
        operationArray[1] = "cosign";
        dataArray[0] = "<SIGNATURE1-256-B64>";
        dataArray[1] = "<SIGNATURE2-256-B64>";
        originalDataArray[0] = "<ORIGINAL-DATA1-B64>";
        originalDataArray[1] = "<ORIGINAL-DATA2-B64>";

        MiniApplet.multiModeSign(operationArray, dataArray, originalDataArray, arrayLength,
        commonSignAlgorithm, commonSignFormat, commonSignParams,
        multiModeResultCallback, errorCallback);
}
```

Autofirma 1.4.2.JAv02

Integración multimodo VI: Ejemplo de contrafirma (masiva)

```
function firmar() {
		// Inicialización de los parámetros de entrada para el método de firma multimodo.
		var operationArray = [];
		var dataArray = [];
		var originalDataArray = [];
		var arrayLength = 2;
		var commonSignAlgorithm = "SHA256withRSA";
		var commonSignFormat = "CAdES";
		var commonSignParams = "target=leafs\n";
		var successCallback = showMassiveResultCallback;
		var errorCallback = showErrorCallback;

		// Asignación de valores para las operaciones del método de firma multimodo.
		operationArray[0] = "countersign";
		operationArray[1] = "countersign";
		dataArray[0] = "<SIGNATURE1-256-B64>";
		dataArray[1] = "<SIGNATURE2-256-B64>";
		originalDataArray[0] = "";
		originalDataArray[1] = "";

		MiniApplet.multiModeSign(operationArray, dataArray, originalDataArray, arrayLength,
		commonSignAlgorithm, commonSignFormat, commonSignParams,
		multiModeResultCallback, errorCallback);
}
```

Autofirma 1.4.2.JAv02

Protocolo de comunicación

- Utilizando servidor intermedio:

afirma://sign?
op=sign&id=ynynfOwjUfDGBltuARRs&key=29102601&keystore=MOZ_UNI&stservlet=http%3A%2F%2F10.90.43.42%3A8080%2Fminiapplet14ja%2Fafirma-signature-storage
%2FStorageService&format=CAdES&algorithm=SHA1withRSA&properties=c2VydmVyVXJsPWh0dHA6Ly9YWC5YWC5YWC5YWDo4MDgwL2FmaXJtYS1zZXJ2ZXItdHJpcGhhc2Utc2lnbmVyL1NpZ25hdHVyZVNlcnZpY2UK&dat=TG9yZW1pcHN1bWRvbG9yc2l0YW1ldA==

- Se genera una única URL que utiliza el protocolo **afirma**
- La URL contiene los datos a firmar (parámetro **dat**), la dirección del servicio **StorageService** y otros parámetros sobre el tipo de firma a generar.
- Si se utiliza **Firefox**, Autofirma utiliza el almacén de certificados del navegador (caso contrario, se utiliza el del Sistema Operativo).

Autofirma 1.4.2.JAv02

Protocolo de comunicación

- Sin servidor intermedio:

```
afirma://service?
ports=53375,49810,59653&v=1&idsession=Za4WBz57MLSeLIF6xuUV
```

```
afirma://sign?
op=sign&keystore=MOZ_UNI&algorithm=SHA1withRSA&format=CAdES&properties=c2VydmVyVXJsPWh0dHA6Ly9YWC5YWC5YWC5YWDo4MDgwL2FmaXJtYS1zZXJ2ZXItdHJpcGhhc2Utc2lnbmVyL1NpZ25hdHVyZVNlcnZpY2UK&dat=TG9yZW1pcHN1bWRvbG9yc2l0YW1ldA==
```

- Se generan 2 URL de protocolo **afirma:**
  - La primera para negociar el puerto de comunicaciones interno
  - La segunda para enviar la información relativa a la firma
- La URL contiene los datos a firmar (parámetro **dat**) y otros parámetros sobre el tipo de firma a generar.

Autofirma 1.4.2.JAv02

Matriz de compatibilidad Autofirma (SocketSSL)

	JRE 1.8 (1.8.0_60+) Socket SSL							
	Firefox	I. Explorer					Chrome	Safari
S.O.	46	8	9	10	11	Edge	51	9
Windows XP SP3	OK	N/S (*)	N/S	N/S	N/S	N/S	OK	N/S
Windows 7	OK	N/S (*)	N/D	N/D	OK	N/S	OK	N/S
Windows 8.1	OK	N/S (*)	N/D	N/D	OK	N/S	OK	N/S
Windows 10	OK	N/S (*)	N/D	N/D	OK	OK	OK	N/S
Ubuntu 14.04	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Ubuntu 16.04	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Guadalinex v9	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Mac OS X 10.11 El Capitán	OK	N/S	N/S	N/S	N/S	N/S	OK	NOK (1)

(1) No funciona correctamente la comunicación con el SocketSSL de Autofirma. El error mostrado en la consola de Safari es: Failed to load resource: El certificado de servidor no es válido. Es posible que el servidor al que intenta conectarse simule ser “127.0.0.1”, lo que podría poner en peligro su información confidencial.
(*) Cuando se usa IE8 la comunicación será siempre mediante servicios intermedios, ya que la comunicación mediante SocketSSL en este navegador no está soportada.
Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente.

Autofirma 1.4.2.JAv02

Matriz de compatibilidad Autofirma (Servlets)

Se puede configurar Autofirma para que utilice los servicios de firma móvil, aunque no se recomienda.

	JRE 1.8 (1.8.0_60+) Servidor intermedio							
	Firefox	I. Explorer					Chrome	Safari
S.O.	46	8	9	10	11	Edge	51	9
Windows XP	OK	OK	N/S	N/S	N/S	N/S	OK	N/S
Windows 7	OK	N/D	N/D	N/D	OK	N/S	OK	N/S
Windows 8.1	OK	N/D	N/D	N/D	OK	N/S	OK	N/S
Windows 10	OK	N/D	N/D	N/D	OK	OK	OK	N/S
Ubuntu 14.04	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Ubuntu 16.04	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Guadalinex v9	OK	N/S	N/S	N/S	N/S	N/S	OK	N/S
Mac OS X 10.11 El Capitán	OK	N/S	N/S	N/S	N/S	N/S	OK	OK

Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente..

Firma móvil

- ¿En qué consiste?
- Esquema de funcionamiento
- ¿Cómo se integra?
- Matriz de compatibilidad móvil

Firma móvil

¿En qué consiste?

- **¿Qué es la firma móvil?**

  Es la posibilidad que permite el miniapplet de firmar en un dispositivo móvil mediante una aplicación nativa, mediante una invocación por protocolo de forma similar a como funciona la aplicación autofirma. Es por tanto una aplicación nativa de cada plataforma**, independiente del navegador web,** que permite a los usuarios la realización de firmas electrónicas **en dispositivos móviles.**

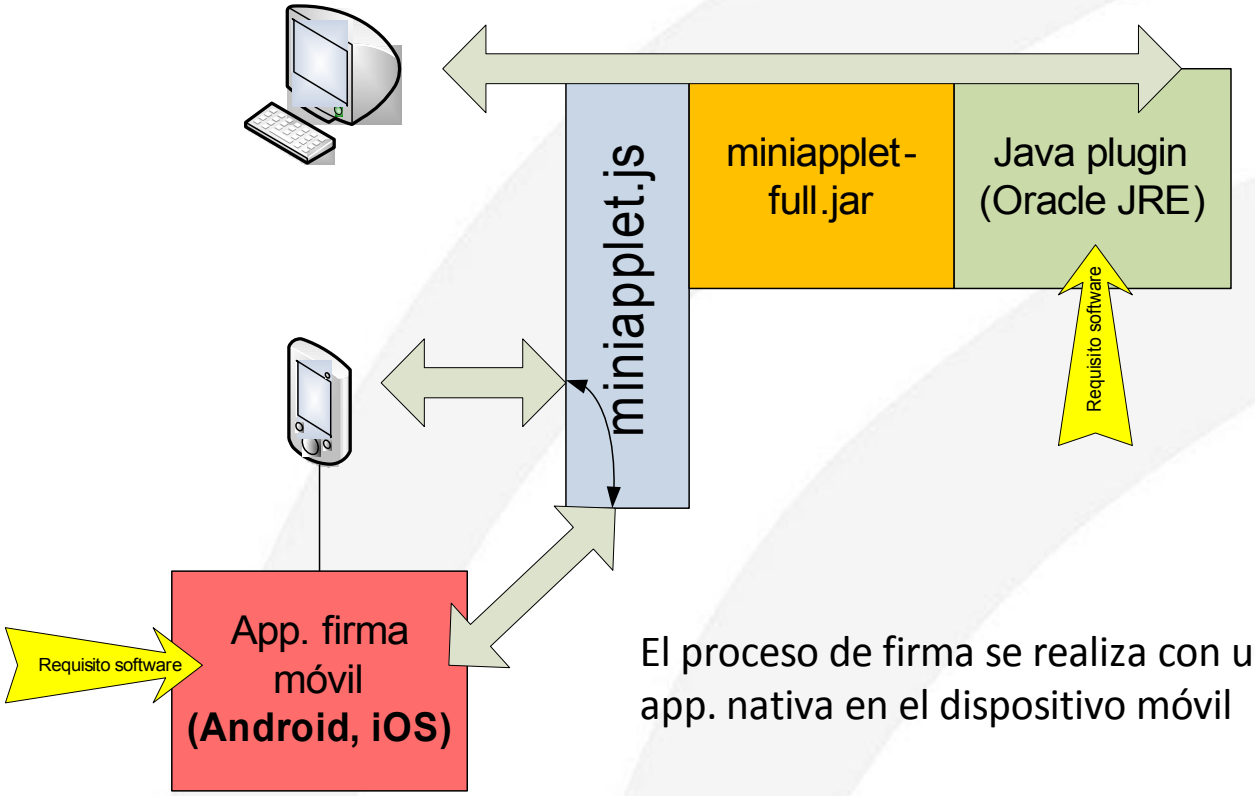
- **¿En qué plataformas está disponible?**

  - En Android 4 o superior: v1.4 (publicado el 28 de Marzo de 2016). Disponible en Google Play: **Cliente movil @firma**
  - En iOS 8 o superior: 1.4.7 (publicado el 13 de Mayo de 2016). Disponible en Apple Store: **Cliente @firma movil**

Firma móvil

Esquema de funcionamiento

- Interacción automática con el **cliente de firma móvil** disponible en dispositivos **Android** e **iOS**.



El proceso de firma se realiza con una app. nativa en el dispositivo móvil

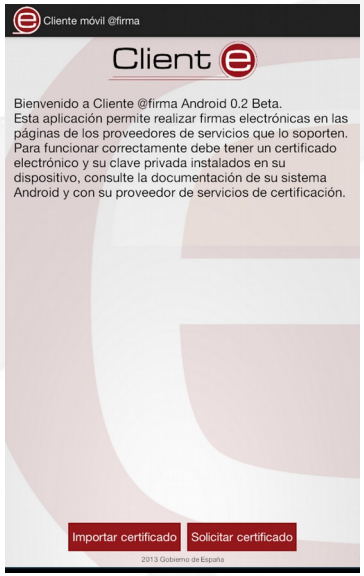
Firma móvil

Esquema de funcionamiento

- Aplicación nativa Android
  - Compatible con Android 4.0 o superior.
  - Firma CAdES y PAdES completa en el propio dispositivo.
  - Firma CAdES, XAdES y PAdES, en tres fases con soporte servidor (Recomendado).
  - Firma de ficheros locales.

Dispositivo Android | Cliente de firma móvil | Servicio de firma móvil

- Aplicación nativa iOS
  - Compatible con iOS 8 o superior



Firma móvil

Esquema de funcionamiento

- **Servicio de firma móvil**: Soporte para el intercambio de firmas entre la aplicación cliente y el componente Miniapplet. Además, completa las **firmas trifásicas**. Se compone de tres servlets **transparentes para la aplicación cliente:**
  - **StorageService**: almacena la firma de forma temporal en el servidor a partir de un identificador que envía la aplicación nativa.
  - **RetrieveService**: recupera la firma del servidor a partir del identificador indicado.
  - **TriPhaseSignatureService**: tratamiento de la firma en el servidor.
- **Proceso de firma móvil**:

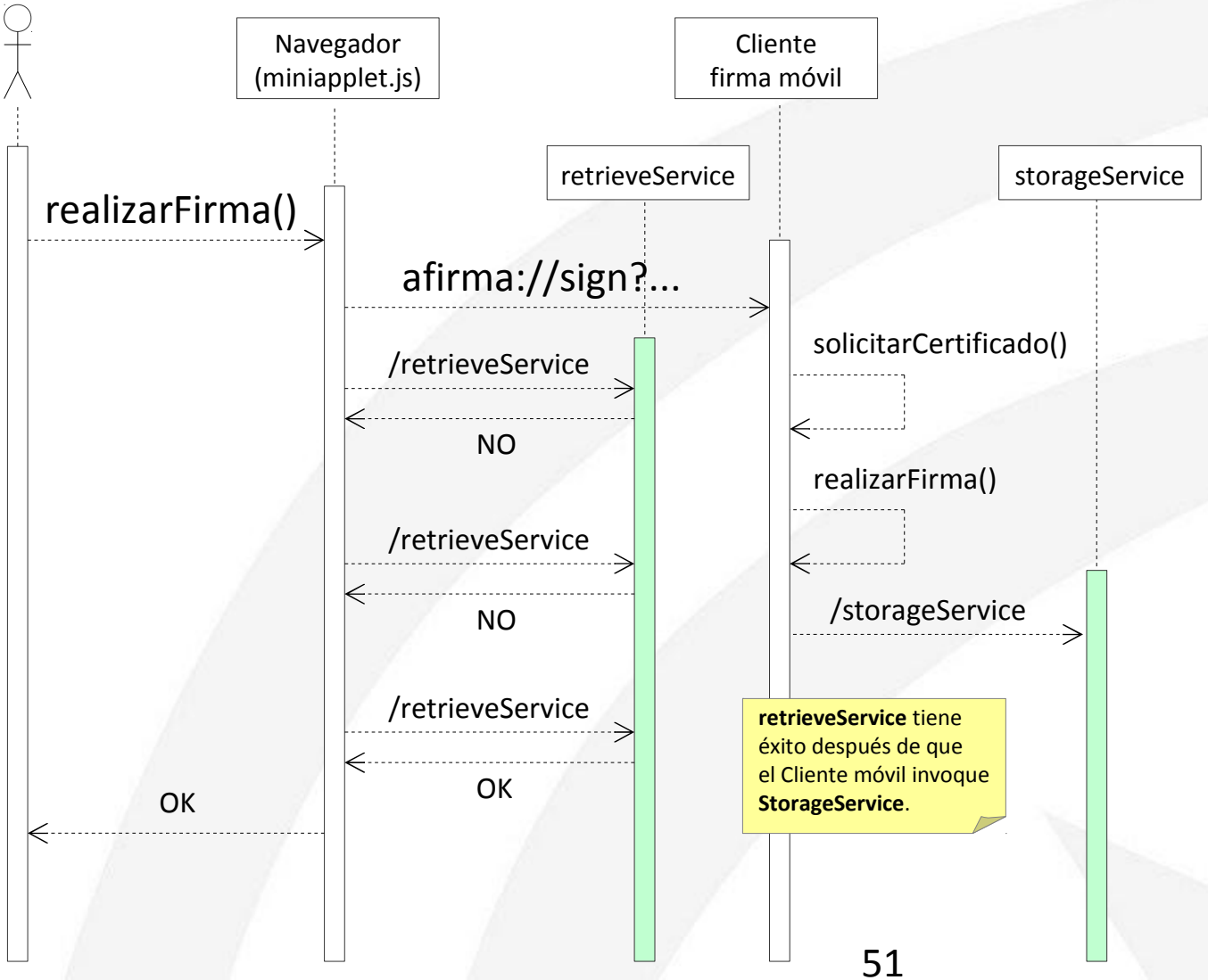
1. Miniapplet invoca a la aplicación nativa pasando como parámetros los datos a firmar.
2. Aplicación nativa solicita el certificado y realiza la firma.

**** Firma en tres fases: tratamiento de la firma mediante el servlet TriPhaseSignatureService: Pre-firma, Firma y Post-Firma***

3. **Aplicación nativa sube la firma al servidor mediante el servlet StorageService**
4. **Miniapplet recupera la firma mediante el servlet RetrieveService y la devuelve.**

Firma móvil

Esquema de funcionamiento



Firma móvil

¿Cómo se integra?

- Integración de la firma móvil

```
...
<script type="text/javascript">

// URL del miniapplet (ubicación de miniapplet-full.jar, miniapplet.js)
var urlmapplet = https://ws024.juntadeandalucia.es/clienteafirma/miniapplet14ja/

// URL del servlet StorageService
var storageService = urlmapplet + 'sign/StorageService';
// URL del servlet RetrieveService
var retrieveService = urlmapplet+ 'sign/RetrieveService';

MiniApplet.cargarMiniApplet(urlmapplet);
MiniApplet.setServlets(storageService, retrieveService); // Servicios de firma móvil

// URL servlet TriPhaseSignatureService
var params = 'serverUrl=' + urlmapplet +'sign/TriPhaseSignatureService\n';

MiniApplet.sign(dataB64, "SHA1withRSA", formato, params+"format=XAdES
Detached\nmode=implicit", firmaNotificacionSuccess, firmaNotificacionError );

</script>
...
```

Firma móvil

Matriz de compatibilidad móvil

	Formato de firma					
	CAdES		XAdES		PAdES	
S.O.	Mono	Tri	Mono	Tri	Mono	Tri
Android 4 (KitKat 4.4.4)	OK (*)	OK (*)	NOK (1)	OK (*)	OK (*)	OK (*)
Android 5 (Lollipop 5.1.0)	OK (*)	OK (*)	NOK (1)	OK (*)	OK (*)	OK (*)
Android 6 (Marshmallow 6.0.0)	OK (*) (**)	OK (*) (**)	NOK (1)	OK (*) (**)	OK (*) (**)	OK (*) (**)
IOS 9	NOK (2)	OK	NOK (3)	OK	NOK (3)	OK

(1) El resultado devuelto no es una firma electrónica.

(2) El cliente genera una firma inválida. Problema con el atributo firmado SigningCertificate, el identificador del certificado ESSCertId no es correcto.

(3) El cliente de firma se cierra al pulsar el botón firmar.

(*) Existe un problema de comunicación con los servicios intermedios que afecta al navegador por defecto de Android. Este navegador no permite el paso de ficheros mediante Base64 usando directamente el método de firma JavaScript cuando superan de cierto tamaño, utilizando Firefox o Chrome se corrige este problema.

(**) Las funciones de abrir y guardar fichero propias del cliente de firma (v1.4 28/03/2016) bajo Android 6.0.0, cierran inesperadamente la aplicación. Este problema se corrige pasando los datos a firmar en Base64 usando directamente el método de firma JavaScript.

Todas las firmas generadas satisfactoriamente (OK) han sido validadas contra @firma5.5 correctamente.

Compatibilidad general del Miniapplet

Entornos de escritorio(miniapplet y autofirma)

	Firefox	I. Explorer		Edge	Chrome	Safari
S.O.	46	8	11	20	50	9
Windows XP SP3	AP (1)(3) AF(3)	AP AF(2)	N/D	N/D	AF	N/D
Windows 7	AP(1)(3) AF(3)	N/D	AP AF	N/D	AF	N/D
Windows 8.1	AP(1)(3) AF(3)	N/D	AP AF	N/D	AF	N/D
Windows 10	AP(1)(3) AF(3)	N/D	AP AF	AF	AF	N/D
Ubuntu 14.04	AP AF	N/D	N/D	N/D	AF	N/D
Ubuntu 16.04	AP AF	N/D	N/D	N/D	AF	N/D
Mac OS X 10.11 El Capitán (4)	AP(4) AF	N/D	N/D	N/D	AF	AP(4)(5) AF(6)

Leyenda:
AP: Compatible con APPLET.
AF: Compatible con AUTOFIRMA.
N/D: Configuración no disponible.
NF: No funciona con ningún cliente (applet ni autofirma).

Colores:
Verde: Configuración compatible.
Amarillo: Configuración parcialmente compatible. Puede presentar algún problema.
Rojo: Configuración no compatible.

(1) Requiere Java 7 o Java 8 últimas versiones. No funciona con Java 6.
(2) Requiere el uso de los servlets (servidor intermedio). No funciona si autofirma está configurado sin servidor intermedio.
(3) En ocasiones se pierde el foco y no es posible seleccionar un certificado en la ventana de selección de certificados. En este caso sólo se puede cerrar el navegador y volver a intentarlo.
(4) Requiere Java 7 o Java 8 en sus últimas versiones. No existe distribución de Java 6 para este sistema operativo.
(5) Existe un bug reconocido en Safari al ejecutar módulos Java en modo seguro. Para poder firmar sin problemas es necesario configurar la ejecución de código Java desde Safari en modo no seguro para los sitios que cargan el miniapplet. (Safari->Preferencias->Seguridad->Ajustes de módulo (Java)->Ejecutar en modo no seguro (Sitio del miniapplet)). https://bugs.openjdk.java.net/browse/JDK-8148430
(6) No funciona correctamente la comunicación con el SocketSSL de Autofirma. El error mostrado en la consola de Safari es: **Failed to load resource: El certificado de servidor no es válido. Es posible que el servidor al que intenta conectarse simule ser "127.0.0.1", lo que podría poner en peligro su información confidencial**. Sólo funciona en configuración con servidor intermedio.

Compatibilidad general del Miniapplet

Entornos móviles

S.O.	Compatible
Android 4 (KitKat 4.4.4)	OK(1)
Android 5 (Lollipop 5.0.2)	OK(1)
Android 6 (Marshmallow)	OK(1)(2)
IOS 9	OK

(1) El navegador stock de android no funciona correctamente cuando el tamaño de la firma es de cierto tamaño. Se recomienda el uso del navegador Chrome o Firefox para las operaciones de firma electrónica.
(2) Las funciones de abrir y guardar fichero provocan que el cliente de firma se cierre. Se recomienda la firma de datos en base 64 usando el correspondiente método del javascript.

Compatibilidad general del Miniapplet

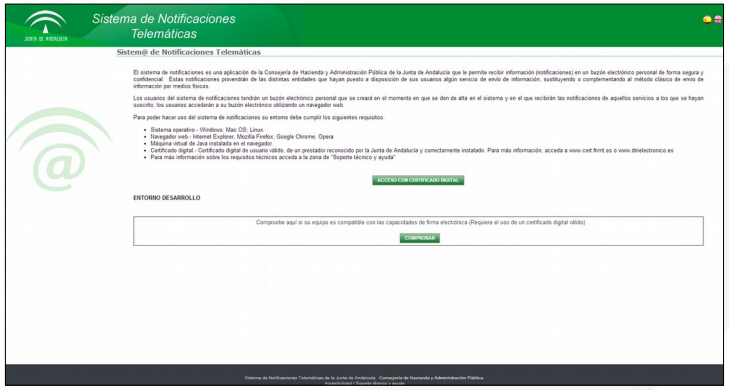
Resumen de compatibilidad de formatos de firma

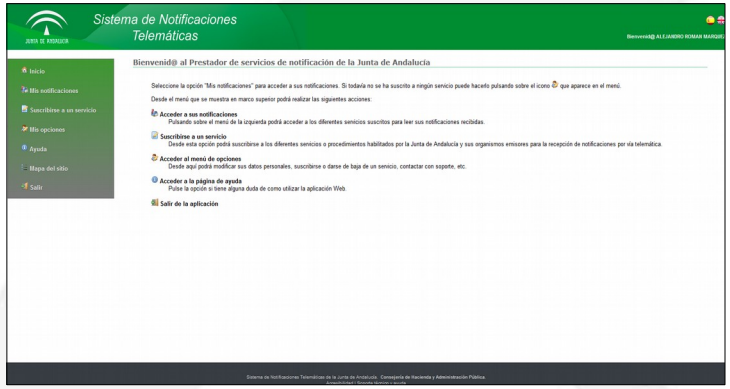
	Formato de firma					
	CAdES		XAdES		PAdES	
S.O.	Mono	Tri	Mono	Tri	Mono	Tri
Miniapplet	OK	OK	OK	OK	OK	OK
Autofirma	OK	OK	OK	OK	OK	OK
Android 4 (KitKat 4.4.4)	OK	OK	NOK	OK	OK	OK
Android 5 (Lollipop 5.0.2)	OK	OK	NOK	OK	OK	OK
Android 6 (Marshmallow)	OK	OK	NOK	OK	OK	OK
IOS 9	NOK	OK	NOK	OK	NOK	OK

Ejemplo de Integración

Notific@

- Firma Miniapplet escritorio:

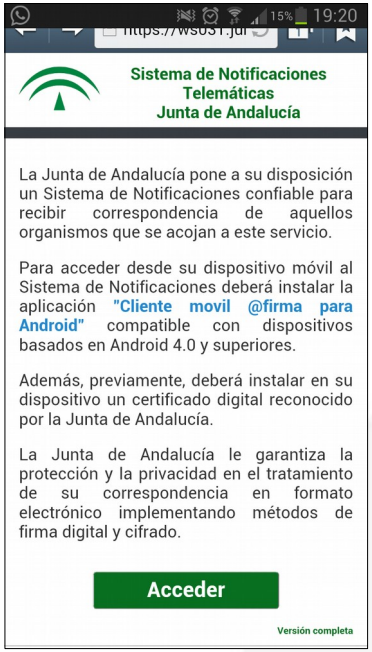


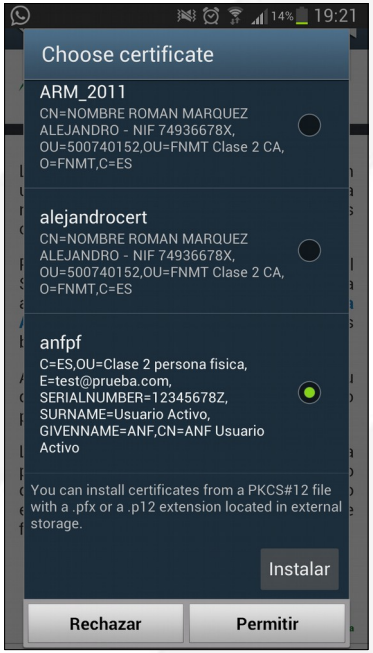


Ejemplo de Integración

Notific@

- Firma Miniapplet móvil:





Muchas gracias

Dirección General de Política Digital

Consejería de Hacienda y Administración Pública