

## Nomina Electronica

Configuracion del Servidor WildFly: Añadir en standalone.xml datasources NOMINADS, INTEGRACION y CONFIGFENEDS

**NOMINADS:** <datasource jta="true" jndi-name="java:/NOMINADS" pool-name="NOMINADS" enabled="true" use-ccm="true">

```
<connection-url>jdbc:oracle:thin:@ip:puertp:servicename</connection-url>
<driver>OracleJDBCdriver</driver>
  <security>
    <user-name>PEDIRLO A QUIEN CORRESPONDA</user-name>
    <password>PEDIRLO A QUIEN CORRESPONDA</password>
  </security>
  <validation>
    <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChe
cker"/>
    <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChe
cker"/>
    <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>
  </validation>
```

</datasource>

**INTEGRACION:** <datasource jta="true" jndi-name="java:/INTEGRACION" pool-name="INTEGRACION" enabled="true" use-ccm="true">

```
<connection-url>jdbc:oracle:thin:@ip:puertp:servicename</connection-url>
<driver>OracleJDBCdriver</driver>
  <security>
    <user-name>PEDIRLO A QUIEN CORRESPONDA</user-name>
    <password>PEDIRLO A QUIEN CORRESPONDA</password>
  </security>
  <validation>
    <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChe
cker"/>
    <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChe
cker"/>
    <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>
  </validation>
```

</datasource>

**CONFIGFENEDS:** <datasource jta="true" jndi-name="java:/INTEGRACION" pool-name="INTEGRACION" enabled="true" use-ccm="true">

```
<connection-url>jdbc:oracle:thin:@ip:puertp:servicename</connection-url>
<driver>OracleJDBCdriver</driver>
  <security>
    <user-name>PEDIRLO A QUIEN CORRESPONDA</user-name>
    <password>PEDIRLO A QUIEN CORRESPONDA</password>
  </security>
  <validation>
    <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChe
cker"/>
    <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChe
cker"/>
    <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>
  </validation>
</datasource>
```

**Crear el driver:** <driver name="OracleJDBCdriver" module="com.oracle">

```
<driver-class>oracle.jdbc.OracleDriver</driver-class>
```

</driver>

**CONFIGURACIONES ESPECIALES 1/1** En la ruta definida por la propiedad de sistema creada en el standalone.xml (CONFIG\_PATH), crear una carpeta en la ruta adafe\_home/com/ada/adafe/certificado, y asegurarse que el usuario que ejecuta el servidor tiene acceso de escritura. En esa ruta se almacenará la información de los clientes de nómina. Una vez terminadas las configuraciones desplegar el artefacto NominaElectronica.war en el servidor wildfly.

**BASE DE DATOS DE CONFIGURACIÓN** Nómina Electrónica usa tablas de configuración alimentadas por otras aplicaciones, para acceder a ellas, tal como se explica en la sección anterior, se utilizan los datasources configurados en el servidor a través de la aplicación apiRestConfig. Es importante recordar que las credenciales proporcionadas para los datasources tengan acceso a todos los schemas requeridos. En esta sección se detallan los campos claves para el correcto funcionamiento de los servicios web que Nómina Electrónica publica para su consumo. TABLA INTEGRACION.TBL\_FE\_CONEXION\_CLIENTE En esta tabla se almacena la información relevante a los datos necesarios para que el cliente interactúe con la DIAN y el software pueda acceder a los datos de los empleados del cliente. Cada cliente tiene su cadena de conexión a base de datos guardada en esta tabla, y el software dinámicamente utiliza la cadena de conexión para acceder a los datos del cliente. Los campos NE\_URL, NEV\_URL y NUMERACION\_URL hacen referencia a dianWsRest y deben tener la siguiente estructura:

CAMPO	VALOR
NE_URL	URL_WSREST/dianWsRest/v1/nomina
NEV_URL	URL_WSREST/dianWsRest/v1/habilitacion
NUMERACION_URL	URL_WSREST/dianWsRest/v1/numeracion

Donde URL\_WSREST es la url donde se ha desplegado dianWsRest.war Por ejemplo, si dianWsRest.war se ha desplegado en el mismo servidor donde se va a desplegar Nómina y el servidor está accesible en la IP 127.0.0.1 puerto 8081, los valores serían:

NE_URL	<a href="http://127.0.0.1:8081/dianWsRest/v1/nomina">http://127.0.0.1:8081/dianWsRest/v1/nomina</a>
NEV_URL	<a href="http://127.0.0.1:8081/dianWsRest/v1/habilitacion">http://127.0.0.1:8081/dianWsRest/v1/habilitacion</a>
NUMERACION_URL	<a href="http://127.0.0.1:8081/dianWsRest/v1/numeracion">http://127.0.0.1:8081/dianWsRest/v1/numeracion</a>

## SERVICIOS PUBLICADOS

### nomina/NominaGeneral

"/nomina/NominaGeneral/{codigoCliente}/{codigoMEmpresa}/{periodo}/{tipoArchivo}" Servicio para enviar a la DIAN las nóminas de todos los empleados del cliente de un determinado periodo.

Ejemplo de uso:

[http://10.1.140.102:8081/NominaElectronica/nomina/NominaGeneral/ADA\\_TESTNE/999999999/202004/1/](http://10.1.140.102:8081/NominaElectronica/nomina/NominaGeneral/ADA_TESTNE/999999999/202004/1/) codigoCliente =ADA\_TESTNE codigoMEmpresa = 999999999 periodo = 202004 tipoArchivo = 1

### NominaIndividual

"/nomina/NominaIndividual/{codigoCliente}/{codigoMEmpresa}/{periodo}/{tipoArchivo}/{codigoEmpleado}" Servicio para enviar la nómina de un empleado en específico.

Ejemplo de uso:

[http://10.1.140.102:8081/NominaElectronica/nomina/NominaIndividual/ADA\\_TESTNE/999999999/202004/1/10](http://10.1.140.102:8081/NominaElectronica/nomina/NominaIndividual/ADA_TESTNE/999999999/202004/1/10) codigoCliente =ADA\_TESTNE codigoMEmpresa = 999999999 periodo = 202004 tipoArchivo = 1 codigoEmpleado = 10

### NominaDeAjuste

"/nomina/NominaDeAjuste/{codigoCliente}/{codigoMEmpresa}/{periodo}/{tipoArchivo} Servicio para modificar o eliminar las nóminas de los empleados preparados en la base de datos

Ejemplo de uso:

[http://10.1.140.102:8081/NominaElectronica/nomina/NominaDeAjuste/ADA\\_TESTNE/999999999/202001/2](http://10.1.140.102:8081/NominaElectronica/nomina/NominaDeAjuste/ADA_TESTNE/999999999/202001/2) codigoCliente =ADA\_TESTNE codigoMEmpresa = 999999999 periodo = 202001 tipoArchivo = 2

### ValidarCliente

"/nomina/ValidarCliente/{codigoCliente}/{codigoMEmpresa}/{periodo}/{tipoArchivo}/{validaciones}" Servicio para realizar la habilitación en la DIAN de un cliente, donde el parámetro validaciones indica el número de nóminas a enviar para obtener la habilitación.

Ejemplo de uso:

[http://10.1.140.102:8081/NominaElectronica/nomina/ValidarCliente/METROPARQUES\\_NE/999999999/202109/1/](http://10.1.140.102:8081/NominaElectronica/nomina/ValidarCliente/METROPARQUES_NE/999999999/202109/1/) 5 codigoCliente = METROPARQUES\_NE codigoMEmpresa = 999999999 periodo = 202109 tipoArchivo = 1 validaciones = 5

**testCliente** "/nomina/testCliente/{codigoCliente}" Servicio que realiza un chequeo de que toda la información del cliente está bien configurada y es accesible. Se realizan las siguientes comprobaciones: - El código del cliente existe en la tabla de configuración: tblFeConexionCliente ADA\_TESTNE: OK El cliente existe en la tabla de certificados: feCertificatelInfo 800167494: OK - El

archivo con las rutas a la información del certificado existe: propiedadesCliente: OK - El certificado del cliente existe: CFX\_CRIPTO : EXISTE OK - Se hace una firma de un xml vacio para comprobar que las claves funcionan: FIRMA EXITOSA OK - Se intenta conectar a la base de datos del cliente:

```
CONEXION A LA BASE DE DATOS DEL CLIENTE EXITOSA OK Se comprueba que el servicio de envío a la DIAN es accesible:  
CONEXION dianWsRest EXITOSA OK:  
{\"estadoTx\":true,\"msg\":null,\"codigoHttp\":200,\"fechaOperacion\":1646340020, \"objectResponse\": \"V 1.0.0\", \"lstObjectResponse\":null}"
```

Ejemplo de uso: [http://10.1.140.102:8081/NominaElectronica/nomina/testCliente/ADA\\_TESTNE](http://10.1.140.102:8081/NominaElectronica/nomina/testCliente/ADA_TESTNE)  
codigoCliente = ADA\_TESTNE

## RETORNO DE LOS SERVICIOS

Todos los servicios retornan la misma estructura json:

CAMPO	TIPO	DESCRIPCION
estadoTx	BOOLEAN	Cierto si el proceso ha terminado correctamente.
mag	STRING	Mensaje de error.
codigoHttp	NUMERIC	Código HTTP devuelto por la DIAN o interno del software en caso de no poder enviar la nómina a la DIAN.
fechaOperacion	STRING	Fecha y hora del evento.
numeroEnvio	NUMERIC	Número único del envío.
lstObjectResponse	STRING	

**Ejemplo exitoso:** { "estadoTx": true, "msg": "", "codigoHttp": 0, "numeroEnvio": 221231102745,

```
"fechaOperacion": "2021-12-31T15:27:45.269+0000",  
"lstObjectResponse": ""
```

}

### Ejemplo fallido, con una nómina errónea:

```
{ "estadoTx": false, "msg": "Error XML DIAN", "codigoHttp": 307, "numeroEnvio": 221231104011,  
"fechaOperacion": "2021-12-31T15:40:11.895+0000", "lstObjectResponse": "15" }
```

### Ejemplo fallido, varias nóminas erróneas:

```
{ "estadoTx": false, "msg": "Error XML DIAN", "codigoHttp": 307, "numeroEnvio": 221231104149,  
"fechaOperacion": "2021-12-31T15:41:49.265+0000",
```

```
"lstObjectResponse": "10-13-15-8778-8979-9332-9398-11715-12744"
```

}

## LOG Y BASE DE DATOS

Cada vez que se realiza un envío de una nómina se almacena la información en la tabla NOMINA.NE\_ENVIADOS\_WS cuyos campos son:

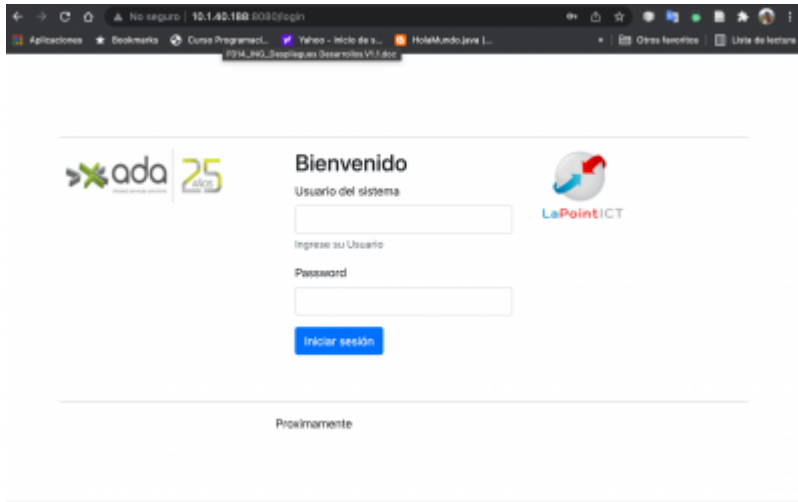
CAMPO	TIPO	DESCRIPCION
ID	NUMERIC	Identificador de la tabla.
PERIODO	VARCHAR	Periodo de generación del xml.
NIT_CLIENTE	NUMERIC	NIT del cliente.
CODIGO_EMPLEADO	NUMERIC	Código del empleado. Puede ser nulo si el error se produce antes de la generación del xml.
FECHA_ENVIO	TIMESTAMP	Fecha y hora de generación del registro.
NUMERO_PREDECESOR	VARCHAR	Consecutivo del xml, se almacena para su uso en caso de tener que generar una nómina de ajuste.
CUNE_PREDECESOR	VARCHAR	CUNE del xml, se almacena para su uso en caso de tener que generar una nómina de ajuste.
FECHA_GENERACION_PREDECESOR	TIMESTAMP	Fecha de generación del xml, se almacena para su uso en caso de tener que generar una nómina de ajuste.
TIPO_ARCHIVO	VARCHAR	Tipo de archivo del xml.
TIPO_NOTA	VARCHAR	Tipo de nota del xml.
ERROR	NUMERIC	Identificador del error. 0 = Sin error.1 = Error durante el proceso de generación del xml 2 = Error de la DIAN
TIPO_XML	NUMERIC	Tipo de xml.
CODIGO_EMPRESA	VARCHAR	Código multiempresa del cliente.
FECHA_REGISTRO	TIMESTAMP	Fecha y hora de generación del registro.
USUARIO_REGISTRO	VARCHAR	Usuario que genera el registro.
CLOB_XML	VARCHAR	Xml enviado a la DIAN
NUMERO_ENVIO	NUMERIC	Número de envío único por transmisión

El Sistema cuenta con una herramienta de configuración que le permite subir los certificados de seguridad y esta compuesta por dos artefactos: - confiFactura (Sistema WEB). - apiRestConfig (Paquete de Servicios)

**confiFactura** Tiene los siguientes componentes: - Login

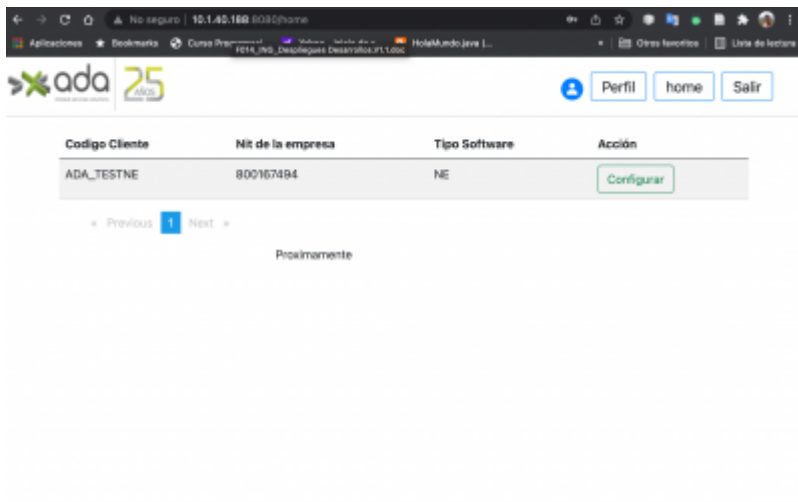
El login esta asociado a las siguientes tablas del sistema era:

- TBL\_FE\_CONEXION\_CLIENTE\_USER
- TBL\_USUARIOS



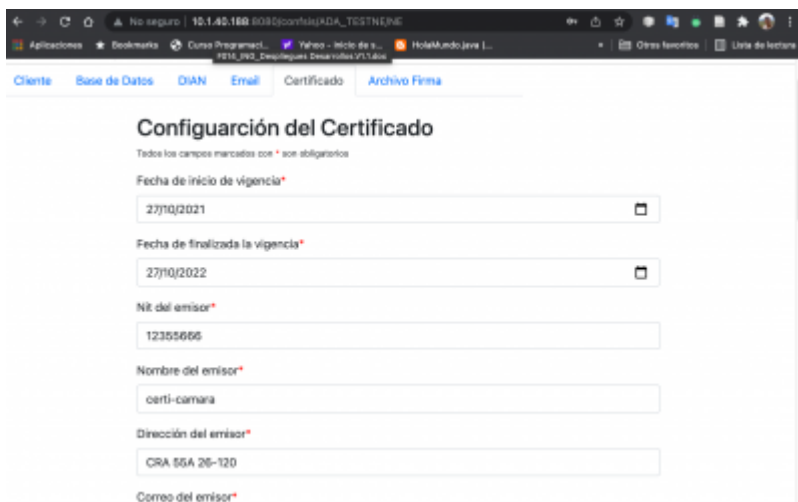
### - Lista de empresas

Una vez ingresa el sistema le muestra la lista de empresas a las que el usuario logueado tiene permisos de configurar



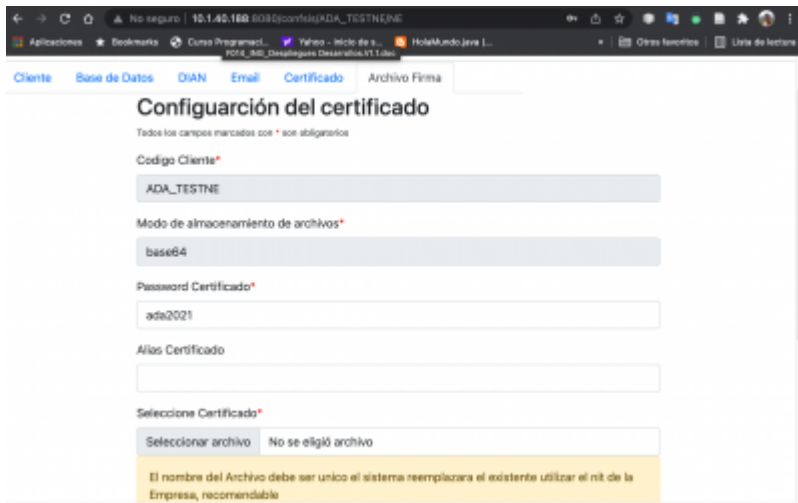
### - Configuración del Certificado

Aqui el usuario debe ingresar información acerca del proveedor del certificado y fechas de vigencia como lo solicita el formulario asociado:



## - Configuración del Archivo

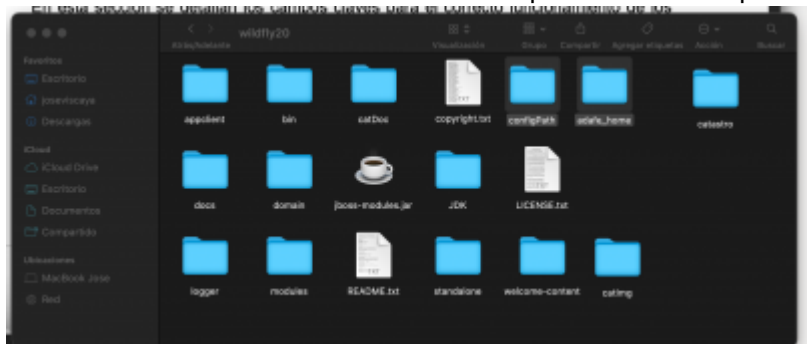
Aquí el usuario selecciona el archivo que contiene el certificado de seguridad para subirlo al sistema



Una vez realizado ese proceso el sistema esta preparado y se puede validar mediante el uso del servicio validado ya descrito en este documento.

## apiRestConfig

Este artefacto de software es quien provee de servicios al sistema anterior el cual debe estar desplegado en el mismo servidor y se debe garantizar la configuración del datasource CONFIGFENEDS el cual esta descrito al inicio de este documento. Además de esto es necesario que exista la carpeta



configPath con la siguiente información:

