

**AMBIENTE DE PRUEBAS
SERVICIO FIRMADOR**

**FIRMADOR, VALIDADOR Y
AUTENTICADOR**

KNOWLEDGE BASE

VERSION 1.11



PF-FVA

Contenido

Introducción	4
Términos empleados.....	4
1. Convertir un documento XML en String a Bytes	5
2. Convertir un documento XML en Bytes a String	5
3. Convertir un documento XML en Bytes a XmlDocument	5
4. Cálculo del hash del documento	6
5. Tipo de dato requerido para el hash del documento al enviar una solicitud de Firma al Servicio Firmador	6
6. Configuración para que la aplicación utilice TLS 1.2	6
7. Configuración de un servicio web WCF como HTTPS	7
8. Configuración del certificado de agente electrónico en el IIS	7
9. Solución al error HTTP Error 403.7 – Forbidden al probar el servicio en un navegador web 10	
10. Pasos para probar el servicio de notificación desde un navegador web.....	11
11. Revisión de la jerarquía de certificados instalada.	11
12. Configuración el servicio notificador para que realice las validaciones de certificado ...	15
13. Implementación del validador del certificado cliente para el servicio de notificación ...	17
14. Solución al error “The remote server returned an unexpected response: (413) Request Entity Too Large”, cuando aparece en el trace del WCF o en la bitácora central del Sinpe, en el servicio de notificación	19
15. Solución al error: “Error al realizar la solicitud HTTP”. “Esto puede deberse a que el certificado del servidor no está configurado correctamente en HTTP.SYS en el caso HTTPS. La causa puede ser también una falta de coincidencia del enlace de seguridad entre el cliente y el servidor”	21
16. Pasos para generar la clase de “ResultadoDeSolicitud” a partir del archivo “ResultadoDeSolicitud_WCF.wsdl” o “ResultadoDeSolicitud_WS.wsdl”.	22
17. Documento Cofirmado y Contrafirmado.	25
18. Solución al error: “System.ServiceModel.Security.SecurityNegotiationException: No se pudo establecer una relación de confianza para el canal seguro SSL/TLS con la autoridad ". ---> System.Net.WebException: Se ha terminado la conexión: No se puede establecer una relación de confianza para el canal seguro SSL/TLS. ---> System.Security.Authentication.AuthenticationException: El certificado remoto no es válido según el procedimiento de validación.”	25
19. Solución al error Hash invalido en la notificación de una firma	27
20. Solución al error 403.16 al invocar un web service o un WCF.....	27
21. Solución al error 403.4 al invocar un web service o un WCF.....	28

22.	Solución al error "The message with Action 'http://tempuri.org/ValidadorDeDocumento/ValideElServicio' cannot be processed at the receiver, due to a ContractFilter mismatch at the EndpointDispatcher. This may be because of either a contract mismatch (mismatched Actions between sender and receiver) or a binding/security mismatch between the sender and the receiver"	29
23.	Solución al error 403.13 al invocar un web service o un WCF.....	30
24.	Solución al error SecureChannelFailure al configurar la identidad de marca en Central Directo	30
25.	Solución al error 403 al invocar un web service o un WCF.....	31
26.	Solución al error "StatusCode Forbidden" al configurar la identidad de marca en Central Directo	32
27.	Pasos para generar la clase "ResultadoDeSolicitud" a partir del archivo "ResultadoDeSolicitud.json"	32
28.	Configurar desde LINUX el balanceador de Oracle para configuración SSL/TLS del servicio de notificación.....	35

Introducción

Durante la fase de utilización del ambiente de pruebas por parte de las entidades financieras, se han generado consultas, que, en conjunto con el equipo de expertos del Banco Central de Costa Rica, se han ido solventando. Esto como parte del apoyo que se les brinda a las entidades para lograr una integración de los sistemas de las mismas con el servicio centralizado de Firma Digital.

Es de este proceso que se ve la necesidad de crear un documento con la base del conocimiento (Knowledge Base), el cual puede ser solicitado al Centro de Operaciones del SINPE, por las entidades que se encuentran interesadas en utilizar el ambiente de pruebas. Se debe tomar en cuenta que en este documento se definen consultas técnicas, las cuales están orientadas a programadores que utilicen el .Net Framework y realicen sus desarrollos en el lenguaje de programación Visual Basic .NET(VB.NET). Dichas respuestas técnicas, son una propuesta de cómo se puede realizar, por lo que no se debe tomar como si fuese el único camino a seguir.

Términos empleados

- Para los fines del presente documento, se entenderá por:
 - ☐ BCCR: Banco Central de Costa Rica.
 - ☐ SINPE: Sistema Nacional de Pagos Electrónicos.
 - ☐ GAUDI: Gestor de Autenticaciones Digitales.
 - ☐ COS: Centro de Operaciones del SINPE.
 - ☐ KB: Base de Conocimiento (knowledge base)

1. Convertir un documento XML en String a Bytes

El siguiente es un ejemplo de como se puede realizar la conversión:

```
Function ConviertaXmlABinario(elXMLString As String) As Byte()  
    Dim elDocumento As Byte()  
    elDocumento = System.Text.Encoding.UTF8.GetBytes(elXMLString)  
  
    Return elDocumento  
End Function
```

2. Convertir un documento XML en Bytes a String

El siguiente es un ejemplo de como se puede realizar la conversión:

```
Function ConviertaBinarioAUnaCadenaXml(elDocumento() As Byte) As String  
    Dim laCadenaXml As String  
    laCadenaXml = System.Text.Encoding.UTF8.GetString(elDocumento)  
  
    Return laCadenaXml  
End Function
```

3. Convertir un documento XML en Bytes a XmlDocument

El siguiente es un ejemplo de como se puede realizar la conversión:

```
Function ConviertaBinarioAXml(elDocumento As Byte()) As XmlDocument  
    Dim elDocumentoXML As XmlDocument  
    Dim laCadenaXml As String  
  
    laCadenaXml = ConviertaBinarioAUnaCadenaXml(elDocumento)  
    elDocumentoXML = ConviertaCadenaXmlAXml (laCadenaXml)  
  
    Return elDocumentoXML  
End Function  
  
Private Function ConviertaBinarioAUnaCadenaXml(elDocumento() As Byte) As String  
    Dim laCadenaXml As String  
    laCadenaXml = System.Text.Encoding.UTF8.GetString(elDocumento)  
  
    Return laCadenaXml  
End Function  
  
Private Function ConviertaCadenaXmlAXml(laCadenaXml As String) As XmlDocument  
    Dim elDocumentoXml As New XmlDocument()  
    elDocumentoXml.LoadXml(laCadenaXml)  
  
    Return elDocumentoXml  
End Function
```

4. Cálculo del hash del documento

El hash del documento se debe de calcular luego de convertirlo a Bytes (binario), utilizando alguno de los algoritmos descritos en el Estándar Electrónico – Firmador, Validador y Autenticador.

El siguiente es un ejemplo de cómo calcular el hash del documento, utilizando el algoritmo SHA256:

```
Public Function GenereElHashDelDocumento(elDocumento As Byte()) As Byte()  
    Dim elAlgoritmoDeHash As HashAlgorithm  
    elAlgoritmoDeHash = New SHA256Managed()  
  
    Return elAlgoritmoDeHash.ComputeHash(elDocumento)  
End Function
```

5. Tipo de dato requerido para el hash del documento al enviar una solicitud de Firma al Servicio Firmador

El tipo de formato del hash debe de ser base64Binary.

```
- <xs:complexType name="SolicitudDeFirma">  
  - <xs:sequence>  
    <xs:element name="CodNegocio" type="xs:int" minOccurs="0"/>  
    <xs:element name="Documento" type="xs:base64Binary" nillable="true" minOccurs="0"/>  
    <xs:element name="FechaDeReferenciaDeLaEntidad" type="xs:dateTime" minOccurs="0"/>  
    <xs:element name="HashDocumento" type="xs:base64Binary" nillable="true" minOccurs="0"/>
```

6. Configuración para que la aplicación utilice TLS 1.2

Existen diversas formas de hacer que las aplicaciones que se están construyendo se comuniquen utilizando la versión de TLS 1.2, seguidamente se detallarán en orden de recomendación, siendo la primera la más recomendada y la última la menos recomendada:

- a. Utilice una versión de Framework que utilice por defecto la versión TLS 1.2. Al utilizar el Framework de .Net para los desarrollos, se debe tener cuidado con la versión configurada, debido a que dependiendo de esto la aplicación utilizará TLS 1.2 por defecto o podría no tener compatibilidad. Lo recomendable es utilizar la versión del Framework más reciente.
- b. Configurar a nivel de código para obligar a la aplicación a utilizar TLS 1.2. Esto es válido pero no es recomendable, si en un futuro se determina que la versión de TLS 1.2 tiene una vulnerabilidad que se solventa utilizando una versión superior del protocolo, se deberá ingresar al código fuente de la aplicación para actualizarla y colocar la versión con el cambio en producción. Lo recomendable es no obligar a la aplicación, esta se debe colocar en un servidor que este configurado para utilizar TLS 1.2, dejando la responsabilidad de la utilización de protocolos seguros al sistema operativo.

Ejemplo de cómo obligar el uso de TLS 1.2:

```
Public Function ServicioDisponibleFirmador() As Boolean  
Implements IServicioEntidadExterna.ServicioDisponibleFirmador  
    System.Net.ServicePointManager.SecurityProtocol =  
        SecurityProtocolType.Tls12  
    Dim elClienteFirmador As New SI.Firmador.FirmadorClient
```

```
Return elClienteFirmador.ValideElServicio()  
End Function
```

7. Configuración de un servicio web WCF como HTTPS

Para que se pueda establecer una comunicación segura se requiere que los servicios web creados por la entidad utilicen HTTPS, esto se realiza a nivel del Web.Config del servicio. En seguida de muestra el ejemplo de cómo realizar la configuración de un servicio que utiliza la tecnología WCF:

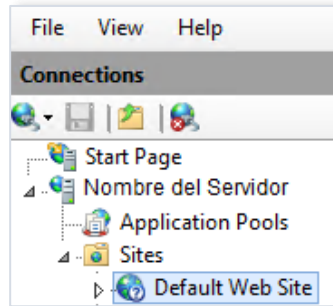
En este ejemplo se configura el servicio con un binding HTTP que utiliza seguridad de transporte, en la dirección que se indica en el Endpoint del servicio es donde se indica que debe ser HTTPS.

```
<system.serviceModel>  
  <behaviors>  
    <serviceBehaviors>  
      <behavior name="behavoirDeEjemplo">  
        <serviceMetadata httpsGetEnabled="true" httpGetEnabled="true"/>  
        <serviceDebug includeExceptionDetailInFaults="false"/>  
      </behavior>  
    </serviceBehaviors>  
  </behaviors>  
  <bindings>  
    <wsHttpBinding>  
      <binding name="webHttpsBindingDeEjemplo">  
        <security mode="Transport" />  
        <transport clientCredentialType="Certificate"/>  
      </binding>  
    </wsHttpBinding>  
  </bindings>  
  <services>  
    <service behaviorConfiguration="behavoirDeEjemplo"  
      name="ServicioDeEjemplo">  
      <endpoint address="https://localhost:8888/ServicioDeEjemplo.svc"  
        binding="wsHttpBinding"  
        bindingConfiguration="webHttpsBindingDeEjemplo"  
        contract="Ejemplo.Servicios.Wcf.ServicioDeEjemplo">  
      </endpoint>  
    </service>  
  </services>  
  <serviceHostingEnvironment multipleSiteBindingsEnabled="true"  
    aspNetCompatibilityEnabled="true"/>  
</system.serviceModel>
```

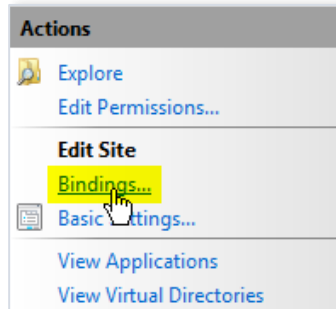
8. Configuración del certificado de agente electrónico en el IIS

Esto se puede realizar siguiendo los siguientes pasos:

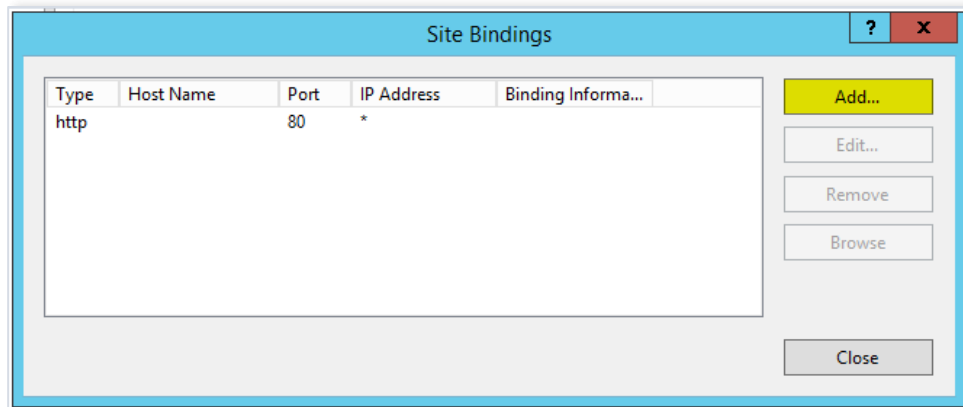
- Abra el IIS.
- Seleccione el sitio dentro del cual se encuentra el servicio a asegurar, en este caso se utilizará "Default Web Site".



c) Del panel derecho, en el grupo Action, seleccione Bindings.



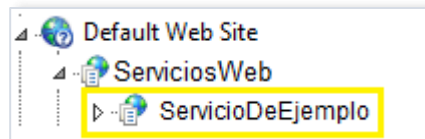
d) Estando en "Site Bindings" de clic sobre el botón "Add".



e) En la ventana que se despliega, realice la configuración como se muestra en la siguiente imagen:

Nota: El certificado SSL es el certificado de agente electrónico que se encuentra instalado en el servidor web.

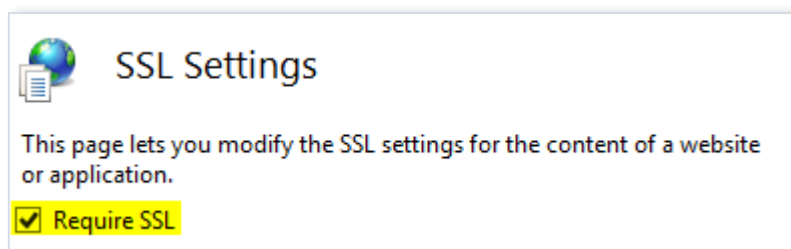
- f) De clic en el botón “Ok”.
- g) Busque y seleccione dentro del sitio, el servicio al que desea activarle la seguridad.



- h) Dentro de “Features View”, de doble clic sobre la opción “SSL Settings”.

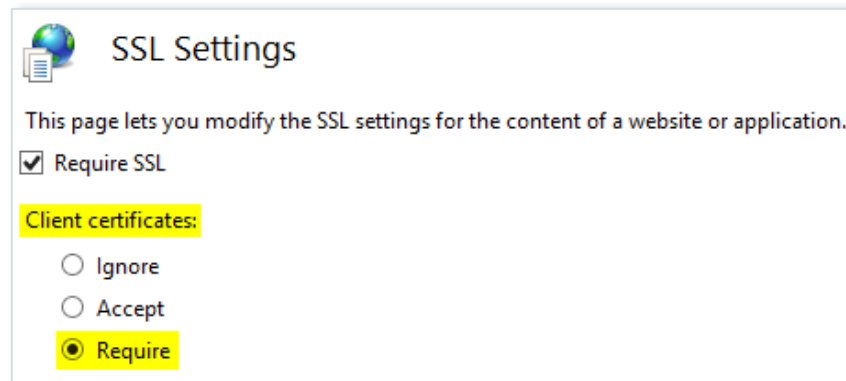


- i) Dentro de “SSL Settings” seleccione el check “Require SSL”.



- j) Se puede configurar SSL para que sólo los clientes que cuenten con un certificado realicen llamados al servicio (opción Require), esto permitirá la validación de dicho certificado donde se verificará:
 - a. El vencimiento: se revisa que el certificado no se encuentre vencido.
 - b. Revocación: el servidor deberá tener acceso al punto de distribución de CRLs que indica el certificado.

- c. Pertenencia a una jerarquía de confianza: la jerarquía se indica en el tab llamado “Ruta de certificación”, los certificados indicados en esa sección deben estar instalados en el servidor web.



Nota:

-La opción “Accept” indica que si viene un certificados se valida y sino igual se puede utilizar el servicio.

-La opción “Ignore” indica que si viene un certificado no se va a realizar la validación.

9. Solución al error HTTP Error 403.7 – Forbidden al probar el servicio en un navegador web

Este error se presenta cuando configuración SSL es “Require SSL, Client certificates: Require”, debido a que el servicio está esperando un certificado de autenticación del cliente y como no recibe ninguno.

HTTP Error 403.7 - Forbidden

The page you are attempting to access requires your browser to have a Secure Sockets Layer (SSL) client certificate that the Web server recognizes.

Most likely causes:

- The page you are attempting to access requires an SSL client certificate.
- You are browsing to the page using HTTP.
- The client certificate has expired or the effective time has not been reached.
- The root certificate (the Certificate Authority certificate) of the client certificate issuing server is not installed on the Web server.

Things you can try:

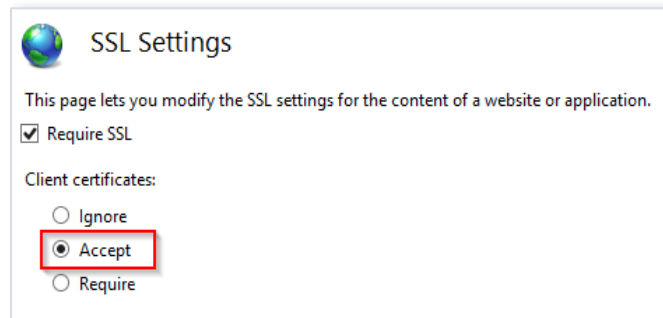
- Contact the site administrator to obtain a valid client certificate for the Web site.
- Try browsing to the page using HTTPS.
- If you have a client certificate installed, check if it has expired or if the effective time has not been reached.
- Verify that the root certificate is installed on the Web server.

Detailed Error Information:

Module	IIS Web Core	Requested URL	https://[redacted]
Notification	BeginRequest	Physical Path	[redacted]
Handler	svc-Integrated-4.0	Logon Method	Not yet determined
Error Code	0x80070005	Logon User	Not yet determined

10. Pasos para probar el servicio de notificación desde un navegador web

- a. Cambie **momentáneamente** la configuración SSL a “Require SSL, Client certificates: Accept”.

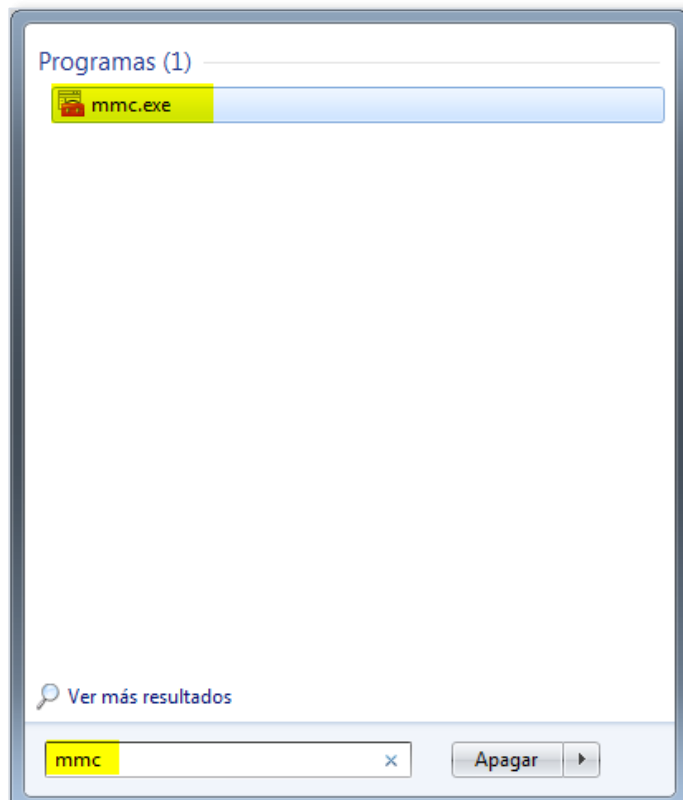


- b. Realice la prueba del servicio desde un navegador web.
- c. Vuelva a configurar el SSL en “Require”. Este paso es indispensable, debido a que es la configuración que se debe de tener siempre.

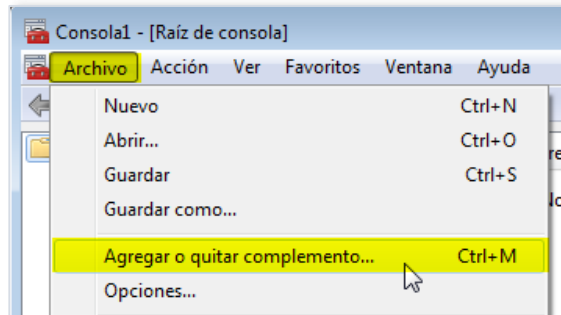
11. Revisión de la jerarquía de certificados instalada.

Para verificar la instalación de la jerarquía, debe seguir los siguientes pasos:

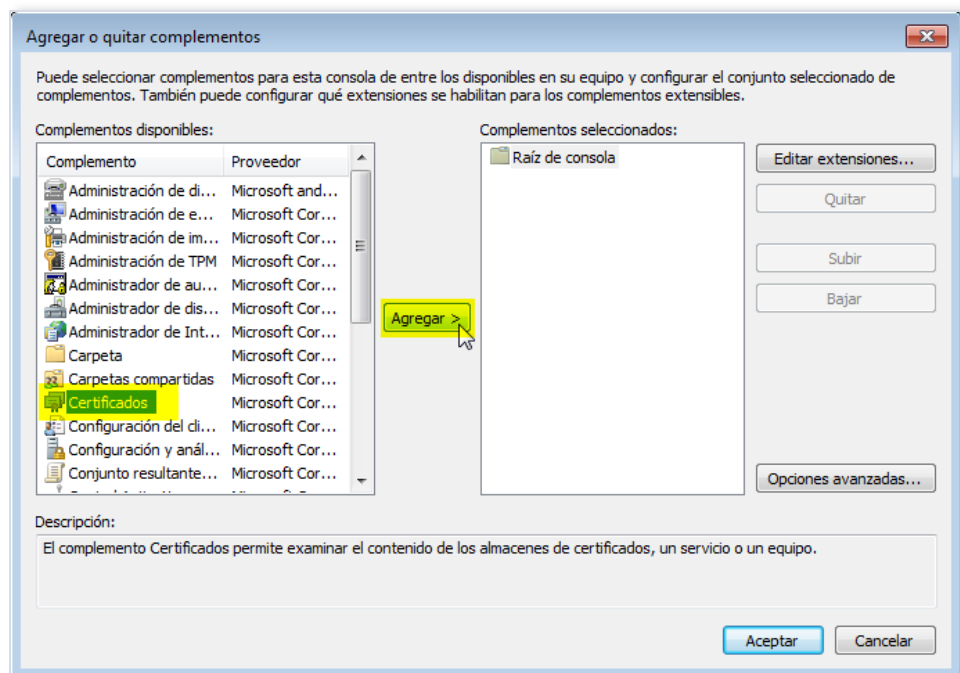
- a. Busque la aplicación “mmc.exe” con el buscador de Windows, la cual corresponde al almacén de certificados.



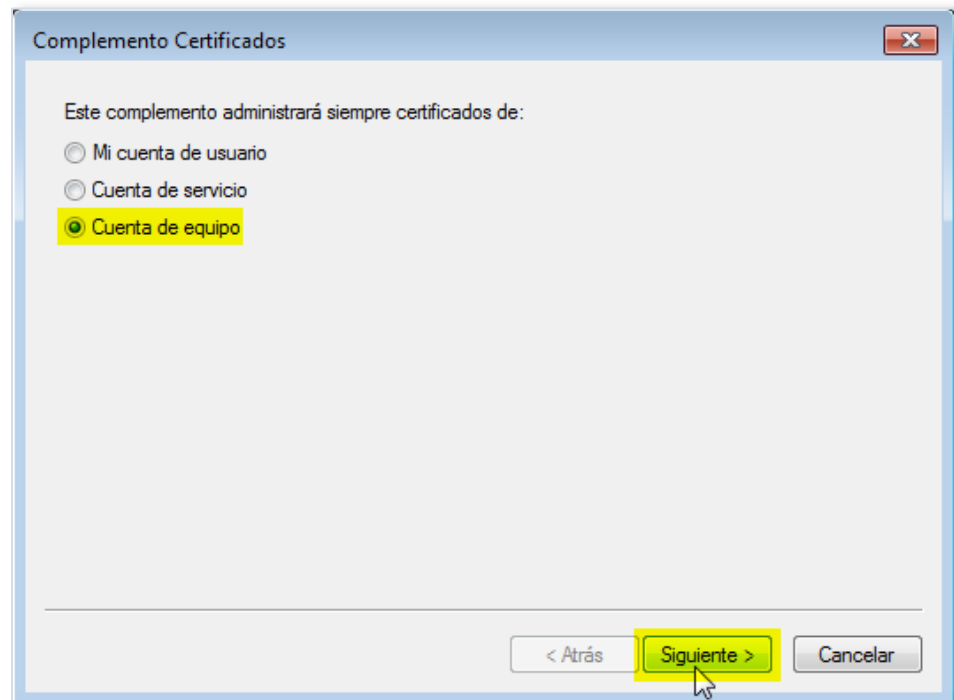
- b. En la consola que se despliega, seleccione “Agregar o quitar complemento”.



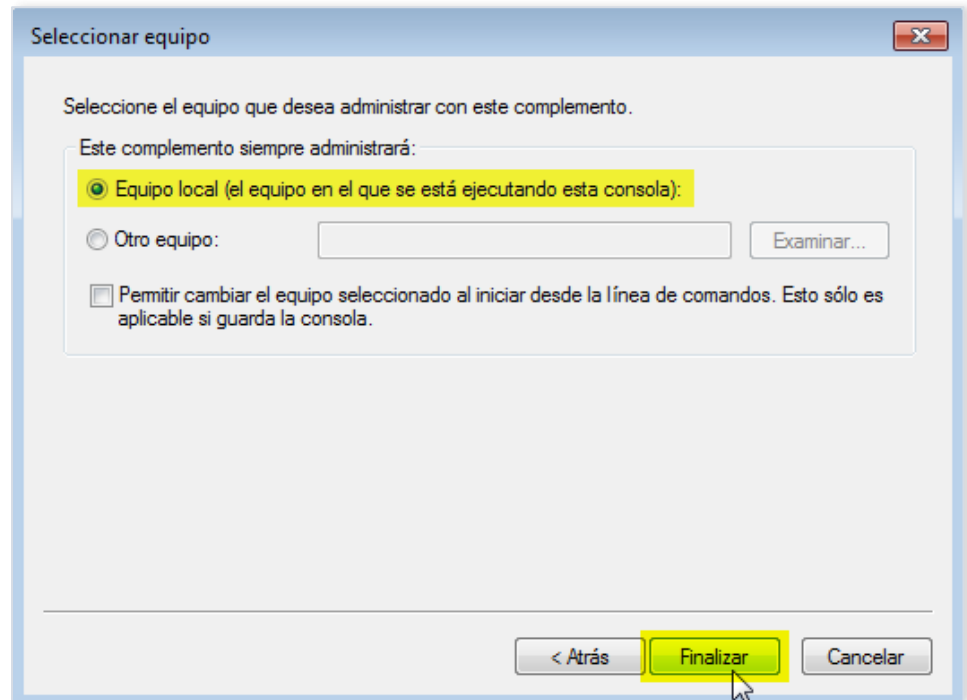
- c. En el panel “Complementos disponibles”, seleccione “Certificados” y luego de clic sobre el botón “Agregar”.



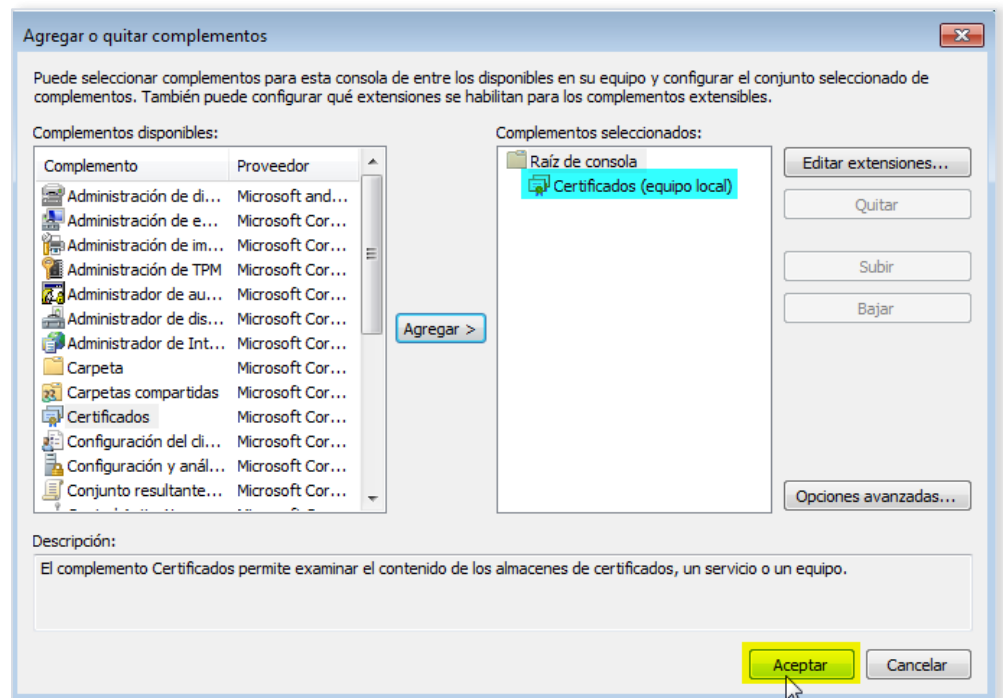
- d. En la ventana que se despliega, seleccione “Cuenta de equipo”.



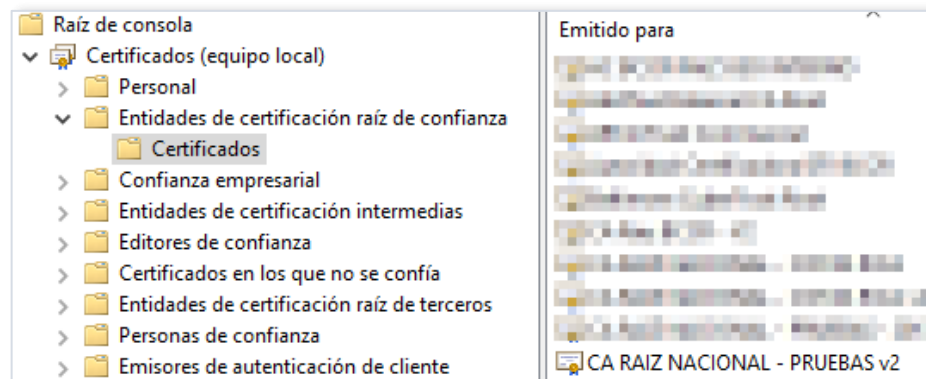
- f. En la ventana “Seleccionar equipo”, elija “Equipo local” y luego de clic sobre el botón “Finalizar”.



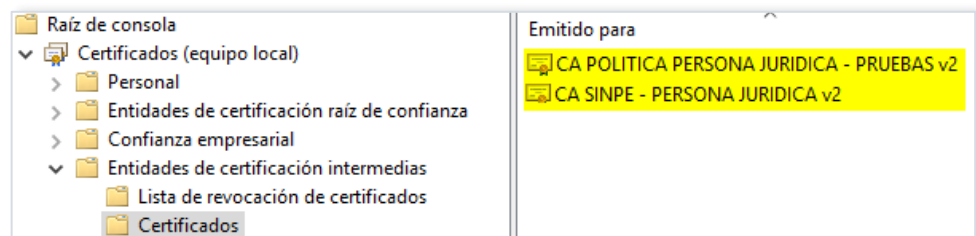
- g. Verifique que en los “Complementos seleccionados” se encuentre el que se escogió anteriormente y seleccione el botón “Aceptar”



- h. Dentro del almacén de certificados vaya a “Certificados” → “Entidades de certificación raíz de confianza” → “Certificados”.
- i. Verifique que se encuentre el certificado “CA RAIZ NACIONAL - PRUEBAS v2”.



- j. Luego diríjase a “Certificados” → “Entidades de certificación intermedias” → “Certificados”.
- k. Verifique que se encuentren los certificados “CA POLITICA PERSONA JURIDICA - COSTA RICA v2” y “CA SINPE - PERSONA JURIDICA v2”.



12. Configuración el servicio notificador para que realice las validaciones de certificado

El archivo de configuración del servicio notificador que desarrolla la entidad, se puede diseñar como se muestra en el siguiente ejemplo, lo cual le permitirá contar con un servicio seguro que valida que sólo es invocado por el BCCR:

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
    <add key="ElSujetoDelCertificado" value="CN=BANCO CENTRAL DE COSTA RICA (AGENTE ELECTRONICO), O=PERSONA JURIDICA, C=CR, SERIALNUMBER=CPJ-4-000-004017"/>
    <add key="ThumbprintDeLaRaiz" value="68A24D2B2CB5C4CE9FE300E3B6FCD1DFEEB9C311" />
  </appSettings>
  <system.web>
    <compilation debug="true" strict="false" explicit="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2"/>
  </system.web>
  <system.serviceModel>
    <services>
```

```

<service
name="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WCF.Re
sultadoDeSolicitud"
behaviorConfiguration="ComportamientoDelServicioDeLaEntidadPersonaliz
ado">
  <endpoint address=""
binding="wsHttpBinding"
bindingConfiguration="EnlaceConSeguridadDeTipoTransporte"
contract="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WC
F.ResultadoDeSolicitud"
listenUri="" />
</endpoint>
</service>
</services>
<behaviors>
  <endpointBehaviors>
    <behavior name="BccrServidor_EndpointBehavior">
      <dataContractSerializer maxItemsInObjectGraph="2147483646" />
    </behavior>
  </endpointBehaviors>
  <serviceBehaviors>
    <behavior name="ComportamientoDelServicioDeLaEntidadPersonalizado">
      <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />
      <serviceCredentials>
        <clientCertificate>
          <authentication certificateValidationMode="Custom"
customCertificateValidatorType="EjemploDeServicioDeNotificacionDePers
onaFisica.ValidadorDeCertificados,
EjemploDeServicioDeNotificacionDePersonaFisica" />
        </clientCertificate>
      </serviceCredentials>
    </behavior>
  </serviceBehaviors>
</behaviors>
<bindings>
  <wsHttpBinding>
    <binding name="EnlaceConSeguridadDeTipoTransporte"
maxReceivedMessageSize="28311552">
      <security mode="Transport">
        <transport clientCredentialType="Certificate" />
      </security>
    </binding>
  </wsHttpBinding>
</bindings>
  <serviceHostingEnvironment aspNetCompatibilityEnabled="true"
multipleSiteBindingsEnabled="true" />
</system.serviceModel>
<system.webServer>
  <modules runAllManagedModulesForAllRequests="true" />
  <directoryBrowse enabled="true" />
</system.webServer>
</configuration>

```

13. Implementación del validador del certificado cliente para el servicio de notificación

El validador de certificado se puede realizar como se indica en el siguiente ejemplo, el cual está desarrollado utilizando el lenguaje VB.Net.

```
Imports System.IdentityModel.Selectors
Imports System.IdentityModel.Tokens
Imports System.Security.Cryptography.X509Certificates

Public Class ValidadorDeCertificados
    Inherits X509CertificateValidator

    Public Overrides Sub Validate(certificate As X509Certificate2)
        If (certificate Is Nothing) Then
            EventLog.WriteEntry("Application", "Certificado Vacio")
        End If
        Dim elSujetoDelCertificado As String
        Dim laFechaActual As Date = Date.Now

        Dim elInicioDelCronometro As New TimeSpan(DateTime.Now.Ticks)

        If Not ElCertificadoEstaVigente(certificate.NotBefore, certificate.NotAfter,
            laFechaActual) Then
            EventLog.WriteEntry("Application", "El certificado no esta vigente")
            Throw New SecurityTokenValidationException("Certificado Vencido")
        Else
            EventLog.WriteEntry("Application", "El certificado esta vigente")
        End If

        elSujetoDelCertificado = ObtenerValorDelAtributo("ElSujetoDelCertificado")
        If Not EsValidoElAtributo(elSujetoDelCertificado, certificate.Subject.ToString)
Then
            EventLog.WriteEntry("Application", "Certificado no valido " &
certificate.Subject.ToString)
            Throw New SecurityTokenValidationException("Certificado no valido")
        End If

        If Not VerificarSiRaizEsValida(certificate) Then
            EventLog.WriteEntry("Application", "Certificado Raiz no valido")
            Throw New SecurityTokenValidationException("Certificado Raiz no valido")
        End If

        Dim elCierreDelCronometro As New TimeSpan(DateTime.Now.Ticks)

        Dim elTotalDeTiempo =
            elCierreDelCronometro.Subtract(elInicioDelCronometro).TotalMilliseconds

        EventLog.WriteEntry("Application", "VerificarSiRaizEsValida tardó: " &
            elTotalDeTiempo & " milisegundos.")
        End Sub

        Private Function ElCertificadoEstaVigente(laFechaDeEmision As Date,
            laFechaDeVencimiento As Date,
            laFechaActual As Date) As Boolean

            Dim esVigente As Boolean = False

            If laFechaDeEmision < laFechaActual And laFechaDeVencimiento > laFechaActual Then
```

```

        esVigente = True
    End If

    Return esVigente
End Function

Public Function VerificarSiRaizEsValida(elCertificado As X509Certificate2) As
Boolean

    Dim certificadoRaizCadena As X509Certificate2
    Dim esCertificadoRaizCorrecto As Boolean = False
    Try
        certificadoRaizCadena = ObtenerCertificadoRaiz(elCertificado)

        esCertificadoRaizCorrecto = EsValidoElCertificadoRaiz(certificadoRaizCadena)
    Finally
    End Try
    Return esCertificadoRaizCorrecto
End Function

Public Function ObtenerCertificadoRaiz(elCertificado As X509Certificate2) As
X509Certificate2
    Dim certificadoRaiz As X509Certificate2 = Nothing
    Dim chain As New X509Chain() 'se construye una cadena de certificacion para
obtener el certificado raiz de la misma
    Dim excepcion As System.Exception = Nothing
    Try
        If elCertificado Is Nothing Then
            EventLog.WriteEntry("Application", "No se encontró el certificado para
validar. Favor asigne el certificado a la clase y luego proceda")
        Else
            chain.ChainPolicy.RevocationMode = X509RevocationMode.Offline 'se
realiza la construccion de la cadena mas rapida posible
            chain.ChainPolicy.RevocationFlag = X509RevocationFlag.EntireChain
            chain.ChainPolicy.UrlRetrievalTimeout = New TimeSpan(0, 0, 0)
            chain.ChainPolicy.VerificationFlags = X509VerificationFlags.AllFlags
            chain.Build(elCertificado)
            certificadoRaiz = chain.ChainElements.Item(chain.ChainElements.Count -
1).Certificate 'se obtiene el último elemento de la cadena que debe de corresponder
al elemento raiz

        End If
        If Not excepcion Is Nothing Then
            Throw excepcion
        End If
    Finally
    End Try
    Return certificadoRaiz
End Function

Private Function EsValidoElCertificadoRaiz(certificadoRaiz As X509Certificate2) As
Boolean
    Dim laRaizEsValida As Boolean
    Dim laHuellaDelaRaiz As String = ObtenerValorDelAtributo("ThumbprintDeLaRaiz")
    Dim laHuellaDelCertificadoRaiz As String = certificadoRaiz.Thumbprint

    If (EsValidoElAtributo(laHuellaDelaRaiz, laHuellaDelCertificadoRaiz)) Then 'se
verifica que el certificado obtenido efectivamente sea el certificado raiz esperado

```

```

        laRaizEsValida = True
    Else
        EventLog.WriteEntry("Application", "Determinando que el certificado raiz es
invalido, la huella:" & laHuellaDelCertificadoRaiz)
        laRaizEsValida = False
    End If
    Return laRaizEsValida
End Function

Private Function EsValidoElAtributo(elAtributoEsperado As String, elAtributoObtenido
As String) As Boolean
    Return elAtributoEsperado.Equals(elAtributoObtenido)
End Function

Private Shared Function ObtenerValorDelAtributo(elAtributo As String) As String
    Return ConfigurationManager.AppSettings.Get(elAtributo).ToString()
End Function

End Class

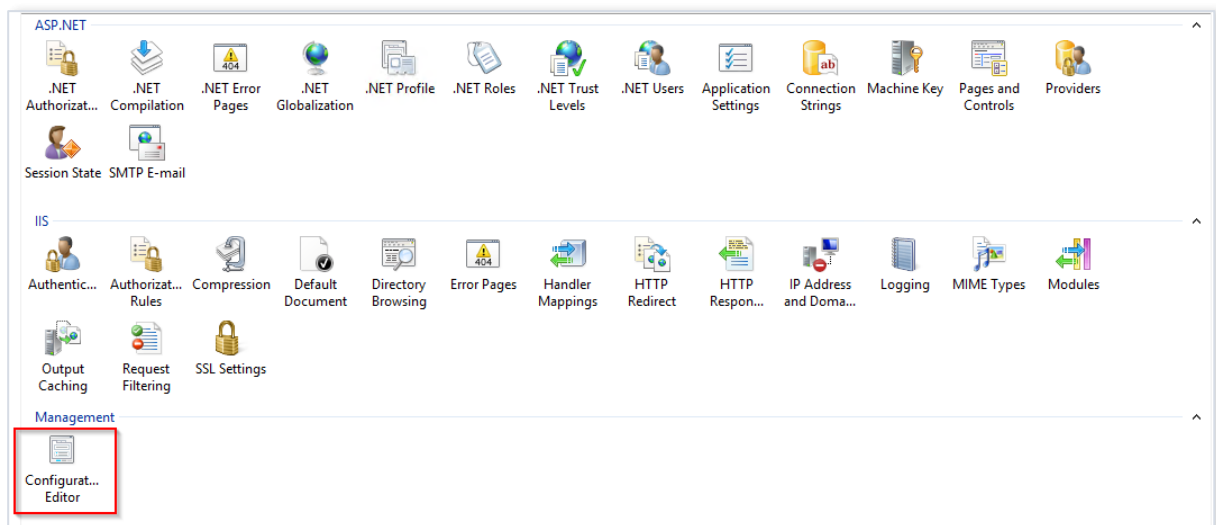
```

14. Solución al error “The remote server returned an unexpected response: (413) Request Entity Too Large”, cuando aparece en el trace del WCF o en la bitácora central del Sinpe, en el servicio de notificación

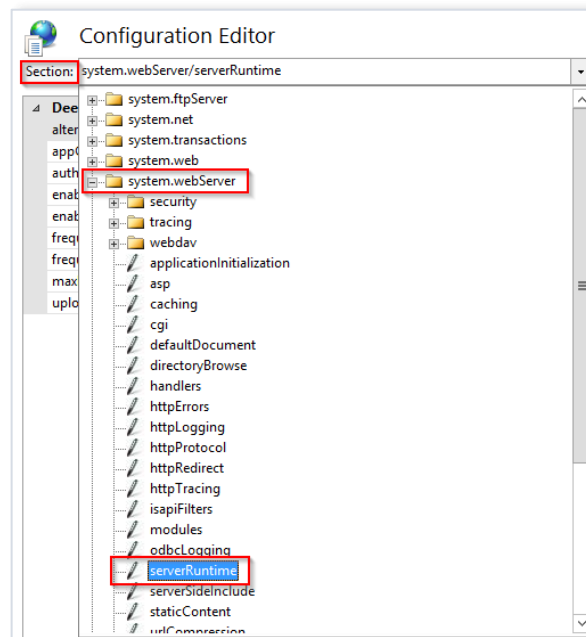
Este error se presenta cuando la entidad rechaza el paquete enviado por el BCCR, el motivo del rechazo es el tamaño del paquete, dado que la entidad tiene configurado a nivel de IIS y de binding del servicio notificador, un tamaño de paquete menor al tamaño del paquete enviado por el BCCR.

Pasos para corregirlo:

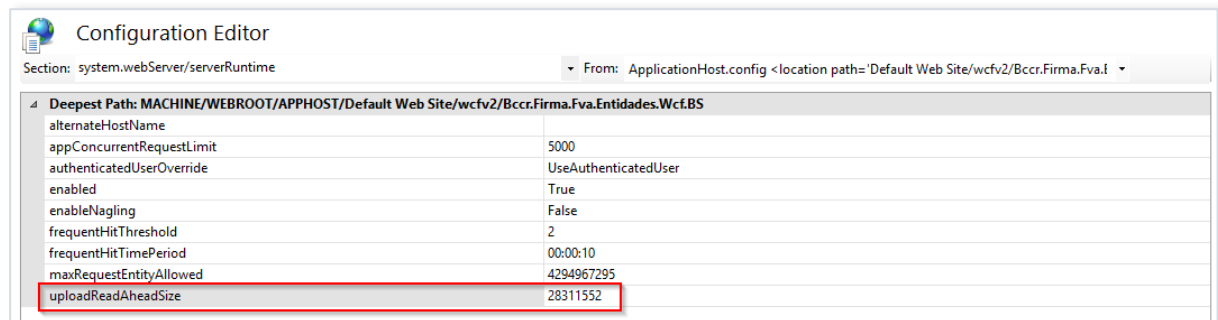
- A. Ajustar el parámetro de tamaño máximo de paquete en el IIS:
 - En el **IIS**, diríjase al directorio virtual donde tiene almacenado el servicio de notificación.
 - En **vista de características**, elegir la opción **Editor de configuración**.



- En la opción **Section**, seleccionar la ruta **system.webServer/serverRuntime**



- Ajustar el valor del parámetro **uploadReadAheadSize**, con el valor **28311552**.



B. Ajustar el parámetro de tamaño máximo de paquete en el **bindingConfiguration** del servicio notificador:

- Identificar el **bindingConfiguration** que utiliza el servicio de notificación.

```
<services>
  <service name="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WCF.ResultadoDeSolicitud"
    behaviorConfiguration="ComportamientoDelServicioDeLaEntidadPersonalizado">
    <endpoint address=""
      binding="wsHttpBinding"
      bindingConfiguration="EnlaceConSeguridadDeTipoTransporte"
      contract="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WCF.ResultadoDeSolicitud"
      listenUri="/">
    </endpoint>
  </service>
</services>
```

- En el **binding**, ajustar el valor del parámetro **maxReceivedMessageSize**, con el valor **28311552**.

```
<bindings>
  <wsHttpBinding>
    <binding name="EnlaceConSeguridadDeTipoTransporte" maxReceivedMessageSize="28311552">
      <security mode="Transport">
        <transport clientCredentialType="Certificate"/>
      </security>
    </binding>
  </wsHttpBinding>
</bindings>
```

15. Solución al error: “Error al realizar la solicitud HTTP”. “Esto puede deberse a que el certificado del servidor no está configurado correctamente en HTTP.SYS en el caso HTTPS. La causa puede ser también una falta de coincidencia del enlace de seguridad entre el cliente y el servidor”.

Este error se puede presentar cuando la entidad está invocando las funcionalidades del servicio Firmador o el BCCR está invocando el servicio de notificación de firma de la entidad. Además, si se levanta el servicio en un navegador se muestra el siguiente error:

No se puede mostrar esta página

Activa TLS 1.0, TLS 1.1 y TLS 1.2 desde Configuración avanzada e intenta conectarte en **https://localhost** de nuevo. Si el error continúa, es posible que este sitio use un protocolo no compatible o un conjunto de cifrado como RC4 ([vínculo para obtener detalles](#)), que no se considere seguro. Ponte en contacto con el administrador del sitio.

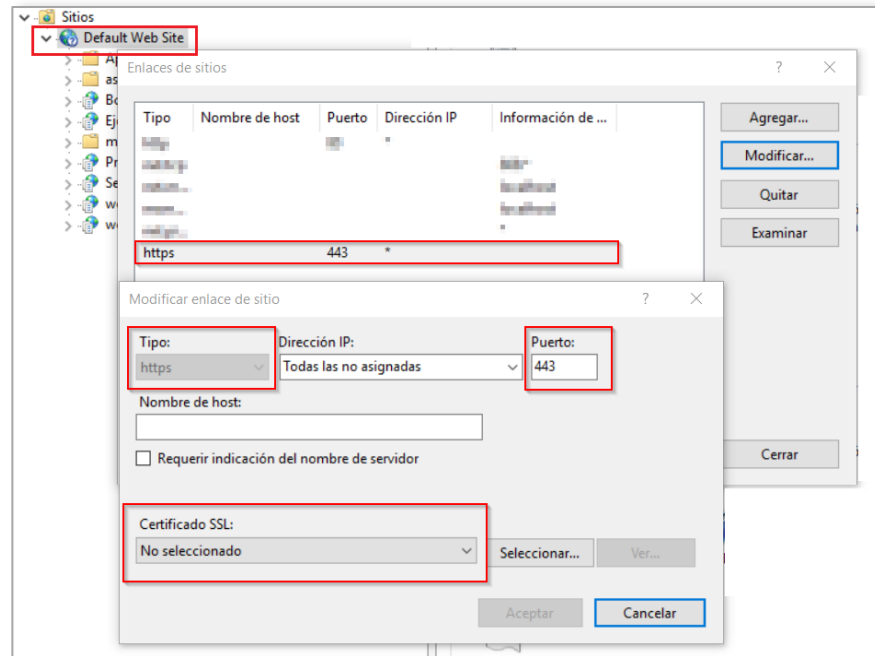
Cambiar configuración

Posibles causas y soluciones:

- El cliente, que intenta acceder a los servicios expuestos del Firmador, no habla TLS 1.2, para solucionarlo consulte el punto [6](#), de este documento.
- El servidor de la entidad no tiene habilitados ciphers para TLS 1.2, para verificar la lista de ciphers que expone el servidor, puede consultar la pregunta 17, del documento de “Preguntas frecuentes del servicio Firmador”, en caso de no exponer ningún cipher para TLS 1.2, revisar las políticas que se aplican al

servidor para que habilite ciphers para TLS 1.2, de los cuales al menos uno coincida con los expuestos por el BCCR, para conocer la lista cipher que expone el BCCR, puede consultar la pregunta 16, del documento de “Preguntas frecuentes del servicio Firmador”.

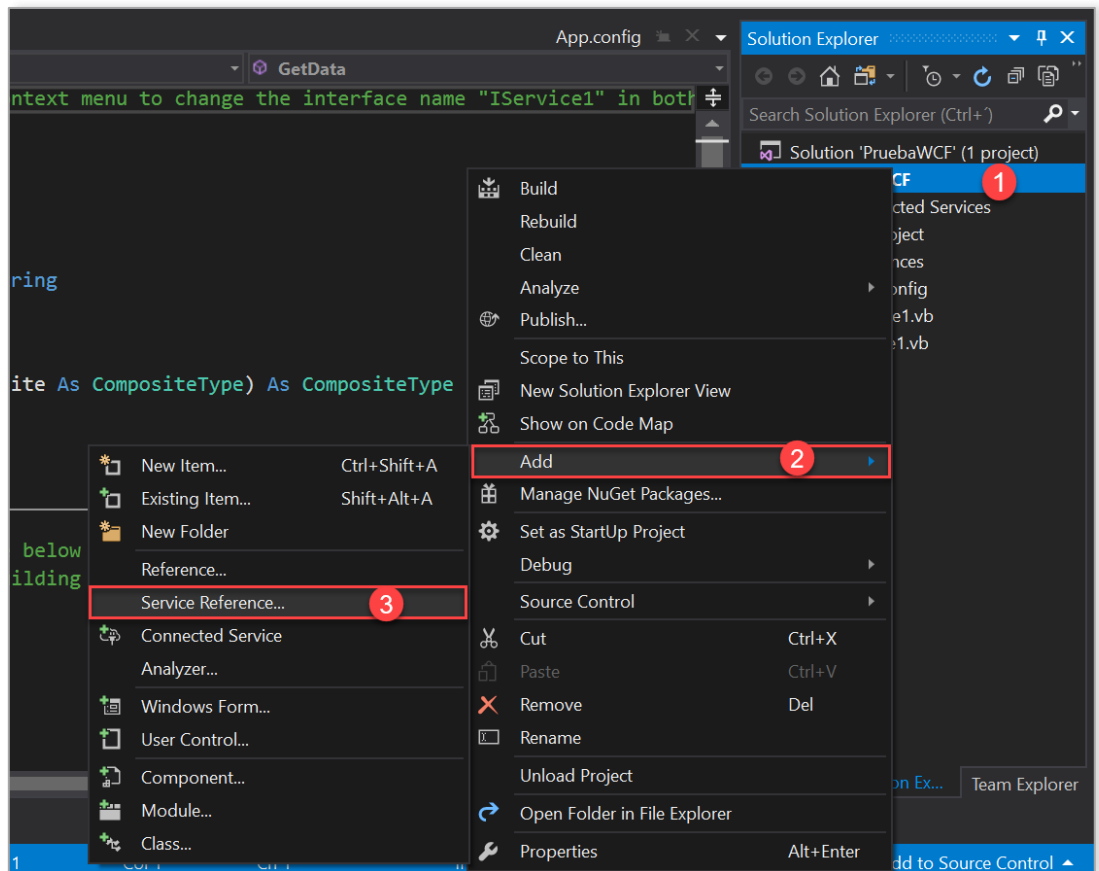
- El sitio que almacena el servicio a consumir no tiene definido, en los bindings (enlaces), un certificado SSL para asegurar el puerto 443 (https).



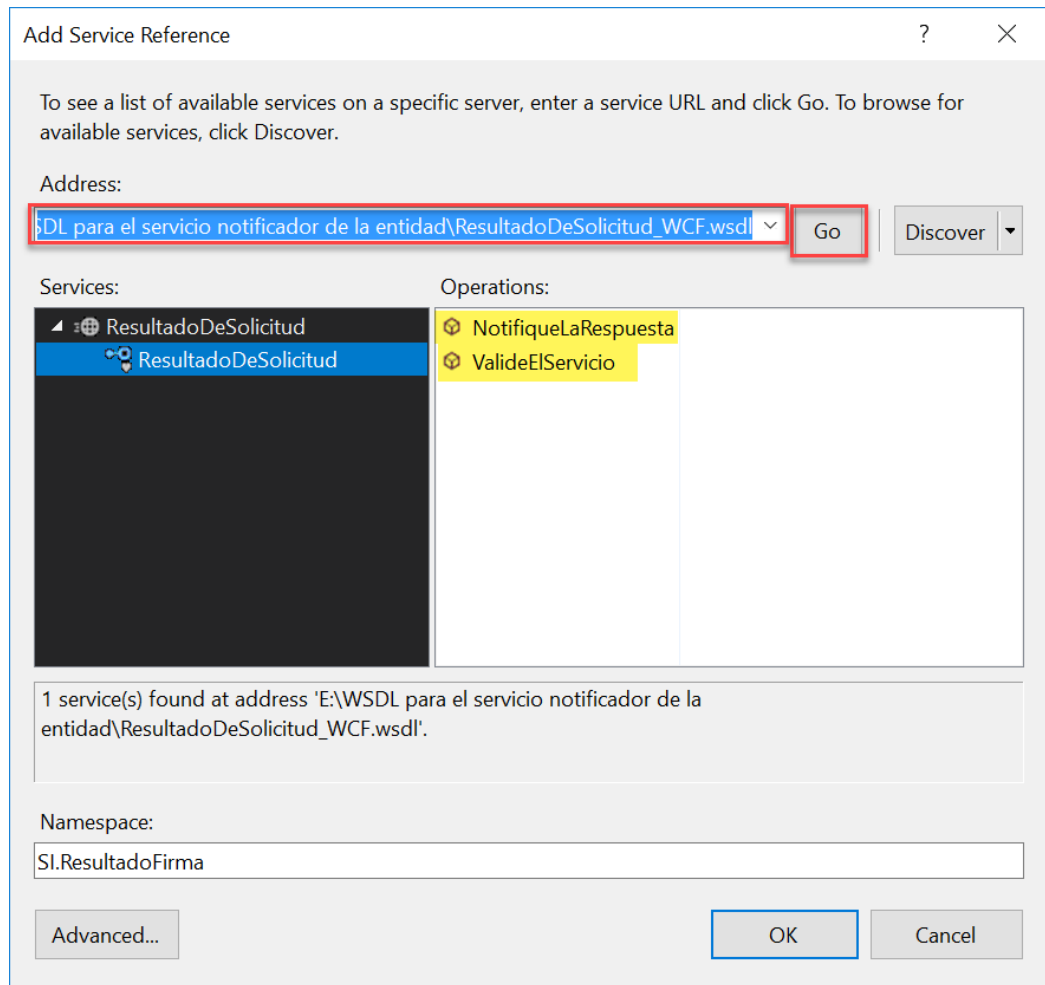
Se debe asegurar el sitio con el certificado de agente electrónico, enviado por el BCCR, puede consultar el punto [8](#), de este documento.

16. Pasos para generar la clase de “ResultadoDeSolicitud” a partir del archivo “ResultadoDeSolicitud_WCF.wsdl” o “ResultadoDeSolicitud_WS.wsdl”.

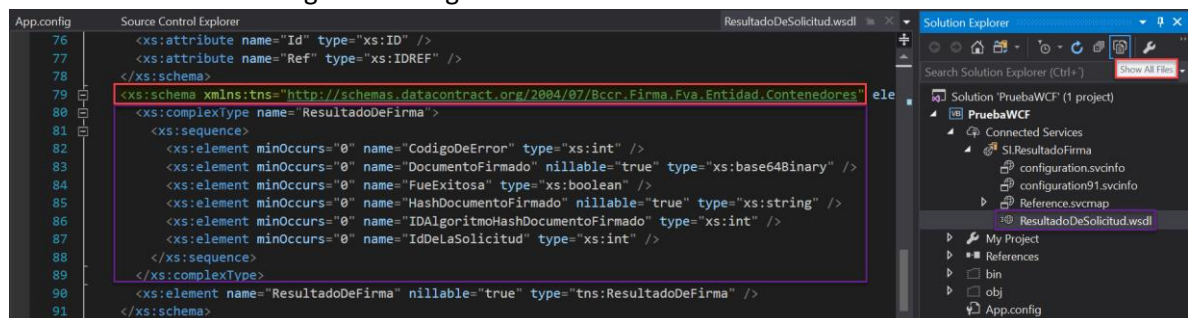
- a. Clic derecho sobre el proyecto en el que desea implementar la clase. Agregar el servicio de referencia (Service Reference)



- b. En el espacio “Address” agregar la ruta donde se encuentra el archivo WSDL, una vez agregada la ruta correctamente dar clic en “Go” debería de aparecer la información como se muestra en la siguiente imagen. Además, se le agrega un nombre significativo al “ServiceReference”.



- c. Clic en “Show All Files”. Desplegar la clase proxy del Service Reference, verificar que el archivo “ResultadoDeSolicitud.wsdl” contenga el objeto “ResultadoDeFirma”, como se muestra en la siguiente imagen.



- d. Es importante señalar que el **parámetro** que recibe el método “NotifiqueLaRespuesta” es de tipo “ResultadoDeFirma” y debe llamarse “elResultado”.

```
Public Sub NotifiqueLaRespuesta(elResultado As ResultadoDeFirma)
```

17. Documento Cofirmado y Contrafirmado.

Un documento cofirmado es cuando se realiza una firma sobre el documento original, puede ser cofirmado n veces.

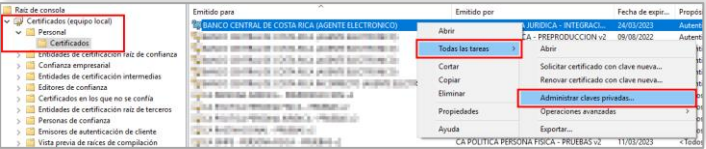
Un documento contrafirmado tiene un tag llamado “CounterSignature” que se agrega al momento de realizar la segunda contrafirma, cada vez que se realice un contrafirma se crea el tag “CounterSignature” dentro del tag “CounterSignature” ya existente.

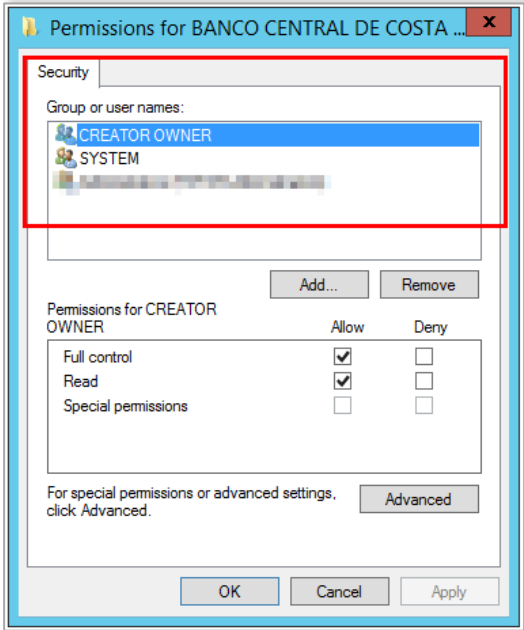
Un documento cofirmado no puede ser contrafirmado; esto se debe a que al tratar de realizar la contrafirma no encuentra el tag “CounterSignature”, además como contiene más de una firma no sabe con cual trabajar, por esta razón falla la contrafirma.

Sin embargo, un documento contrafirmado puede ser cofirmado, debido a que al realizar la cofirma lo que se firma es el documento original, la cofirma no tiene requisitos.

18. Solución al error:

“System.ServiceModel.Security.SecurityNegotiationException: No se pudo establecer una relación de confianza para el canal seguro SSL/TLS con la autoridad ". ---> System.Net.WebException: Se ha terminado la conexión: No se puede establecer una relación de confianza para el canal seguro SSL/TLS. --> System.Security.Authentication.AuthenticationException: El certificado remoto no es válido según el procedimiento de validación.”

Causa	Posible acción por tomar
El certificado que asegura el servicio no está disponible.	En el binding de IIS, para el puerto seguro (443), garantice que se esté usando el certificado de agente electrónico proporcionado por el BCCR. Consultar el punto 8 .
El usuario del pool no tiene permisos sobre la llave privada del certificado de agente electrónico.	Verificar que el usuario del pool que ejecuta la aplicación que consume los servicios del Firmador, tenga permisos sobre la llave privada del certificado de agente electrónico. Para esto abra una ventana MMC, busque los certificados de “equipo local” y en el store “Personal”, elija el certificado de agente electrónico, clic derecho -> Todas la tareas -> Administrar claves privadas. 

	En la lista mostrada debe estar el usuario del pool. 
El dominio del SAN, contenido en el certificado que asegura el servicio, no coincide con el dominio en URL del servicio.	Verifique que la dirección del URL coincide exactamente con el campo SAN del certificado expuesto en HTTP. Por ejemplo: si el URL es https://pruebas_bccr.central.bccr.fi.cr/ServicioDeNotificacion.svc El SAN debe ser pruebas_bccr.central Si el campo SAN tiene un comodín de dominio (* = asterisco), verifique que el nombre del servidor se encuentre al mismo nivel del *. Por ejemplo: si requiere un certificado https para la maquina: pruebas_bccr, cuyo nombre completo es pruebas_bccr.central.bccr.fi.cr, requiere un certificado de agente con SAN "*.central.bccr.fi.cr" y no basta con uno que diga solamente "*.bccr.fi.cr".
No se puede validar la cadena de confianza del certificado que asegure el servicio.	Garantizar que el cliente puede validar la cadena de confianza completa del certificado https que se encuentra "hosteando" el servicio. Para eso se puede usar la herramienta del sistema operativo: certutil -url certificado.cer Para cada uno de los miembros de la cadena de confianza. Adicionalmente, se debe garantizar que la cadena de confianza esté instalada correctamente en el cliente que invoca.

El cliente y el servidor no utilizan el mismo protocolo de comunicación, utilizando distintas versiones de TLS.	Garantizar que el cliente y el servidor utilicen el mismo protocolo de comunicación, por ejemplo: TLS 1.2.
--	--

19. Solución al error Hash invalido en la notificación de una firma

Si en la notificación de una firma se recibe el código de error 8 (Hash invalido), se debe verificar la manera en la que se calcula el hash del documento, para esto consulte el punto [4](#). Además, se debe verificar el tipo de codificación utilizado al enviar dicho hash en la solicitud de firma, para esto consulte el punto [5](#).

20. Solución al error 403.16 al invocar un web service o un WCF

Este error puede suceder porque en el store de raíces de confianza, existen certificados no auto emitidos.

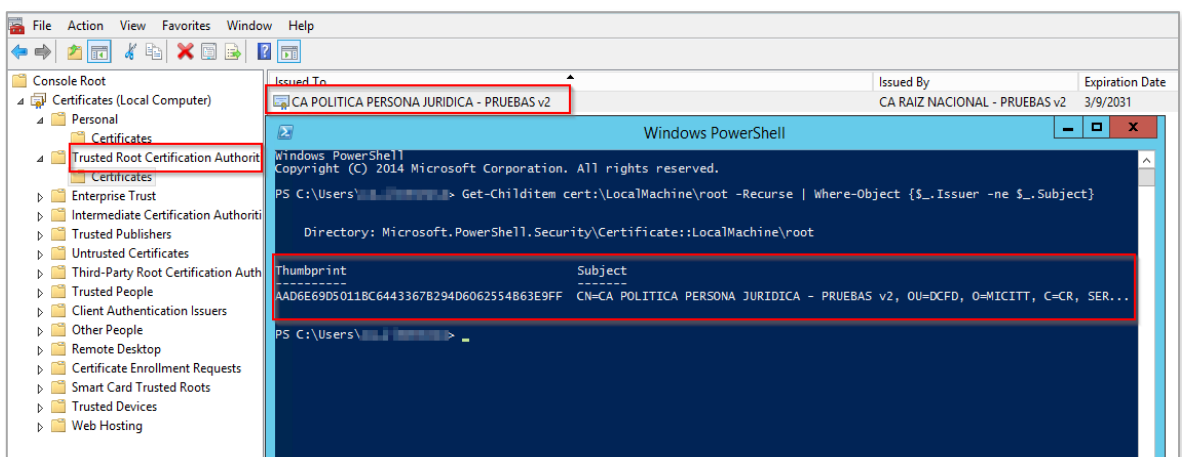
Para verificarlo diríjase al servidor que rechaza la conexión, abra una consola de **Power Shell** y ejecute el comando **Get-Childitem cert:\LocalMachine\root -Recurse | Where-Object {\$_.Issuer -ne \$_.Subject}**

Lo correcto es que el comando no retorne ningún resultado:

```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\...> Get-Childitem cert:\LocalMachine\root -Recurse | Where-Object {$_.Issuer -ne $_.Subject}
PS C:\Users\...>
```

Si el comando retorna algún certificado, este debe moverse del store de “Entidades de certificación raíz de confianza”, al store que le corresponda.



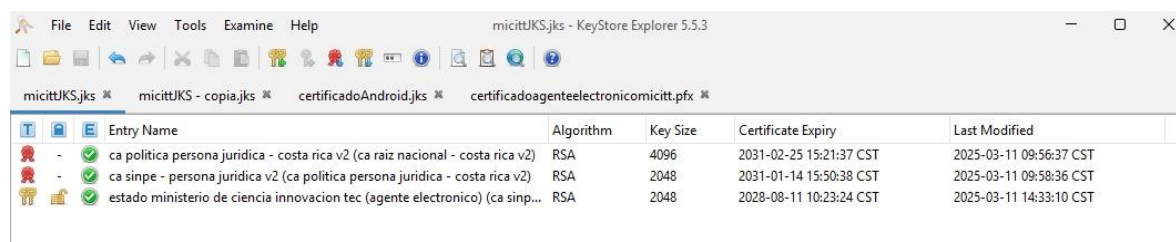
Por ejemplo, el certificado “CA Política Persona Jurídica – Pruebas v2”, debe estar en el store “Entidades de certificación Intermedias” y debería moverse a dicho store.

Para aplicaciones en Java.

Se debe verificar el KeyStore (para ello se puede descargar el [KeyStore Manager](#)) del servidor donde se originan las solicitudes de autenticación/firma, en el mismo se deben disponer de 3 certificados únicamente, que se indican a continuación

- Llave pública de ca sinpe - persona jurídica v2
- Llave pública de ca política persona jurídica - costa rica v2
- Llave privada del certificado de Agente Electrónico de la entidad

Un Ejemplo de este KeyStore sería:



The screenshot shows the KeyStore Explorer 5.5.3 interface. The 'micittJKS.jks' store is selected, displaying three entries in a table. Each entry has a status icon (red flag for errors, green checkmark for success), an entry name, an algorithm, key size, certificate expiry, and last modified date.

	Entry Name	Algorithm	Key Size	Certificate Expiry	Last Modified
🚩	ca politica persona juridica - costa rica v2 (ca raiz nacional - costa rica v2)	RSA	4096	2031-02-25 15:21:37 CST	2025-03-11 09:56:37 CST
✅	ca sinpe - persona juridica v2 (ca politica persona juridica - costa rica v2)	RSA	2048	2031-01-14 15:50:38 CST	2025-03-11 09:58:36 CST
✅	estado ministerio de ciencia innovacion tec (agente electronico) (ca sinp...	RSA	2048	2028-08-11 10:23:24 CST	2025-03-11 14:33:10 CST

Además como punto importante es verificar que **sólo existan estos 3 elementos**, ya que si existe otro certificado con llave privada los errores continuarán.

21. Solución al error 403.4 al invocar un web service o un WCF

Este error sucede cuando se invoca un web service o un WCF y el método invocado intenta acceder a un recurso que no tiene disponible o no tiene permisos sobre el mismo.

Por ejemplo, si el método intenta escribir un log en la ruta “C:\logs” y el usuario del pool no tiene permisos para escribir en esta ruta, entonces al cliente se le muestra el error 403 y en el lado del servicio, en el log de Failed request se encuentra el “statusCode = 403.4”.

Para solucionarlo se le debe dar permisos al usuario del pool para que pueda escribir en la ruta donde se guarda el log.

22. Solución al error "The message with Action

'http://tempuri.org/ValidadorDeDocumento/ValideElServicio' cannot be processed at the receiver, due to a ContractFilter mismatch at the EndpointDispatcher. This may be because of either a contract mismatch (mismatched Actions between sender and receiver) or a binding/security mismatch between the sender and the receiver"

- El error sucede porque el Binding del cliente y del servicio no tienen la misma configuración de seguridad.

El servicio está configurado con un Binding con "security mode="Transport"" y "transport clientCredentialType="Certificate"". El cliente debe tener esta misma configuración y utilizar el certificado de agente electrónico de la entidad para invocar el servicio.

Endpoint del servicio

```
<service behaviorConfiguration="BccrServiceFva_ServiceBehavior"
  name="Bccr.Firma.Fva.Entidades.ValidarDocumento.Wcf.SI.Servicios.ValidadorDeDocumentos">
  <endpoint address=""
    behaviorConfiguration="BccrServidorFva_EndpointBehavior"
    binding="wsHttpBinding"
    bindingConfiguration="WSHttpBinding_BccrFva"
    name="WSHttpBinding_ISelladorElectronicoConControlDeLlave"
    contract="Bccr.Firma.Fva.Entidades.ValidarDocumento.Wcf.SI.Servicios.ValidadorDeDocumentos" />
</service>
```

Binding del servicio

```
<binding name="WSHttpBinding_BccrFva" maxReceivedMessageSize="28311552">
  <security mode="Transport">
    <transport clientCredentialType="Certificate" proxyCredentialType="None" realm="" />
  </security>
</binding>
```

- También revisar que la referencia al servicio se creó con la descripción de este actualizada. Por ejemplo, si utiliza un archivo WSDL verificar que el mismo tiene la descripción del servicio actualizada y no apunta un servicio obsoleto.
- Si es el servicio de notificación de la entidad, verificar que el endpoint del servicio tenga el contrato apuntando directo a la clase ResultadoDeSolicitud y no a una interfaz (I ResultadoDeSolicitud)

```
<service name="wsPuntoGAUDI.ResultadoDeSolicitud"
  behaviorConfiguration="BehaviorResultadoGaudi">
  <endpoint address=""
    binding="wsHttpBinding"
    bindingConfiguration="EnlaceConSeguridadDeTipoTransporte"
    contract="wsPuntoGAUDI.IResultadoDeSolicitud"
    listenUri="" />
</service>
```

```
15
16
17
18
19
20
21
22
23
24
25
<service name="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WCF.ResultadoDeSolicitud"
  behaviorConfiguration="ComportamientoDelServicioDeLaEntidadPersonalizado">
  <endpoint address=""
    binding="wsHttpBinding"
    bindingConfiguration="EnlaceConSeguridadDeTipoTransporte"
    contract="EjemploDeServicioDeNotificacionDePersonaFisica.Servicios.WCF.ResultadoDeSolicitud"
    listenUri="" />
</endpoint>
</service>
```

23. Solución al error 403.13 al invocar un web service o un WCF

Este error sucede cuando se invoca un web service o un WCF con un certificado que no es válido para el servidor que recibe el llamado.

Por ejemplo, los servicios de GAUDI se deben invocar con el certificado de Agente Electrónico de la entidad, este certificado debe pertenecer a la jerarquía nacional. Si la entidad invoca los servicios de GAUDI con un certificado que no es el de Agente Electrónico, el BCCR lo validaba y lo rechaza dado que no es el certificado esperado.

Cuando la entidad realizaba la invocación con el certificado incorrecto en el log de Failed Request se encuentra el error 403.13 indicando que el servidor rechazaba el certificado del cliente.

24. Solución al error SecureChannelFailure al configurar la identidad de marca en Central Directo

Al configurar la identidad de marca y colocar el URL del servicio de notificación se muestra el error SecureChannelFailure

The screenshot shows a web form titled 'Configuración' with the following fields and content:

- Nombre:** [Redacted]
- Código:** 1
- Configuración:** Seleccione el tipo de servicio de la dirección y luego digite la dirección https del servicio donde se realizarán las notificaciones de firma.
- Tipo de servicio:** Web Service (dropdown menu)
- Canal:** Privado (dropdown menu)
- URL:** https://[Redacted]Notificador.wsdl
- Error Message:** Ocurrió un problema al tratar de consultar la dirección. **Detalle técnico Status: SecureChannelFailure**

La entidad estaba desarrollando con JAVA y no validaba correctamente el certificado de agente electrónico del Banco Central

Para solucionar el error se debe incluir la jerarquía del certificado de Agente electrónico del BCCR en los keystores de java correspondientes: "sslcerts" y "keystore".

25. Solución al error 403 al invocar un web service o un WCF

Al invocar un web service o un WCF se obtiene el error 403, y en el lado del servicio, en el log de Failed request se encuentra el "statusCode = 403".

```
<failedRequest url="https://[REDACTED]/ServicioNotificacionGAUDI/ResultadoDeSolicitud.svc"
  siteId="1"
  appPoolId="PoolGAUDI"
  processId="3676"
  verb="POST"
  remoteUserName=""
  userName=""
  tokenUserName="NT AUTHORITY\IUSR"
  authenticationType="anonymous"
  activityId="{00000000-0000-0000-BD0D-0080070000FF}"
  failureReason="STATUS_CODE"
  statusCode="403"
  triggerStatusCode="403"
  timeTaken="31"
  xmlns:freb="http://schemas.microsoft.com/win/2006/06/iis/freb"
>
```

Al activar el trace de WCF del lado del servidor se muestra el error "Client certificate is required. No certificate was found in the request"

The screenshot displays the WCF Trace tool in Visual Studio. The top section shows a list of activities with columns for Description, Level, Thread ID, Process Name, Time, Trace Identifier, Activity Name, and Source. An error activity is highlighted with a red icon and the message "Client certificate is required. No certificate was found in the request." Below the table, the "Basic Information" pane shows details for the selected error, including the Name, Activity Name, Time, Level, Source, Process, and Thread. The "General Properties" pane provides further details, including the TraceRecord Severity (Error), TraceIdentifier, Description (Client certificate is required. No certificate was found in the request...), AppDomain, Source, Content-Length, and Content-Type.

Description	Level	Thread ID	Process Name	Time	Trace Identifier	Activity Name	Source
From: Listen at https://piprjgvm.poderjudicial.go.cr:84...	Transfer	68	w3wp	18/1/2024 15:20:52.0702033	Trace Transfer		System.Serv...
Activity boundary.	Start	68	w3wp	18/1/2024 15:20:52.0702033	https://docs.microsoft.com...	Receive byt...	System.Serv...
Connection information.	Information	68	w3wp	18/1/2024 15:20:52.0702033	https://docs.microsoft.com...	Receive byt...	System.Serv...
Client certificate is required. No certificate w...	Error	68	w3wp	18/1/2024 15:20:52.0702033	https://docs.microsoft.com...	Receive ...	System S...
→ Sent a message over a channel.	Information	68	w3wp	18/1/2024 15:20:52.0702033	https://docs.microsoft.com...	Receive byt...	System.Serv...
Authentication failed for HTTP(S) connection	Information	68	w3wp	18/1/2024 15:20:52.0702033	https://docs.microsoft.com...	Receive byt...	System.Serv...
To: Listen at https://piprjgvm.poderjudicial.go.cr:8443...	Transfer	68	w3wp	18/1/2024 15:20:52.0702033	Trace Transfer	Receive byt...	System.Serv...
Activity boundary.	Stop	68	w3wp	18/1/2024 15:20:52.0858021	https://docs.microsoft.com...	Receive byt...	System.Serv...

Basic Information

Name	Value
Activity Name	Receive bytes on connection 'https://[REDACTED]/ServicioNotificacionGAUDI/ResultadoDeSolicitud.svc'.
Time	2024-01-18 15:20:52.0702
Level	Error
Source	System.ServiceModel
Process	w3wp
Thread	68

General Properties

Name	Value
[TraceRecord] Severity	Error
TraceIdentifier	https://docs.microsoft.com/dotnet/framework/wcf/diagnostics/tracing/System.ServiceModel.Channels+httpsClientCertificateNotPresent
Description	Client certificate is required. No certificate was found in the request. This might be because the client certificate could not be successfully validated by the operating system or IIS. For information on how to bypass...
AppDomain	/LM/W3SVC/1/ROOT/ServicioNotificacionGAUDI-2-133500864514772904
Source	System.ServiceModel.Channels.HttpsChannelListener`1[System.ServiceModel.Channels.IReplyChannel]/38568216
Content-Length	1427
Content-Type	application/json; charset=utf-8

Este error sucede cuando antes del servidor existe un firewall que examina los paquetes. Entonces el firewall desarma el paquete, lo examina y luego lo vuelve a armar, pero lo arma sin el certificado del cliente que invoca el servicio.

Para corregirlo se debe cambiar la configuración del firewall para que no examine los paquetes y solamente los reciba y los envíe al servidor.

26. Solución al error “StatusCode Forbidden” al configurar la identidad de marca en Central Directo

Al configurar la identidad de marca y colocar el URL del servicio de notificación se muestra el error StatusCode Forbidden



Este error se da porque la entidad está rechazando el certificado del BCCR

Se debe aplicar lo siguiente:

- Revisar acceso a los CRLs, para esto utilizar la "Guía para validar los CRLs de certificados de la jerarquía del ambiente de producción del firmador" (<https://www.bccr.fi.cr/firma-digital/DocFirmaDigital/Guia-validar-CRLs-certificados-jerarquia-ambiente-produccion-firmador.pdf>)
- Revisar la jerarquía nacional del ambiente de producción de GAUDI, para esto utilizar la "Guía para instalar los certificados de la jerarquía del ambiente de producción del firmador ." (<https://www.bccr.fi.cr/firma-digital/DocFirmaDigital/Guia-instalar-certificados-jerarquia-ambiente-produccion-firmador.pdf>) y ejecutar los pasos del punto 20 de este documento.
- Revisar las validaciones realizadas al certificado del BCCR desde el código fuente, para esto consultar los puntos 12 y 13 de este documento.

27. Pasos para generar la clase “ResultadoDeSolicitud” a partir del archivo “ResultadoDeSolicitud.json”.

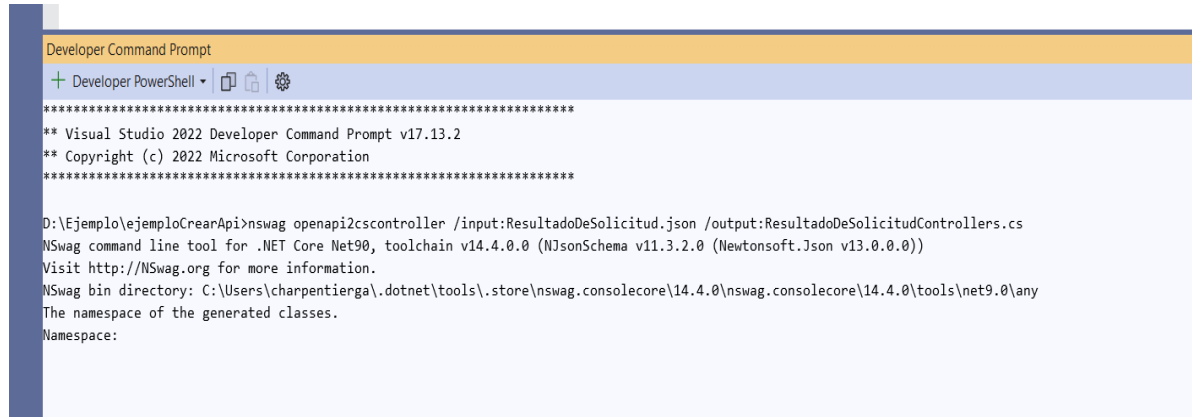
- a. Instalar NSwagStudio (GUI) desde la consola de comandos con el siguiente comando: `dotnet tool install --global NSwag.ConsoleCore`

```
Developer Command Prompt
+ Developer PowerShell | [icon] [icon]
*****
** Visual Studio 2022 Developer Command Prompt v17.13.2
** Copyright (c) 2022 Microsoft Corporation
*****

D:\Ejemplo\ejemploCrearApi>
D:\Ejemplo\ejemploCrearApi>dotnet tool install --global NSwag.ConsoleCore
You can invoke the tool using the following command: nswag
Tool 'nswag.consolecore' (version '14.4.0') was successfully installed.

D:\Ejemplo\ejemploCrearApi>
```

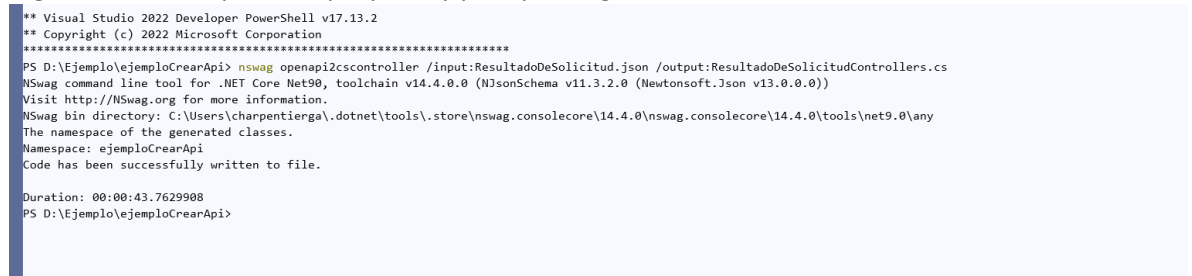
- b. Generar el código del controller con el siguiente comando: nswag openapi2cscontroller /input:ResultadoDeSolicitud.json /output:ResultadoDeSolicitudControllers.cs



```
Developer Command Prompt
+ Developer PowerShell
*****
** Visual Studio 2022 Developer Command Prompt v17.13.2
** Copyright (c) 2022 Microsoft Corporation
*****

D:\Ejemplo\ejemploCrearApi>nswag openapi2cscontroller /input:ResultadoDeSolicitud.json /output:ResultadoDeSolicitudControllers.cs
NSwag command line tool for .NET Core Net90, toolchain v14.4.0.0 (NJsonSchema v11.3.2.0 (Newtonsoft.Json v13.0.0.0))
Visit http://NSwag.org for more information.
NSwag bin directory: C:\Users\charpentierga\.dotnet\tools\.store\nswag.consolecore\14.4.0\nswag.consolecore\14.4.0\tools\net9.0\any
The namespace of the generated classes.
Namespace:
```

- c. Ingresar el namespace del proyecto y para poder generar la clase



```
** Visual Studio 2022 Developer PowerShell v17.13.2
** Copyright (c) 2022 Microsoft Corporation
*****
PS D:\Ejemplo\ejemploCrearApi> nswag openapi2cscontroller /input:ResultadoDeSolicitud.json /output:ResultadoDeSolicitudControllers.cs
NSwag command line tool for .NET Core Net90, toolchain v14.4.0.0 (NJsonSchema v11.3.2.0 (Newtonsoft.Json v13.0.0.0))
Visit http://NSwag.org for more information.
NSwag bin directory: C:\Users\charpentierga\.dotnet\tools\.store\nswag.consolecore\14.4.0\nswag.consolecore\14.4.0\tools\net9.0\any
The namespace of the generated classes.
Namespace: ejemploCrearApi
Code has been successfully written to file.

Duration: 00:00:43.7629908
PS D:\Ejemplo\ejemploCrearApi>
```

- d. Utilizar la clase en el proyecto, se debe limpiar el código autogenerated que no se necesite y debe quedar de la siguiente manera.

Nota: También puede utilizar este código en su implementación y omitir los pasos anteriores.

```
namespace ejemploCrearApi
{
    using System = global::System;

    public interface IImplementacion
    {

        /// <returns>OK</returns>

        System.Threading.Tasks.Task NotifiqueLaRespuestaAsync(ResultadoDeFirma
body);

        /// <returns>OK</returns>

        System.Threading.Tasks.Task<bool> ServicioDisponibleAsync();

    }
}
```

```

public partial class Controller : Microsoft.AspNetCore.Mvc.ControllerBase
{
    private IImplementacion _implementacion;

    public Controller(IImplementacion implementation)
    {
        _implementacion = implementation;
    }

    public System.Threading.Tasks.Task
    NotifiqueLaRespuesta([Microsoft.AspNetCore.Mvc.FromBody] ResultadoDeFirma
    body)
    {
        return _implementacion.NotifiqueLaRespuestaAsync(body);
    }

    public System.Threading.Tasks.Task<bool> ServicioDisponible()
    {
        return _implementacion.ServicioDisponibleAsync();
    }
}

public partial class ResultadoDeFirma
{
    public int IdDeLaSolicitud { get; set; }

    public byte[] DocumentoFirmado { get; set; }

    public bool FueExitosa { get; set; }

    public intCodigoDeError { get; set; }

    public int IdAlgoritmoHashDocumentoFirmado { get; set; }

    public string HashDocumentoFirmado { get; set; }

    public byte[] HashDelDocumentoFirmadoEnBytes { get; set; }
}
}

```

28. Configurar desde LINUX el balanceador de Oracle para configuración SSL/TLS del servicio de notificación

Se muestra una interfaz similar a la siguiente:

- En el campo **Certificate** debe ir el **certificado de Agente Electrónico** de la entidad:
 - Debe estar en formato **PEM**.
 - Solo debe contener el certificado de Agente Electrónico de la entidad.

Aunque el certificado hoja puede tener extensión .cer, .crt o .pem, lo importante es que esté en formato PEM, texto plano con encabezados:

-----BEGIN CERTIFICATE-----

.
.
.

-----END CERTIFICATE-----

- En el campo **Certificate Chain** debe ir la cadena de certificados:
 - Archivo PEM que contenga:
 - Primero el certificado intermedio **CA SINPE - PERSONA JURIDICA v2**
 - Segundo el certificado intermedio **CA POLITICA PERSONA JURIDICA - COSTA RICA v2**
 - Tercero el certificado raíz **CA RAIZ NACIONAL - COSTA RICA v2**

Deben ir concatenados en ese orden similar al siguiente ejemplo:

PEM encoded chain:

```
-----BEGIN CERTIFICATE-----
MIIFZTCCBLWgAwIBAgITSAACH3ZMacgxwd6uAACAAAIftANBgkqhkiG9w0BAQ0SF
ADCBmzEZMBcGA1UEBRMQ01BKLTQ0tMDAwLTAwNDANzELMAkGA1UEBhMCQ1IxDJAi
BgNVBAoTG0JBTKNPIENFTlRSOUwgREUgO09TVEEgUklDQTEiMCAGA1UECzMZRElW
SVNJOT04gU0lTVEVNOVMgREUgUEFHNTZEnMCUGA1UEAxMeQ00EgU0l0UEUgLSB0RVJT
T05BIEpVUklESUNBIHYyMB4XDTI1MDgwMTIwMTYxMFOxDTI1MDczMTIwMTYxMFOw
gY4xGTAXBgNVBAUENQSi0zLTUwMS0wNjkyMjcxCzAJBgNVBAYTAkNSMRkwFwYD
V0OKExBORVJT05BIEpVUklESUNBMUkwRwYDV0QDE0BTT0l0IFNPTFVDSU90RVMg
SU5URUd5SQUxFTUyBTT0NJRURBRCB8BTk90SU1BICBR0VOVEUgRUXF01RST05J008p
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ08AMIIBCgKCAQEAqaIAP+aJdhoImmILdHw7
8r54M4Ui8m4EeX6njYsWaqnWxs5GRns7ecj0xexPVq7DvL/ebMxyV5zfEy5duWxm
Ozw0iAD0bw/knrTuS/mVknx47MP50ws886GcLzc21j8Ds0VB49LD4d86U1tgz8Dt
VrX0w28Lr7Ys3zhrMOTTP+AukR57N/Ph6AZhvOR0o1+1wUihfgmPRgo1R8e8b08+
l062utepRhu8zhqGKc3WLTnEKBCjI4qd8Rm0cMiPINGvM3+FrDhY0o1RDEl0jFge
KQtFbjf7Gxh/40TQ87+m3sHCULhaLd+Xc+YnS2YPP6/Q89+pdmqY0bs+/c47wwzd
2QIDAQAB04ICEzCCAg8wDgYDVR0PAAOH/BAQDAgTwMB0GA1UdDgQWBBS6Z7/SMmQl
i6W0y6KXDuLDRWPwuTAdBgNVHREEFjAUGHlqLmZpcmlhZG9yc29pb15jb20wHwYD
VR0fjBBgwFoAUNqzkWHLA834d7JPsvyPhyq55SrgwYwYDVR0fBFwwWjBYoFagVIZS
aHR0cDovL2ZkaS5zaW5wZS5maS5jci9yZXBvc2l0b3Jpb3p090SUYMFNJTlBFIjIw
LSUyMFBFUlNPTkE1MjBKVVJjRElDQSUyMHYyKDIPmNyYbDCBmgYIKwYBBQUHAQEE
gY0wYgYowKAYIKwYBBQUHMAAGGH0dHA6Ly9vY3NwLnNpbmB1LmZpLmNyL29jc3Aw
XGyYIKwYBBQUHMAKGUmh0dHA6Ly9mZGkuc2lucGUuZmkuaY3IvcmlwY3NpdG9yaW8v
Q0E1MjBTSU50RSUyMCO1MjB0RVJT05BjTIwS1VSSURJ00E1MjB2MjgyKS5jcnc0w
POYJKwYBBAGCNxUHBDALgYmKwYBBAGCNxUIhcTqW4LR4zWVWRuRUC+ZcYhQXLA4F/
hJmAcocA80gCAWOCAC0wH0YDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMCCG
CS5GAQ00BgjCVCgQaMBGwCgYIKwYBBQUHAwIwCgYIKwYBBQUHAwEwFOYDVR0gBA4w
DDAKBghggTwBAQEBBzANBgkqhkiG9w0BAQ0FAAOCAQEAq8sj9IGSisKuVRukS1hq
UYo/jGTLzTeaulNgM18DYJsnTuAPaqpHT79T+Sm2hPIajq0TECFP/1nDi/9CZpN
vNvbErH06Dk8BndF7+gzX9qecGq8KZFyZsNLw/XnPMFebaDGOzFX97/13yrfMi
VbaPw7JPJx0Zm384N7I7f2kLK0oi7zwA5I2F38gH+wmzJWRYxRPcGeCwVWYcxUud3
tgLCtT3vkFUEAt4bx9tDitIddWkoWeFD28fZPD/9fLBgJN6b7Bdp7T0sNOA10tTi
aZR+rphP6Ugkfw76RFgEDd3XlawcHsL63X8RD5USa0sM0sVga56i5g0D/D2kkCT
Hg==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIMJDCCGygAwIBAgITCAAAAadUITYy+pzigAAAAABjANBgkqhkiG9w0BAQ0F
ADB/MRkwFwYDVQQFEExBDUEotMi0xMDAtMDk4MzExMQswCQYDVQQGEwJDUjEPMA0G
A1UEChMGTU1DSVRUMQ0wCwYDVQQLEwREUgQ0ZEMTUwMwYDVQQDEyx0QSBQT0xJVElD
QSBORVJT05BIEpVUklESUNBI0gQ09TVEEgUklDQSB2MjAeFw0yMzAxMTQyMTQw
MzhaFw0zMTAxMTQyMTUwMzhaMIGbMRkwFwYDVQQFEExBDUEotNC0wMDAtMDA0MDE3
MQswCQYDVQQGEwJDUjEKMCIgA1UEChMbQkFOQ08gQ0VOVFJBTCEBRSB0T1NUQSBS
SUNBMSIwIAYDVQQLEw1ESVZJU0lPTiBTSVNURU1BUyBERSBQQUdPMScwJQYDVQQD
Ex5DQSBTSU5QRSATIFBFU1NPTkEgS1VSSURJQ0EgdjIwggEiMA0GCSCGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQDSwCna+nW0zq10QPh2kGPW06vvYwUBVm7Tgy0Ldjb
8+8V1gYfvj8nj12ijV3UnpSrEK3uZ6hmbMv+FoHHPcfJHtWUNbhYrUNVPe+rPKh
3OG/2LH8i+56ABCo4M6qaQJRRjHVfJ3laUOW+EBogdhr1aE5A39rCmC5ou8u+r1A
ZdMjUDEjkyBdUx1o37zRMtFukHHAhPT4RQss3VxT8PZ2UPdqdijGw1hAGmxvYuG
DKNWJEJFFfj/ETsLx3iWdBVYSvZcdy5Hr16qMawLmqK0i211fg0Q/Cf+YOM0Zta
```

- En el campo **Private Key** debe ir la **llave privada del certificado de Agente Electrónico** de la entidad:
 - Archivo .key generado junto con el .req.