eID Digital Signature Service

Guía para desarrolladores de Software

Rolosa HyJ S.A. - MICITT

27 de Julio de 2014

Resumen

El presente documento es una guía para desarrolladores de software útil como punto de entrada para integrar el eID DSS en una aplicación web.

Tabla de contenido

Introducción

El eID DSS provee un web service que puede ser utilizado por aplicaciones web para integrar la creación de firmas digitales dentro de los flujos de trabajo del negocio mismo de dichas aplicaciones web. El eID DSS soporta diferentes formatos de documentos y sus correspondientes formatos de firma digital. el eID soporta firmas XAdES-X-L v1.4.2. Los documentos soportados son:

- Documentos XML
- Documentos ODF (Open Office)
- Documentos OOXML (Microsoft Office 2007 y 2010).
- Contenedores ZIP conteniendo los archivos que se desean firmar como un todo.

Este web service está construido de acuerdo al estándar OASIS DSS el cual se describe en esta guía para desarrolladores.

El protocolo involucra 3 actores:

- La aplicación web que desea que el usuario firme un documento.
- El servicio DSS que ayuda a que el usuario final firme el documento.
- El usuario final (cliente del navegador) que realiza la firma digital del documento.

Los namespaces XML utilizados a lo largo del documento son descritos en la siguiente tabla:

Prefix	Namespace
dss	urn:oasis:names:tc:dss:1.0:core:schema
async	urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0
ds	http://www.w3.org/2000/09/xmldsig#
soap	http://www.w3.org/2003/05/soap-envelope
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
wst	http://docs.oasis-open.org/ws-sx/ws-trust/200512
ec	http://www.w3.org/2001/10/xml-exc-c14n#
dssp	urn:be:e-contract:dssp:1.0
vr	urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#

Pre-requisitos

Para proseguir con el presente documento, es necesario disponer de:

- Una Tarjeta Inteligente para Firma Digital de Costa Rica
- Un lector de tarjetas apropiado
- Certificados Digitales de la cadena de la confianza de la Firma Digital de Costa Rica
- Drivers necesarios para la Tarjeta Inteligente, dependiendo el sistema operativo en el que se desee realizar los trabajos.

Los drivers y certificados digitales tienen que ser obtenidos desde el sitio de Soporte de Firma Digital de Costa Rica:

https://www.soportefirmadigital.com/sfd/default.aspx

Ejecución básica del protocolo

La ejecución básica del protocolo consiste de tres mensajes de petición/respuesta.

- La aplicación web sube el documento para ser firmado al DSS
- La petición real de firmado desde la aplicación web hacia el DSS
- La aplicación web descarga el documento firmado del DSS

Subiendo documentos

La ejecución del protocolo comienza con una petición SOAP versión 1.2 desde la aplicación web hacia el DSS. Este primer paso permite que la aplicación web envíe el documento a ser firmado hacia el DSS. También permite que la aplicación web y el DSS pueda establecer un contexto de seguridad. El documento no es transmitido a través del POST del navegador.

```
<soap:Envelope>
    <soap:Body>
        <dss:SignRequest Profile="urn:be:e-contract:dssp:1.0">
            <dss:OptionalInputs>
                <dss:AdditionalProfile>
            urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
                </dss:AdditionalProfile>
                <wst:RequestSecurityToken>
                    <wst:TokenType>
        http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct
                    </wst:TokenType>
                    <wst:RequestType>
                http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
                    </wst:RequestType>
                    <wst:Entropy>
                        <wst:BinarySecret Type=
            "http://docs.oasis-open.org/ws-sx/ws-trust/200512/Nonce">
                            ...
                        </wst:BinarySecret>
                    </wst:Entropy>
                    <wst:KeySize>256</wst:KeySize>
```

```
                </wst:RequestSecurityToken>
                <dss:SignaturePlacement WhichDocument="doc1"
                    CreateEnvelopedSignature="true"/>
                <dss:SignatureType>
                    urn:be:e_contract:dssp:signature:xades-x-l
                </dss:SignatureType>
            </dss:OptionalInputs>
            <dss:InputDocuments>
                <dss:Document ID="doc1">
                   <dss:Base64Data MimeType="...">the document</dss:Base64Data>
                </dss:Document>
            </dss:InputDocuments>
        </dss:SignRequest>
    </soap:Body>
</soap:Envelope>
```

Se utiliza el perfil asíncrono abstracto de OASIS (DSSAsync) dada la naturaleza de esta primera petición

Dado el hecho de que el documento transmitido puede contener información sensible, la petición SOAP es transmitida sobre SSL. Esto permite a la aplicación web autenticarse y confiar en el DSS.

A través de WS-SecureConcersation, la aplicación web y el DSS pueden establecer una clave de sesión compartida. Esta clave de sesión será utilizada en intercambio de mensajes subsecuentes. No se utilizan las características de WS_Addressing debido a que no se necesitan las capacidades de ruteo definidas en la especificación del WS-SecureConversation.

Si se requiere, la aplicación web puede autenticarse a sí misma con una cabecera SOAP de WS_Security. Esto permite que el DSS pueda operar en dos modos:

- Modo de desarrollo, donde las aplicaciones web pueden acceder el DSS sin estar autorizadas
- Modo de Producción, donde las aplicaciones web deben ser autorizadas por el DSS.

El DSS soporta múltiples tipos de firma, se incluye un elemento opcional <dss:SignatureType>.

Se definen los siguiente URIs de tipos de firmas

- urn:be:e-contract:dssp:signature:xades-x-l
- urn:be:e-contract:dssp:signature:xades-baseline
- urn:be:e-contract:dssp:signature:pades-baseline

Si el elemento <dss:SIgnatureType> no es provisto, el DSS determinara el tipo de firma más apropiado.

El servidor DSS responde como se muestra a continuación:

```
<soap:Envelope>
    <soap:Body>
        <dss:SignResponse Profile="urn:be:e-contract:dssp:1.0">
            <dss:Result>
                <dss:ResultMajor>
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
                </dss:ResultMajor>
            </dss:Result>
            <dss:OptionalOutputs>
                <async:ResponseID>responseId</async:ResponseID>
                <wst:RequestSecurityTokenResponseCollection >
                    <wst:RequestSecurityTokenResponse>
                        <wst:TokenType>
       http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct
                        </wst:TokenType>
                        <wst:RequestedSecurityToken>
                            <wsc:SecurityContextToken wsu:Id="token-ref">
                                <wsc:Identifier>
                                    token-id
                                </wsc:Identifier>
                            </wsc:SecurityContextToken>
                        </wst:RequestedSecurityToken>
                        <wst:RequestedAttachedReference>
                            <wsse:SecurityTokenReference>
                                <wsse:Reference URI="#token-ref"/>
                            </wsse:SecurityTokenReference>
                        </wst:RequestedAttachedReference>
                        <wst:RequestedUnattachedReference>
                            <wsse:SecurityTokenReference>
                                <wsse:Reference
ValueType="http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct"
                URI="token-id" />
                            </wsse:SecurityTokenReference>
                        </wst:RequestedUnattachedReference>
                        <wst:RequestedProofToken>
```

```
                        <wst:ComputedKey>
            http://docs.oasis-open.org/ws-sx/ws-trust/200512/CK/PSHA1
                        </wst:ComputedKey>
                    </wst:RequestedProofToken>
                    <wst:Entropy>
                        <wst:BinarySecret Type=
            "http://docs.oasis-open.org/ws-sx/ws-trust/200512/Nonce">
                            ...
                        </wst:BinarySecret>
                    </wst:Entropy>
                    <wst:KeySize>256</wst:KeySize>
                    <wst:Lifetime>
                        <wsu:Created>...</wsu:Created>
                        <wsu:Expires>...</wsu:Expires>
                    </wst:Lifetime>
                </wst:RequestSecurityTokenResponse>
            </wst:RequestSecurityTokenResponseCollection>
        </dss:OptionalOutputs>
    </dss:SignResponse>
</soap:Body>
</soap:Envelope>
```

A través del elemento <async:ResponseID>, la aplicación web puede referenciar el documento cargado, que tiene que ser firmado. El token de conversación segura y la correspondiente clave de prueba-de-posesión, son utilizadas para asegurar los mensajes subsecuentes entre la aplicación y el DSS. La clave de prueba-de-posesión es generada utilizando el algoritmo P_SHA-1.

POST del Navegador

Los siguientes mensajes utilizan un POST del Navegador puesto que el DSS requiere que el usuario final interactué con el navegador web y el dispositivo de creación de firma. (ej. tarjeta inteligente).

La aplicación web inicia un POST del Navegador hacia el servidor DSS a través de la siguiente pagina HTML.

```
<html>
    <head><title>DSS Browser POST</title></head>
    <body>
        <p>Redirecting to the DSS Server...</p>
        <form method="post" action="https://www.e-contract.be/dss-ws/start">
            <input type="hidden" name="PendingRequest" value="..."/>
        </form>
        <script type="text/javascript">
            window.onload = function() {
                document.forms[0].submit();
            }
        </script>
    </body>
</html>
```

Aquí el campo PendingRequest del formulario HTML contiene el mensaje codificado en base64 <async:PendingRequest>. Este mensaje se ve a continuación:

```
<async:PendingRequest Profile="urn:be:e-contract:dssp:1.0">
    <dss:OptionalInputs>
        <dss:AdditionalProfile>
            urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
        </dss:AdditionalProfile>
        <async:ResponseID>responseId</async:ResponseID>
        <dssp:PendingRequestContext ID="requiredId"
            IssueInstant="some dateTime"
            Destination="web application landing page URL">
            <ds:Signature>
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod
                        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm=
                        "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
                    <ds:Reference URI="">
                        <ds:Transforms>
                            <ds:Transform Algorithm=
                  "http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                <ds:Transform Algorithm=
                                    "http://www.w3.org/2001/10/xml-exc-c14n#"/>
```

```
                        </ds:Transforms>
                        <ds:DigestMethod
                           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>...</ds:DigestValue/>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue>...</ds:SignatureValue>
                <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                        <wsse:Reference ValueType=
        "http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct"
                            URI="token-id" />
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
        </dssp:PendingRequestContext>
    </dss:OptionalInputs>
</async:PendingRequest>
```

A través del <async:ResponseID>, la aplicación web hace referencia al documento que fue transmitido previamente al servidor DSS por la llamada SOAP descrita en la sección anterior (Subiendo Documentos).

El elemento <dssp:PendingRequestContext> es utilizado para asegurar el mensaje transmitido.

La firma a nivel del mensaje se realiza utilizando la clave de prueba de posesión del token de seguridad. La firma XML está utilizando URI="" para hacer referencia al documento puesto que el elemento <async:PendingRequest> de más alto nivel no define un atributo de identificación.

Después de firmar el documento, el servidor DSS responde con la siguiente pagina HTML:

```
<html>
    <head><title>DSS Browser POST</title></head>
    <body>
        <p>Redirecting to the web application...</p>
        <form method="post" action="value of PendingRequestContext Destination
 attribute">
            <input type="hidden" name="SignResponse" value="..."/>
        </form>
        <script type="text/javascript">
            window.onload = function() {
                document.forms[0].submit();
            }
        </script>
    </body>
</html>
```

Donde el campo SignResponse del formulario HTML contiene el mensaje <dss:SignResponse> codificado en base64. Este mensaje se ve a continuación:

```
<dss:SignResponse Profile="urn:be:e-contract:dssp:1.0">
    <dss:Result>
        <dss:ResultMajor>
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
        </dss:ResultMajor>
    </dss:Result>
    <dss:OptionalOutputs>
        <async:ResponseID>responseId</async:ResponseID>
        <dssp:SignResponseContext ID="responseId"
            InResponseTo="previous requiredId request value"
            IssueInstance="..."
            Destination="...">
            <ds:Signature>
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod
                        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm=
                    "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
                    <ds:Reference URI="">
                        <ds:Transforms>
                            <ds:Transform Algorithm=
                "http://www.w3.org/2000/09/xmldsig#enveloped-signature">
```

```
                            <ds:Transform Algorithm=
                                "http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1"/>
                        <ds:DigestValue>...</ds:DigestValue/>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue>...</ds:SignatureValue>
                <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                        <wsse:Reference ValueType=
        "http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct"
                            URI="token-id" />
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
        </tns:SignResponseContext>
    </dss:OptionalOutputs>
</dss:SignResponse>
```

El Servidor DSS firma el mensaje con la clave de prueba-de-posesión del token de seguridad. La aplicación web debería por supuesto verificar esta firma.

En el caso de que el usuario final cancele la operación de firma, el servicio DSS retorna en <dss:SignResponse> un <dss:ResultMajor> de RequesterError y un <dss:ResultMinor> de:

urn:be:e-contract:dssp:1.0:resultminor:user-cancelled

En el caso que el servicio DSS detecta un problema con el entorno de ejecución del cliente. el servicio retorna en <SignResponse> un <dss:ResultMajor> de RequesterError y un <dss:ResultMinor> de:

urn:be:e-contract:dssp:1.0:resultminor:client-runtimeg

Descarga del documento firmado

Finalmente, la aplicación web puede solicitar al servidor DSS el documento firmado a través de una llamada SOAP:

```
<soap:Envelope>
    <soap:Header>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="timestamp">
                <wsu:Created>...</wsu:Created>
                <wsu:Expires>...</wsu:Expires>
            </wsu:Timestamp>
            <wsc:SecurityContextToken wsu:Id="token-ref">
                    <wsc:Identifier>
                        token-id
                    </wsc:Identifier>
            </wsc:SecurityContextToken>
            <ds:Signature>
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod
                        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm=
                    "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
                    <ds:Reference URI="#timestamp">
                        <ds:Transforms>
                            <ds:Transform Algorithm=
                                "http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod
                           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>...</ds:DigestValue/>
                    </ds:Reference>
                    <ds:Reference URI="#body">
                        <ds:Transforms>
                            <ds:Transform Algorithm=
                                "http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod
                           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>...</ds:DigestValue/>
                    </ds:Reference>
```

```
                </ds:SignedInfo>
                <ds:SignatureValue>...</ds:SignatureValue>
                <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                        <wsse:Reference URI="#token-ref" />
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
        </wsse:Security>
    </soap:Header>
    <soap:Body wsu="body">
        <async:PendingRequest Profile="urn:be:e-contract:dssp:1.0">
            <dss:OptionalInputs>
                <dss:AdditionalProfile>
            urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
                </dss:AdditionalProfile>
                <async:ResponseID>responseId</async:ResponseID>
                <wst:RequestSecurityToken>
                    <wst:RequestType>
        http://docs.oasis-open.org/ws-sx/ws-trust/200512/Cancel
                    </wst:RequestType>
                    <wst:CancelTarget>
                        <wsse:SecurityTokenReference>
                            <wsse:Reference ValueType=
        "http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/sct"
                                URI="token-id"/>
                        </wsse:SecurityTokenReference>
                    </wst:CancelTarget>
                </wst:RequestSecurityToken>
            </dss:OptionalInputs>
        </async:PendingRequest>
    </soap:Body>
</soap:Envelope>
```

La aplicación web tiene que firmar la petición SOAP utilizando WS-Security para dar el DSS la seguridad de que solamente la aplicación web original será la que descargue el documento firmado.

La aplicación web puede también cancelar instantáneamente el token de seguridad con un elemento <wst:RequestSecurityToken>.

Y al final el servidor DSS retorna el documento firmado:

```
<soap:Envelope>
    <soap:Body>
        <dss:SignResponse Profile="urn:be:e-contract:dssp:1.0">
            <dss:Result>
                <dss:ResultMajor>
                    urn:oasis:names:tc:dss:1.0:resultmajor:Success
                </dss:ResultMajor>
                <dss:ResultMinor>
    urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
                </dss:ResultMinor>
            </dss:Result>
            <dss:OptionalOutputs>
                <dss:DocumentWithSignature>
                    <dss:Document ID="doc1">
                        <dss:Base64Data MimeType="...">
                            the signed document
                        </dss:Base64Data>
                    </dss:Document>
                </dss:DocumentWithSignature>
                <wst:RequestSecurityTokenResponseCollection>
                    <wst:RequestSecurityTokenResponse>
                        <wst:RequestedTokenCancelled/>
                    </wst:RequestSecurityTokenResponse>
                </wst:RequestSecurityTokenResponseCollection>
            </dss:OptionalOutputs>
            <dss:SignatureObject>
                <dss:SignaturePtr WhichDocument="doc1"/>
            </dss:SignatureObject>
        </dss:SignResponse>
    </soap:Body>
</soap:Envelope>
```

Extensiones

SE definen varias extensiones en la ejecución del protocolo.

Identidad del firmante

La aplicación web no siempre puede pre-determinar que archivo de identidad será utilizado por el usuario final para firmar el documento. La aplicación web puede por ejemplo verificar el documento firmado a través un protocolo de verificación del DSS. Se permite utilizar la opción de entrada <dss:ReturnSignerIdentity> y su correspondiente opción de salida <dss:SignerIdentity> dentro del contexto del POST del Navegador. Estos elementos fueron definidos originalmente dentro del contexto de la verificación de firma.

La presencia del <dss:ReturnSignerIdentity> instruirá al servidor DSS a retornar un <dss:SignerIdentity> como parte del proceso de creación de la firma.

Localización

La aplicación web puede incluir el parámetro opcional <dss:Language> para indicar la localización a ser utilizada por el servidor DSS, como parte de la petición POST del Navegador.

SOAP con documentos adjuntos

Para documentos de gran tamaño, la aplicación web y el servicio DSS pueden utilizar SOAP with attachements (SwA). Para documentos aun más grandes, esto puede tener un impacto positivo en el performance del servicio DSS. En el caso del elemento <dss:Document> se vería así:

```
<dss:Document ID="doc1">
    <dss:AttachmentReference MimeType="..." AttRefURI="cid:...">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>...</ds:DigestValue>
    </dss:AttachmentReference>
</dss:Document>
```

El valor de DigestValue tiene que estar siempre presente dado el hecho de que se firma el cuerpo del mensaje SOAP.

El servidor DSS deberá utilizar solamente SwA para la descarga de documentos cuando la aplicación web utiliza SwA para subir los documentos.

Verificación de Firma

Para verificación de firmas también se permite la utilización de SwA para mejorar el performance.

La solicitud de verificación de firma se muestra a continuación:

```
<soap:Envelope>
    <soap:Body>
        <dss:VerifyRequest Profile="urn:be:e-contract:dssp:1.0">
            <dss:OptionalInputs>
                <vr:ReturnVerificationReport>
                    <vr:IncludeVerifier>false</vr:IncludeVerifier>
                    <vr:IncludeCertificateValues>
                        true
                    </vr:IncludeCertificateValues>
                </vr:ReturnVerificationReport>
            </dss:OptionalInputs>
            <dss:InputDocuments>
                <dss:Document ID="document-id">
                    <dss:Base64Data MimeType="text/plain">
                        ...
                    </dss:Base64Data>
                </dss:Document>
            </dss:InputDocuments>
        </dss:VerifyRequest>
    </soap:Body>
</soap:Envelope>
```

Se utiliza el perfil de OASIS para Reportes de Verificación de firmas.

La correspondiente respuesta SOAP se muestra a continuación:

```
<soap:Envelope>
    <soap:Body>
        <dss:Response RequestID="the original request id">
            <dss:Result>
                <dss:ResultMajor>
                    urn:oasis:names:tc:dss:1.0:resultmajor:Success
                </dss:ResultMajor>
            </dss:Result>
            <dss:OptionalOutputs>
                <vr:VerificationReport>
                    <vr:IndividualReport>
                        <vr:SignedObjectIdentifier>
                            <vr:SignedProperties>
                                <vr:SignedSignatureProperties>
                                    <xades:SigningTime>
                                        2010-09-13T15:35:49.767+02:00
                                    </xades:SigningTime>
                                </vr:SignedSignatureProperties>
                            </vr:SignedProperties>
                        </vr:SignedObjectIdentifier>
                        <dss:Result>
                            <dss:ResultMajor>
                                urn:oasis:names:tc:dss:1.0:resultmajor:Success
                            </dss:ResultMajor>
                        </dss:Result>
                        <vr:Details>
                            <vr:IndividualCertificateReport>
                                <vr:CertificateIdentifier>
                                    <xmldsig:X509IssuerName>
                                       SERIALNUMBER=200612, CN=Citizen CA, C=BE
                                    </xmldsig:X509IssuerName>
                                    <xmldsig:X509SerialNumber>
                                        212676479325590784000842949420577262 32
                                    </xmldsig:X509SerialNumber>
                                </vr:CertificateIdentifier>
                                <vr:Subject>
                                    CN=..., C=BE
                                </vr:Subject>
                                <vr:ChainingOK>
                                    <vr:ResultMajor>
```

```
                                    urn:oasis:names:tc:dss:1.0:detail:valid
                                </vr:ResultMajor>
                            </vr:ChainingOK>
                            <vr:ValidityPeriodOK>
                                <vr:ResultMajor>
                                    urn:oasis:names:tc:dss:1.0:detail:valid
                                </vr:ResultMajor>
                            </vr:ValidityPeriodOK>
                            <vr:ExtensionsOK>
                                <vr:ResultMajor>
                                    urn:oasis:names:tc:dss:1.0:detail:valid
                                </vr:ResultMajor>
                            </vr:ExtensionsOK>
                            <vr:CertificateValue>
                                base64 encoded signer certificate
                            </vr:CertificateValue>
                            <vr:SignatureOK>
                                <vr:SigMathOK>
                                    <vr:ResultMajor>
                                    urn:oasis:names:tc:dss:1.0:detail:valid
                                    </vr:ResultMajor>
                                </vr:SigMathOK>
                            </vr:SignatureOK>
                            <vr:CertificateStatus>
                                <vr:CertStatusOK>
                                    <vr:ResultMajor>
                                    urn:oasis:names:tc:dss:1.0:detail:valid
                                    </vr:ResultMajor>
                                </vr:CertStatusOK>
                            </vr:CertificateStatus>
                        </vr:IndividualCertificateReport>
                    </vr:Details>
                </vr:IndividualReport>
            </vr:VerificationReport>
        </dss:OptionalOutputs>
    </dss:Response>
</soap:Body>
</soap:Envelope>
```

La firma es identificada unívocamente por el elemento <xades:igningTime>. Es certificado del firmante es entregado en el elemento <vr:CertificateValue>.

Esquema XML

El esquema XML para el protocolo del Digital Signature Service se muestra a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema   xmlns="http://www.w3.org/2001/XMLSchema"   targetNamespace="urn:be:e-
contract:dssp:1.0"
    elementFormDefault="qualified" attributeFormDefault="unqualified"
    xmlns:tns="urn:be:e-contract:dssp:1.0" xmlns:ds="http://www.w3.org/2000/09/
xmldsig#"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <import namespace="http://www.w3.org/2000/09/xmldsig#"
        schemaLocation="xmldsig-core-schema.xsd" />

   <element name="PendingRequestContext" type="tns:PendingRequestContextType" />

    <complexType name="PendingRequestContextType">
        <sequence>
            <element ref="ds:Signature" />
        </sequence>
        <attribute name="ID" type="xs:ID" use="required" />
        <attribute name="IssueInstant" type="xs:dateTime" use="required" />
        <attribute name="Destination" type="xs:anyURI" use="required" />
    </complexType>

    <element name="SignResponseContext" type="tns:SignResponseContextType" />

    <complexType name="SignResponseContextType">
        <sequence>
            <element ref="ds:Signature" />
        </sequence>
        <attribute name="ID" type="xs:ID" use="required" />
        <attribute name="IssueInstant" type="xs:dateTime" use="required" />
        <attribute name="Destination" type="xs:anyURI" use="required" />
        <attribute name="InResponseTo" type="xs:NCName" use="required" />
    </complexType>

</schema>
```