**DOCUMENTO CONTRATO DEL SERVICIO**

**FECHA DE PUBLICACION:** SEPTIEMBRE 2021

**NOMBRE:** NOTE\_WS

**VERSION:** 3.03

**CONTROL DE CAMBIOS**

|  |  |  |
| --- | --- | --- |
| **Fecha modif.** | **Versión** | **Descripción** |
| 01/07/2010 | 1.0 | Versión inicial del documento |
| 26/06/2013 | 1.1 | Revisión del documento |
| 08/01/2014 | 1.2 | Se ha añadido el método obtenerNotificacionesPendientesConAsunto |
| 11/02/2014 | 1.3 | Observaciones – Recomendaciones |
| 30/09/2015 | 1.4 | Se añaden los métodos: obtenerAcuseRecibo y generarPDF |
| 17/12/2015 | 1.5 | Se añade método existeNIF |
| 23/02/2016 | 1.6 | Se añaden los métodos : obtenerNotificacionesPendientesTotal y obtenerNotificacion |
| 22/09/2016 | 1.7 | Se añade método preAltaAbonado |
| 15/12/2016 | 1.8 | Se modifica el método enviarNotificacion para añadir información en el retorno |
| 24/11/2017 | 1.9 | Se modifica tras la integración con la plataforma Notific@  Se añade método obtenerNotificacionesPendientesTotalGestor  Se modifica método preAltaAbonado (medioPreAlta a obligatorio) |
| 29/05/2018 | 2.0 | Se modifica el método existeNIF para añadir información en el retorno y nuevos tipos de retorno  Se modifican el método enviarNotificacion para incorporar tipos de respuesta |
| 06/06/2018 | 2.1 | Se añaden los métodos: enviarNotificacionNotifica y obtenerNotificacionesPendientesTotalAcuse |
| 08/10/2018 | 2.2 | Integración con Ley 39.  Se modifica el método de preAltaAbonado para indicar que el nombre es obligatorio. |
| 14/01/2019 | 2.3 | Se incluye el método obtenerNotificacionAcuse |
| 14/02/2019 | 2.4 | Se añade el parámetro de retorno codarchivo en los métodos obtenerNotificacionesPendientesTotalAcuse y obtenerNotificacionAcuse |
| 29/04/2019 | 2.5 | Se incluye el método consultaNotificacion |
| 13/12/2019 | 2.6 | Se incluye el método enviarNotificacionGfic |
| 31/01/2020 | 2.7 | Se incluye el método de la versión 2.0 del Adviser |
| 06/03/2020 | 2.8 | Se añade la posibilidad de enviar el Procedimiento Administrativo al enviar la notificación: enviarNotifiacion y enviarNotifiacionGfic |
| 22/05/2020 | 2.9 | Se incluyen los métodos para la integración con LEMA: localizaNotificaciones, peticionNotificacion, enviarNotificacionUnidadRegistro y recepcionNotificacionUnidadRegistro |
| 12/02/2021 | 3.0.1 | Se incluye la integración con NOTE\_POSTAL |
| 19/04/2021 | 3.0.2 | Se realiza las modificaciones para la estabilización de NOTE\_POSTAL |
| 28/09/2021 | 3.0.3 | Se realiza las modificaciones para la integración con IPAE |

**SISTEMA DE INFORMACION AL QUE ACCEDE: NOTE**

**BREVE DESCRIPCION FUNCIONAL DEL WEBSERVICE (información extendida en documento de ámbito funcional)**

WebService que permite a aplicaciones externas interactuar con la aplicación que gestiona las notificaciones telemáticas (NOTE)

**URLS DE ACCESO:**

Desarrollo: http://deswebservices.madrid.org/note\_ws/services

Producción: http://prowebservices.madrid.org/note\_ws/services

**RELACIÓN DE MÉTODOS:**

El web service NOTE provee varios métodos.

A continuación se muestra una recopilación de todos ellos en el orden que se recomienda sean invocados.

Los métodos en **ROJO** no se evolucionan, la funcionalidad que contemplan

está incluida en el método anteriormente mencionado en **Negrita**.

**existeNIF:** Comprueba si el interesado está dado de alta en el servicio

de notificaciones telemáticas, si es así, devuelve los datos del abonado

activo. En caso de que no tenga email informado, de devuelve como que no existiera. Esta comprobación es redundante, ya que en este caso autorizado y aceptado = ‘N’, con lo que no se devolvería igualmente.

* **existeAbonado:** Comprueba si el interesado está dado de alta

en el servicio de notificaciones telemáticas**.**

**existeNIFPostal:** Comprueba si el interesado está dado de alta en el servicio de notificaciones telemáticas, si no es así lo da de alta, aunque con los campos de AUTERIZADO y AUTENITCADO con el valor No. En caso de que ya existiera pero no estuviera informado el email, lo actualiza

**enviarNotificacion:** Envía una notificación y devuelve el número de

notificación y el estado de la misma (código y descripción).

*(Nota: si la aplicación está integrada con la plataforma Notific@, también se enviará a esta plataforma).*

**enviarNotificacionAdjuntoUnico:** Envío de una notificación

con un único adjunto.

**enviarNotificacionPostal:** Envía una notificación vía postal y devuelve el número de notificación y el estado de la misma (código y descripción).

**preAltaAbonado:** Realiza el Pre-Alta de un Abonado en el servicio de

notificaciones con los datos aportados, devuelve el resultado de dicha

operación.

**obtenerNotificacionesPendientesTotal:** Devuelve las notificaciones

pendientes con toda su información, generadas por una aplicación

ordenadas por fecha de alta en el sistema. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerNotificacionesPendientes:** Devuelve las notificaciones

pendientes generadas por una aplicación, se devuelven las notificaciones con estado puesta a disposición, error de entrega,

rechazada o rechazada automáticamente por el sistema

ordenadas por fecha de alta en el sistema. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerNotificacionesPendientesConAsunto:** Método con las

misma funcionalidad, entrada y salida que

obtenerNotificacionesPendientes, pero incluye el campo Asunto

de NOTE en su salida. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerNotificacionesPendientesTotalAcuse:** Devuelve las notificaciones pendientes con toda su información, generadas por una aplicación ordenadas por fecha de alta en el sistema, incluye el número de referencia y el código de archivo del acuse si corresponde. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerNotificacionesPendientesTotalGestor:** Devuelve las notificaciones pendientes con toda su información, generadas por una aplicación ordenadas por fecha de alta en el sistema, incluido el número de referencia del acuse si corresponde. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerNotificacion:** Devuelve una notificación generada por una

aplicación, se devuelve la notificacion con toda la información de la

misma. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**obtenerAcuseRecibo:** Devuelve el PDF del Acuse de recibo

de una notificación correspondiente a una aplicación

determinada.

**generarPDF:** Devuelve el PDF del Acuse de recibo de una

notificación. Uso interno para versiones anteriores.

**obtenerNotificacionAcuse:** Devuelve una notificación generada por una aplicación, se devuelve la notificacion con toda la información de la

misma, incluye el número de referencia y el código de archivo del Acuse si corresponde. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**eliminarNotificacionesPendientes:** Elimina las notificaciones

pendientes de la cola de mensajes generadas por una aplicación.

**enviarRemesa:** Envía una remesa de notificaciones.

***enviarRemesaAdjuntoUnico:*** *Envía una remesa de notificaciones con un único adjunto en cada notificación****.***

**obtenerDUE:** Devuelve la Dirección Electrónica Única asociada a los

datos de un interesado.

**enviarNotificacionNotifica:** Envía una notificación a la plataforma Notific@, devuelve el número de notificación y el estado de la misma.

**consultaNotificacion:** Devuelve la información de la notificación o notificaciones que coincidan con los parámetros de entrada. En caso de que haya sido rechazada automáticamente postal, devuelve informado el campo dsestado\_postal.

**enviarNotificacionGfic:** Envía una notificación y devuelve el número de notificación y el estado de la misma. Si la notificación incluye adjuntos, no se envía el adjunto como tal, sino los parámetros que lo identifican.

***AdviserWS***

El web service NOTE además tiene métodos tras la integración con Notific@, de utilización interna.

**datadoOrganismo:** Realiza la recepción del Datado de un envío.

**certificacionOrganismo:** Realiza la recepción de la Certificación de un envío.

***AdviserWsV2***

Nuevo servicio de la versión 2.0 del Adviser, corresponde a la integración con Notific@, de utilización interna.

 **sincronizarEnvio:** Realiza la recepción del datato y certifiado de un envío. Se realiza las modificaciones necesarias para poder recibir los diferentes cambios de estado provocados por información de envíos postales

***LEMA***

Nuevos servicios para la integración con LEMA, de utilización interna.

 **localizaNotificaciones:** Realiza la recuperación de notificaciones que están pendiestes de acceso, dirigidas a la CM.

 **peticionNotificacion:** Realiza la comparecencia de las nofificaciones y recupera todos los documentos asociados a ellas.

 **enviarNotificacionUnidadRegistro:** Realiza el envío en registro de los documentos de una notificación a otra unidad de registro.

 **recepcionNotificacionUnidadRegistro:** Realiza la recepción de los documentos de una notificación en la unidad de registro.

**DESCRIPCION DE LOS METODOS:**

|  |
| --- |
| **existeNIF** |

Comprueba si el interesado está dado de alta en el servicio de notificaciones telemáticas, si existe, devuelve los datos del abonado activo, y la información de estar autorizado o no para poder enviarle notificaciones. En caso de que no tenga email informado, de devuelve como que no existiera. Esta comprobación es redundante, ya que en este caso autorizado y aceptado = ‘N’, con lo que no se devolvería igualmente.

*String* ***existeNIF*** *(String codigoApp, String nifCif)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| nifCif | Nº de Identificación del abonado | String | SI (\*) |

**Comentarios:**

* El nº de identificación tiene que estar completo, con los ceros a la izquierda si es que los lleva.

**Retorno:**

Parámetro existeNIF:

* 1: Existe el abonado
* 0: No existe el abonado
* 4: El abonado está inactivo por unificación
* XML esquema (aplicación sin integración con Notific@)

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="existeNIF" type="xs:int"/>

<xs:attribute name="cdAbonado" type="xs:string"/>

<xs:attribute name="tlNombre" type="xs:string"/>

<xs:attribute name="tlTelefonoMovil" type="xs:string"/>

<xs:attribute name="tlEmail" type="xs:string"/>

<xs:attribute name="tlDireccion" type="xs:string"/>

<xs:attribute name="cdPostal" type="xs:string"/>

<xs:attribute name="fcAlta" type="xs:string"/>

<xs:attribute name="fcBorrado" type="xs:string"/>

<xs:attribute name="tlNifcif" type="xs:string"/>

<xs:attribute name="cdTipodoc" type="xs:string"/>

<xs:attribute name="cdProv" type="xs:string"/>

<xs:attribute name="cdMuni" type="xs:string"/>

<xs:attribute name="itAutorizado" type="xs:string"/>

</xs:complexType>

</xs:element>

</xs:schema>

* XML esquema (aplicación con integración en Notific@)

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="existeNIF" type="xs:int"/>

<xs:attribute name="cdAbonado" type="xs:string"/>

<xs:attribute name="tlNombre" type="xs:string"/>

<xs:attribute name="tlTelefonoMovil" type="xs:string"/>

<xs:attribute name="tlEmail" type="xs:string"/>

<xs:attribute name="tlDireccion" type="xs:string"/>

<xs:attribute name="cdPostal" type="xs:string"/>

<xs:attribute name="fcAlta" type="xs:string"/>

<xs:attribute name="fcBorrado" type="xs:string"/>

<xs:attribute name="tlNifcif" type="xs:string"/>

<xs:attribute name="cdTipodoc" type="xs:string"/>

<xs:attribute name="cdProv" type="xs:string"/>

<xs:attribute name="cdMuni" type="xs:string"/>

<xs:attribute name="fcPrealta" type="xs:string"/>

<xs:attribute name="itAceptado" type="xs:string"/>

<xs:attribute name="itAutorizado" type="xs:string"/>

</xs:complexType>

</xs:element>

</xs:schema>

**Ejemplos de los tipos de respuestas del servicio *existeNIF****:*

1. **Aplicación sin integración con Notific@**

* **EXISTE ABONADO NO AUTORIZADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>1</existeNIF>

<cdAbonado>XXXXXXXXYAPE</cdAbonado>

<tlNombre>APELLIDO1 APELLIDO2 NOMBRE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>correo@mail.com</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado>N</itAutorizado>

</abonado>

* **EXISTE ABONADO AUTORIZADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>1</existeNIF>

<cdAbonado>AXXXXXXXXEMP</cdAbonado>

<tlNombre>EMPRESA</tlNombre>

<tlTelefonoMovil>666666666</tlTelefonoMovil>

<tlEmail>pruebas\_usuario@madrid.org</tlEmail>

<tlDireccion>CALLE 2</tlDireccion>

<cdPostal>28000</cdPostal>

<fcAlta>01/01/2018 01:01:01</fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>AXXXXXXXX</tlNifcif>

<cdTipodoc>2</cdTipodoc>

<cdProv>28</cdProv>

<cdMuni>123</cdMuni>

<itAutorizado>S</itAutorizado>

</abonado>

* **NO EXISTE ABONADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>0</existeNIF>

<cdAbonado></cdAbonado>

<tlNombre></tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail></tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado></itAutorizado>

</abonado>

* **EXISTE ABONADO INACTIVO POR UNIFICACIÓN**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>4</existeNIF>

<cdAbonado>XXXXXXXXYAPE</cdAbonado>

<tlNombre>APELLIDO1 APELLIDO2 NOMBRE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>correo@mail.com</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado></itAutorizado>

</abonado>

1. **Aplicación con integración con Notific@**

* **EXISTE ABONADO NO AUTORIZADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>1</existeNIF>

<cdAbonado>XXXXXXXXYAPE</cdAbonado>

<tlNombre>APELLIDO1 APELLIDO2 NOMBRE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>correo@mail.com</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<fcPrealta>01/01/2018 10:11:12</fcPrealta>

<itAceptado>N</itAceptado>

<itAutorizado>N</itAutorizado>

</abonado>

* **EXISTE ABONADO AUTORIZADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>1</existeNIF>

<cdAbonado>AXXXXXXXXEMP</cdAbonado>

<tlNombre>EMPRESA</tlNombre>

<tlTelefonoMovil>666666666</tlTelefonoMovil>

<tlEmail>pruebas\_usuario@madrid.org</tlEmail>

<tlDireccion>CALLE 2</tlDireccion>

<cdPostal>28000</cdPostal>

<fcAlta>01/01/2018 12:59:59</fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>AXXXXXXXX</tlNifcif>

<cdTipodoc>2</cdTipodoc>

<cdProv>28</cdProv>

<cdMuni>123</cdMuni>

<fcPrealta></fcPrealta>

<itAceptado>N</itAceptado>

<itAutorizado>S</itAutorizado>

</abonado>

* **NO EXISTE ABONADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>0</existeNIF>

<cdAbonado></cdAbonado>

<tlNombre></tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail></tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<fcPrealta></fcPrealta>

<itAceptado></itAceptado>

<itAutorizado></itAutorizado>

</abonado>

* **EXISTE ABONADO INACTIVO POR UNIFICACIÓN**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>4</existeNIF>

<cdAbonado>XXXXXXXXYAPE</cdAbonado>

<tlNombre>APELLIDO1 APELLIDO2 NOMBRE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>correo@mail.com</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado></itAutorizado>

</abonado>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **existeNIFPostal** |

Comprueba si el interesado está dado de alta en el servicio de notificaciones telemáticas, si existe, devuelve los datos del abonado activo, y la información de estar autorizado o no para poder enviarle notificaciones. En caso de que no existiera se dará de alta con los campos ITAutorizado, ITAceptado = ‘N’, de manera que se garantice la correcta compatibilidad con el sistema de notificaciones telemáticas actual y su gestión de abonados. Además, en el caso de que se informe del campo email

*String* ***existeNIFPostal*** *(String codigoApp, String nifCif,String nombre, String cdTipoDoc, String email, String cdProcedimiento)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| nombre | Nombre del interesado | Cadena | SI |
| nifCif | NIF/CIF del interesado | Cadena | SI |
| cdTipoDoc | Tipo de documento (1 NIF 2 CIF) | Numérico | SI |
| email | Dirección de correo del interesado | Cadena | No |
| cdprocedimiento | Código de procedimiento | Cadena | SI |

**Comentarios:**

* Primeramente, consultará en la tabla NOTE\_APLICACION si la aplicación está dada de alta en NOTE y su fecha baja es nula. Si no es así se devolverá un error
* Después se comprobará que el procedimiento tenga CIE asociado, consultado que el campo CIE sea igual a ‘S’ en la tabla IPAE\_PROCEDIMIENTO
* El email no es obligatorio, pero si se informa, se empleará para actualizar el registro en caso de que no esté informado.

**Retorno:**

Parámetro existeNIF:

* 1: Existe el abonado o no existía y con la información facilitada se ha podido dar de alta
* 0: No existe el abonado o no existía y con la información facilitada no se ha podido dar de alta
* 4: El abonado está inactivo por unificación
* XML esquema (aplicación con integración en Notific@)

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://servicio.note\_ws">

<soapenv:Header/>

<soapenv:Body>

<ser:existeNIFPostal soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<codigoApp xsi:type="soapenc:string" />

<nifCif xsi:type="soapenc:string" />

<nombre xsi:type="soapenc:string" />

<cdTipoDoc xsi:type="soapenc:int" />

<email xsi:type="soapenc:string" />

<cdProcedimiento xsi:type="soapenc:long" />

</ser:existeNIFPostal>

</soapenv:Body>

</soapenv:Envelope>

* **EXISTE EL USUARIO PREVIAMENTE / NO EXISTE PERO SE HA PODIDO DAR DE ALTA**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIFPostal.xsd">

<existeNIF>1</existeNIF>

<cdAbonado>12345678B PEP</cdAbonado>

<tlNombre>PEPE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>pepe@prueba.es</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>12345678B</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado>N</itAutorizado>

</abonado>

* **NO EXISTE EL USUARIO PREVIAMENTE Y NO SE PUEDE DAR DE ALTA**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIFPostal.xsd">

<existeNIF>0</existeNIF>

<cdAbonado></cdAbonado>

<tlNombre></tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail></tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>S</tlNifcif>

<cdTipodoc>0</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado></itAutorizado>

</abonado></abonado>

* **NO EXISTE ABONADO Y NO PUEDE DARSE DE ALTA CON LA INFORMACION QUE SE HA DADO**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIF.xsd">

<existeNIF>0</existeNIF>

<cdAbonado></cdAbonado>

<tlNombre></tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail></tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>XXXXXXXXY</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<fcPrealta></fcPrealta>

<itAceptado></itAceptado>

<itAutorizado></itAutorizado>

</abonado>

* **EXISTE EL USUARIO PREVIAMENTE PERO ESTÁ UNFICADO Y NO SE HA PODIDO DAR DE ALTA**

<abonado xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/existeNIFPostal.xsd">

<existeNIF>4</existeNIF>

<cdAbonado>12345678B PEP</cdAbonado>

<tlNombre>PEPE</tlNombre>

<tlTelefonoMovil></tlTelefonoMovil>

<tlEmail>pepe@prueba.es</tlEmail>

<tlDireccion></tlDireccion>

<cdPostal></cdPostal>

<fcAlta></fcAlta>

<fcBorrado></fcBorrado>

<tlNifcif>12345678B</tlNifcif>

<cdTipodoc>1</cdTipodoc>

<cdProv></cdProv>

<cdMuni></cdMuni>

<itAutorizado></itAutorizado>

</abonado>

* **EL PROCEDIMIENTO NO TIENE CIE ASOCIADO**

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 02 ERR003 - El Procedimiento Administrativo no tiene CIE asociado.</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides06.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

* **EL PROCEDIMIENTO NO EXISTE**

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 02 ERR002 -No existe el Procedimiento Administrativo.</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides05.madrid.org</ns1:hostname>

</detail></soapenv:Fault>

* **LA APLICACION NO TIENE PERMISO**

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 02 La APLICACION ER2EG no tiene permiso de acceso al servicio de notificaciones teleméticas.</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides05.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

**Excepciones:** ICMException. Excepción en caso de que la aplicación o el procedimiento no exista o este no tenga CIE asociado o se produzca un error interno.

|  |
| --- |
| **existeAbonado** |

Comprueba si el interesado está dado de alta en el servicio de notificaciones telemáticas.

*int* ***existeAbonado****(String codigoApp, String apellidos, String nifCif)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| apellidos | Apellidos del abonado | String | NO (\*) |
| nifCif | Nº de Identificación del abonado | String | SI (\*) |

**Comentarios:**

* Los apellidos son obligatorios en el caso de que el Abonado sea Persona Física.
* El nº de identificación tiene que estar completo, con los ceros a la izquierda si es que los lleva.

**Retorno:**

* 1: Existe el abonado
* 0: No existe el abonado

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **enviarNotificacion** |

Procedimiento que envía una notificación y devuelve el número de notificación y el estado de la misma.

*String* **enviarNotificacion** *(String codigoApp, String codigoOrigen, String codigoOrigenInt, String cifrada, String refAportacion, String uniAdmin, String servicio, String asunto, String cuerpo, String codigoAbonado, String [] nombresAdjunto, String [] datosAdjunto, String cdProcedimiento)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicación | String | SÍ |
| codigoOrigen | Referencia | String | SI |
| codigoOrigenInt | Referencia interna | String | SI |
|  |  |  |  |
| cifrada | Cifrado del cuerpo del mensaje: Valores posibles:  true/false o 1/0 | String | SI |
| refAportacion | Nº de Referencia de la aportación | String | NO |
| uniAdmin | Descripción Unidad Administrativa | String | SI |
| servicio | Descripción servicio | String | SI |
| asunto | Descripción del asunto | String | SI |
| cuerpo | Cuerpo de la notificación | String | NO |
| codigoAbonado | DUE del interesado | String | SI |
| nombresAdjunto | Nombre de los adjuntos | String [] | NO |
| datosAdjunto | Datos de los adjuntos en base64 | String [] | NO |
| cdProcedimiento | Código del procedimiento | String | NO |

**Comentarios:**

El nombre de cada adjunto debe contener la extensión.

Los parámetros de codigo origen son textos libres.

**Retorno:**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="enviarNotificacion">

<xs:complexType>

<xs:attribute name="nmnotificacion" type="xs:string"/>

<xs:attribute name="cdestadonotificacion" type="xs:string"/>

<xs:attribute name="dsestadonotificacion" type="xs:string"/>

</xs:complexType>

</xs:element>

</xs:schema>

**Ejemplos de los tipos de respuestas del servicio *enviarNotificacion****:*

* **1-PUESTA A DISPOSICION (OK)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion>XXXXXX</nmnotificacion>

<cdestadonotificacion>**1**</cdestadonotificacion>

<dsestadonotificacion>**PUESTA A DISPOSICION**</dsestadonotificacion>

</enviarNotificacion>

* **5-ERROR ENTREGA (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion> XXXXXX </nmnotificacion>

<cdestadonotificacion>**5**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENTREGA**</dsestadonotificacion>

</enviarNotificacion>

* **0-No ha sido aceptada la DEH del interesado (NIF), en estos momentos no es posible enviar la Notificación. (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion></nmnotificacion>

<cdestadonotificacion>**0**</cdestadonotificacion>

<dsestadonotificacion>**No ha sido aceptada la DEH del interesado XXXXXXXXX, en estos momentos no es posible enviar la Notificación.**</dsestadonotificacion>

</enviarNotificacion>

* **9-ERROR ENVIO: (Descripción del error) (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion></nmnotificacion>

<cdestadonotificacion>**9**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENVIO: Se ha producido un error …** </dsestadonotificacion>

</enviarNotificacion>

* **10-ERROR ENVIO: (Descripción del error) (ERROR)**

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 02 ERR002 -No existe el Procedimiento Administrativo.</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides06.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **enviarNotificacionPostal** |

Procedimiento que envía una notificación vía postal y devuelve el número de notificación y el estado de la misma.

*String* ***enviarNotificacionPostal****(String codigoApp, String codigoOrigen, String codigoOrigenInt, String cifrada, String refAportacion, String uniAdmin, String servicio, String asunto, String cuerpo, String codigoAbonado, String [] nombresAdjunto, String [] datosAdjunto,Long cdProcedimiento,String linea1, String linea2,String codigoPostal, Boolean normalizado)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicación | String | SÍ |
| codigoOrigen | Referencia | String | SI |
| codigoOrigenInt | Referencia interna | String | SI |
| cifrada | Cifrado del cuerpo del mensaje: Valores posibles:  true/false o 1/0 | String | SI |
| refAportacion | Nº de Referencia de la aportación | String | NO |
| uniAdmin(\*) | Descripción Unidad Administrativa | String | SI |
| servicio | Descripción servicio | String | SI |
| asunto | Descripción del asunto | String | SI |
| cuerpo | Cuerpo de la notificación | String | NO |
| codigoAbonado | DUE del interesado | String | SI |
| nombresAdjunto | Nombre de los adjuntos | String [] | Si \* |
| datosAdjunto | Datos de los adjuntos en base64 | String [] | Si \* |
| codigoPostal | Código postal | String | Si |
| linea1 | Línea 1 de la dirección de entrega del envío sin normalizar. | String | Si |
| linea2 | Línea 2 de la dirección de entrega del envío sin normalizar. | String | Si |
| cdProcedimiento | Código del procedimiento | Long | Si |
| normalizado | Indica si el documento está normalizado (\*\*) | Booleano | Si |

**Comentarios:**

El nombre de cada adjunto debe contener la extensión.

Los parámetros de codigo origen son textos libres.

(\*) Es necesario pasar un solo adjunto en base64 y que sea un pdf

(\*\*) En caso de que el documento reserve en la primera página el espacio en blanco necesario para introducir la ventanilla del sobre (NCC, titular, destinatario, domicilio) se debe rellenar con true. En caso contrario debe rellenarse con false y el CIE adjuntará una página adicional con la ventana para el ensobrado, incrementando el coste del servicio por la página impresa y por el peso adicional del servicio postal.

Un documento está normalizado si en la primera página ya se deja el espacio necesario para imprimir la dirección del destinatario que se manda en el procedimiento. En caso de que no lo esté el CIE imprimirá esa primera página

**Retorno:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<ns1:enviarNotificacionPostalResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://servicio.note\_ws">

<enviarNotificacionPostalReturn xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"><![CDATA[<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion>160348</nmnotificacion>

<cdestadonotificacion>1</cdestadonotificacion>

<dsestadonotificacion>PUESTA A DISPOSICION</dsestadonotificacion>

</enviarNotificacion>]]></enviarNotificacionPostalReturn>

</ns1:enviarNotificacionPostalResponse>

</soapenv:Body>

</soapenv:Envelope>

**Ejemplos de los tipos de respuestas del servicio *enviarNotificacion****:*

* **1-PUESTA A DISPOSICION (OK)**

<enviarNotificacionPostal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacionPostal.xsd">

<nmnotificacion>XXXXXX</nmnotificacion>

<cdestadonotificacion>**1**</cdestadonotificacion>

<dsestadonotificacion>**PUESTA A DISPOSICION**</dsestadonotificacion>

</enviarNotificacionPostal >

* **2-NO SE HA PASADO UN ADJUNTO**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 - Debe enviar al menos un ficheros pdf</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides05.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

**2- SE HA PASADO UN ADJUNTO PERO NO ES UN PDF O NO SE HA PASADO EN BASE64**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 - Ha excedido el tamanio maximo de ficheros. El total no puede superar 15 Megas</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides06.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

**3- SE HA PASADO UNO O MAS ADJUNTOS PERO SE HA EXCEDIDO EL TAMAÑO MAXIMO**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<ns1:enviarNotificacionPostalResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://servicio.note\_ws">

<enviarNotificacionPostalReturn xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"><![CDATA[<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion>160956</nmnotificacion>

<cdestadonotificacion>1</cdestadonotificacion>

<dsestadonotificacion>PUESTA A DISPOSICION</dsestadonotificacion>

</enviarNotificacion>]]></enviarNotificacionPostalReturn>

</ns1:enviarNotificacionPostalResponse>

</soapenv:Body>

</soapenv:Envelope>

* **4-ERROR ENTREGA (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion> XXXXXX </nmnotificacion>

<cdestadonotificacion>**5**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENTREGA**</dsestadonotificacion>

</enviarNotificacion>

* **5-ERROR ENVIO: (Descripción del error) (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion></nmnotificacion>

<cdestadonotificacion>**9**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENVIO: Se ha producido un error …** </dsestadonotificacion>

</enviarNotificacion>

* **6-ERROR ENVIO: Enviar Notificación Postal con Procedimiento Administrativo sin CIE**

*<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">*

*<soapenv:Body>*

*<soapenv:Fault>*

*<faultcode>soapenv:Server.userException</faultcode>*

*<faultstring>java.rmi.RemoteException: 02 02 ERR003 - El Procedimiento Administrativo no tiene CIE asociado.</faultstring>*

*<detail>*

*<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">DESKTOP-ALDK1UG</ns1:hostname>*

*</detail>*

*</soapenv:Fault>*

*</soapenv:Body>*

*</soapenv:Envelope>*

* **7-ERROR ENVIO: Enviar Notificación con Postal Procedimiento Administrativo no existente en IPAE**

*<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">*

*<soapenv:Body>*

*<soapenv:Fault>*

*<faultcode>soapenv:Server.userException</faultcode>*

*<faultstring>java.rmi.RemoteException: 02 02 ERR002 -No existe el Procedimiento Administrativo.</faultstring>*

*<detail>*

*<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">DESKTOP-ALDK1UG</ns1:hostname>*

*</detail>*

*</soapenv:Fault>*

*</soapenv:Body>*

*</soapenv:Envelope>*

* **8-ERROR ENVIO: Enviar Notificación con abonado no dado de alta**

*<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">*

*<soapenv:Body>*

*<ns1:enviarNotificacionResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://servicio.note\_ws">*

*<enviarNotificacionReturn xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"><![CDATA[<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">*

*<nmnotificacion></nmnotificacion>*

*<cdestadonotificacion>9</cdestadonotificacion>*

*<dsestadonotificacion>ERROR ENVIO:No existe el Procedimiento Administrativo.</dsestadonotificacion>*

*</enviarNotificacion>]]></enviarNotificacionReturn>*

*</ns1:enviarNotificacionResponse>*

*</soapenv:Body>*

*</soapenv:Envelope>*

|  |
| --- |
| **enviarNotificacionAdjuntoUnico** |

Procedimiento envía una notificación que como mucho tiene un adjunto.

*void* **enviarNotificacionAdjuntoUnico** *(String codigoApp, String codigoOrigen, String codigoOrigenInt, String cifrada, String refAportacion, String uniAdmin, String servicio, String asunto, String cuerpo, String codigoAbonado, String [] nombresAdjunto, String [] datosAdjunto)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicación | String | SÍ |
| codigoOrigen | Referencia | String | SI |
| codigoOrigenInt | Referencia interna | String | SI |
| cifrada | Cifrado del cuerpo del mensaje: Valores posibles:  true/false o 1/0 | String | SI |
| refAportacion | Nº de Referencia de la aportación | String | NO |
| uniAdmin | Descripción Unidad Administrativa | String | SI |
| servicio | Descripción servicio | String | SI |
| asunto | Descripción del asunto | String | SI |
| cuerpo | Cuerpo de la notificación | String | NO |
| codigoAbonado | DUE del interesado | String | SI |
| nombresAdjunto | Nombre de los adjuntos | String | NO |
| datosAdjunto | Datos de los adjuntos en base64 | String | NO |

**Comentarios:**

El nombre de cada adjunto debe contener la extensión.

**Retorno:**

N/A

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **preAltaAbonado** |

Realiza el alta de un abonado en el servicio de notificaciones con los datos aportados, devuelve el resultado de dicha operación.

En el caso de que se hubiera dado de alta desde existeNIF, (campo NOTE\_Postal =’P’. Si se invoca a este método se actualizará el email, el campo medio y si comprobando el medio, aplicara, se cambiará el campo autorizdo.

*String* ***preAltaAbonado****(String cdApp, String nombre, String nifCif, String tlEmail, String tlMedioPreAlta)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| cdApp | Código de aplicación | String | SI |
| nombre | Nombre del interesado | String | SI |
| nifCif | NIF/CIF del interesado | String | SI |
| email | Dirección de correo del interesado | String | SI |
| medioPreAlta | Nombre del medio que lo realiza | String | SI |

**Retorno:**

ParámetropreAltaAbonado:

* 0 : Correcto,
* -1: Error (descripción del error)
* 1: Ya existe el abonado con NIF/CIF: (NIF/CIF)
* 2 : La APLICACION no tiene permiso de acceso al servicio de notificaciones telemáticas:(CDAPP)
* 3: Error (devuelto por la librería de sistemas al validar el NIF/CIF)
* 4: Error al validar el NIF/CIF: (nif/cif)
* 5: Error datos de preAltaAbonado:
  + NIF/CIF no informado o
  + email no informado o
  + nombre no informado o
  + interesadao ya existente pero con su DEH inactiva
* 6: Error abonado inactivo por unificación
* XML esquema

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<xs:schema xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* elementFormDefault=*"qualified"*>

<xs:element name=*"preAltaAbonado"*>

<xs:complexType>

<xs:attribute name=*"cdStatus"* type=*"xs:string"*/>

<xs:attribute name=*"dsStatus"* type=*"xs:string"*/>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionesPendientesTotal** |

Devuelve las notificaciones pendientes generadas por una aplicación, se devuelven las notificaciones con toda la información de la misma.

*String* ***obtenerNotificacionesPendientesTotal*** *(String cdAplicacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="numero\_abonado" type="xs:string" />

<xs:element name="codigo\_abonado" type="xs:string" />

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="fecha\_notificacion" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="dsestado\_postal" type="xs:string"/>

l <xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="fecha\_notificacion"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref=" dsestado\_postal"/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionesPendientes** |

Devuelve las notificaciones pendientes generadas por una aplicación, se devuelven las notificaciones con estado puesta a disposición, error de entrega, rechazada o rechazada automáticamente por el sistema ordenadas por fecha de alta en el sistema.

*String* ***obtenerNotificacionesPendientes****(String cdAplicacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| cdAplicacion | Código de aplicación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="codigo\_abonado" type="xs:string" />

</xs:complexType>

</xs:element>

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionesPendientesConAsunto** |

Método con las misma funcionalidad, entrada y salida que obtenerNotificacionesPendientes, pero incluye el campo Asunto de NOTE en su salida

*String* ***obtenerNotificacionesPendientesConAsunto****(String cdAplicacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="codigo\_abonado" type="xs:string" />

</xs:complexType>

</xs:element>

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionesPendientesTotalAcuse** |

Devuelve las notificaciones pendientes generadas por una aplicación, se devuelven las notificaciones con toda la información de la misma, junto con el número de referencia y el código de archivo del acuse de recibo si corresponde.

*String* ***obtenerNotificacionesPendientesTotalAcuse*** *(String cdAplicacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="numero\_abonado" type="xs:string" />

<xs:element name="codigo\_abonado" type="xs:string" />

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="fecha\_notificacion" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="dsestado\_postal" type="xs:string"/>

<xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="cddocum" type="xs:string"/>

<xs:element name="codarchivo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="fecha\_notificacion"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref=" dsestado\_postal"/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

<xs:element ref="cddocum"/>

<xs:element ref="codarchivo"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionesPendientesTotalGestor** |

Devuelve las notificaciones pendientes generadas por una aplicación, se devuelven las notificaciones notificaciones con toda la información de la misma, junto con el número de referencia del acuse de recibo si corresponde.

*String* ***obtenerNotificacionesPendientesTotalGestor*** *(String cdAplicacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="numero\_abonado" type="xs:string" />

<xs:element name="codigo\_abonado" type="xs:string" />

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="fecha\_notificacion" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="dsestado\_postal" type="xs:string"/>

<xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="cddocum" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="fecha\_notificacion"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref=" dsestado\_postal"/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

<xs:element ref="cddocum"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacion** |

Devuelve toda la información de una notificación perteneciente a una aplicación a partir de su número de identificación, uno de los datos que devuelve es el propio acuse de recibo si corresponde con su estado.

*String* ***obtenerNotificacion*** *(String cdAplicacion, String nmnotificacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| nmnotificacion | Número de notificación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="numero\_abonado" type="xs:string"/>

<xs:element name="codigo\_abonado" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="fecha\_notificacion" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="fecha\_notificacion"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista, o la notificación no pertenezca a la aplicación o se produzca un error interno.

|  |
| --- |
| **obtenerNotificacionAcuse** |

Devuelve toda la información de una notificación perteneciente a una aplicación a partir de su número de identificación, uno de los datos que devuelve es el propio acuse de recibo si corresponde con su estado, junto con su número de referencia y códio de archivo.

*String* ***obtenerNotificacion*** *(String cdAplicacion, String nmnotificacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| nmnotificacion | Número de notificación | String | SI |

**Retorno:**

* XML con la lista de notificaciones pendientes, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="numero\_abonado" type="xs:string"/>

<xs:element name="codigo\_abonado" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="firma\_error" type="xs:string"/>

<xs:element name="firma\_lectura" type="xs:string"/>

<xs:element name="firma\_puesta\_disposicion" type="xs:string"/>

<xs:element name="firma\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="fecha\_notificacion" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="dsestado\_postal" type="xs:string"/>

<xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="cddocum" type="xs:string"/>

<xs:element name="codarchivo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="firma\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="firma\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="firma\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="firma\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="fecha\_notificacion"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref=" dsestado\_postal"/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

<xs:element ref="cddocum"/>

<xs:element ref="codarchivo"/>

</xs:sequence>

<xs:attribute name="identificador" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista, o la notificación no pertenezca a la aplicación o se produzca un error interno.

|  |
| --- |
| **obtenerAcuseRecibo** |

Método que devuelve el Acuse de recibo de una notificación que correspondiente a una aplicación determinada.

*String* ***obtenerAcuseRecibo*** *(String codigoApp, String nmnotificacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| nmnotificacion | Identificador notificación | String | SI |

**Retorno:**

* XML esquema

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="acuse\_recibo">

<xs:complexType>

<xs:attribute name="cdStatus" type="xs:string"/>

<xs:attribute name="dsStatus" type="xs:string"/>

<xs:attribute name="notificacion" type="xs:string"/>

<xs:attribute name="itFirmado" type="xs:boolean"/>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o la notificación no pertenezca a la aplicación indicada o se encuentre en un estado al que no le corresponda acuse de recibo o se produzca un error interno.

|  |
| --- |
| **generarPDF** |

Método que devuelve el PDF del Acuse de recibo de una notificación de un abonado.

*String***generarPDF***(String nmNotificacion, String nmAbonado, String accion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| nmnotificacion | Identificador notificación | String | SI |
| nmAbonado | Número de abonado | String | SI |
| accion | Accion para la que se necesita: firma o consulta | String | SI |

**Retorno:**

* PDF: Acuse de recibo de la notificación en formato base64.

**Excepciones:** ICMException. Excepción en caso de que la notificación no exista o se produzca un error interno.

|  |
| --- |
| **eliminarNotificacionesPendientes** |

Procedimiento que elimina las notificaciones pendientes de la aplicación que solicita el servicio. Se comprobarán que existe la aplicación, la notificación y que la notificación pertenece a dicha aplicación; en caso de que no se cumpla alguna de las restricciones anteriores se elevará una excepción.

*void* **eliminarNotificacionesPendientes** *(String codigoApp, String[] identificadoresNotificaciones)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicación | String | SÍ |
| identificadoresNotificaciones | Identificadores de las notificaciones a eliminar | String[] | SÍ |

**Retorno:**

N/A

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista, no tenga notificaciones pendientes o se produzca un error interno

|  |
| --- |
| **enviarRemesa** |

Procedimiento envía una remesa de notificaciones, las notificaciones se quedarán en una tabla temporal NOTE\_NOTIFICACION\_TMP hasta la Puesta en Disposición mediante la ejecución diaria de un proceso batch

*void* **enviarRemesa** *(String remesa)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| Remesa | Remesa de notificaciones en formato XML | String | SI |

**Retorno:**

N/A

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno

**Esquema de la remesa:**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="codigo\_abonado" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="adjunto">

<xs:complexType>

<xs:simpleContent>

<xs:extension base="xs:string">

<xs:attribute name="nombre" type="xs:string" use="required"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

</xs:element>

<xs:element name="adjuntos">

<xs:complexType>

<xs:sequence>

<xs:element ref="adjunto" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="unidad\_administrativa"/>

<xs:element ref="servicio"/>

<xs:element ref="asunto"/>

<xs:element ref="cuerpo"/>

<xs:element ref="abonado"/>

<xs:element ref="adjuntos"/>

</xs:sequence>

<xs:attribute name="codigo\_origen" type="xs:string" use="required"/>

<xs:attribute name="codigo\_origen\_interno" type="xs:string" use="required"/>

<xs:attribute name="cifrada" type="xs:boolean" use="required"/>

<xs:attribute name="ref\_aportacion" type="xs:string" use="optional"/>

</xs:complexType>

</xs:element>

<xs:element name="remesa">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="codigo\_app" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="servicio" type="xs:string"/>

<xs:element name="unidad\_administrativa" type="xs:string"/>

</xs:schema>

|  |
| --- |
| ***enviarRemesaAdjuntoUnico*** |

Procedimiento envía una remesa de notificaciones con un único adjunto, las notificaciones se quedarán en una tabla temporal NOTE\_NOTIFICACION\_TMP hasta la Puesta en Disposición mediante la ejecución diaria de un proceso batch

*void* **enviarRemesaAdjuntoUnico** *(String remesa)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| Remesa | Remesa de notificaciones en formato XML | String | SI |

**Retorno:**

N/A

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno

**Esquema de la remesa:**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="abonado">

<xs:complexType>

<xs:attribute name="codigo\_abonado" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="adjunto">

<xs:complexType>

<xs:simpleContent>

<xs:extension base="xs:string">

<xs:attribute name="nombre" type="xs:string" use="required"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

</xs:element>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="unidad\_administrativa"/>

<xs:element ref="servicio"/>

<xs:element ref="asunto"/>

<xs:element ref="cuerpo"/>

<xs:element ref="abonado"/>

<xs:element ref="adjunto" minOccurs="0" maxOccurs="1"/>

</xs:sequence>

<xs:attribute name="codigo\_origen" type="xs:string" use="required"/>

<xs:attribute name="codigo\_origen\_interno" type="xs:string" use="required"/>

<xs:attribute name="cifrada" type="xs:boolean" use="required"/>

<xs:attribute name="ref\_aportacion" type="xs:string" use="optional"/>

</xs:complexType>

</xs:element>

<xs:element name="remesa">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="codigo\_app" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="servicio" type="xs:string"/>

<xs:element name="unidad\_administrativa" type="xs:string"/>

</xs:schema>

|  |
| --- |
| **obtenerDUE** |

Devuelve la Dirección Electrónica Única asociada a los datos de un interesado.

*String* ***obtenerDUE*** *(String codigoApp, String apellidos, String nifCif)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| apellidos | Apellidos del abonado | String | NO (\*) |
| nifCif | Nº de Identificación del abonado | String | SI (\*) |

**Comentarios:**

* Los apellidos son obligatorios en el caso de que el Abonado sea Persona Física.
* El nº de identificación tiene que estar completo, con los ceros a la izquierda si es que los lleva.

**Retorno:**

* DEU: Dirección electrónica única del abonado.

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **enviarNotificacionNotifica** |

Procedimiento que envía una notificación a la plataforma Notific@, devuelve el número de notificación y el estado de la misma.

**enviarNotificacionNotifica** *(String nmnotificacion)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| nmnotificacion | Número de notificación | String | SÍ |

**Retorno:**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="enviarNotificacion">

<xs:complexType>

<xs:attribute name="nmnotificacion" type="xs:string"/>

<xs:attribute name="cdestadonotificacion" type="xs:string"/>

<xs:attribute name="dsestadonotificacion" type="xs:string"/>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno

|  |
| --- |
| **consultaNotificacion** |

Devuelve la información de una notificación o notificaciones pertenecientes a una aplicación que coincidan con los parámetros de entrada.

*String* ***consultaNotificacion*** *(String codigoApp, String codigoOrigen, String codigoOrigenInt, String uniAdmin, String servicio, String codigoAbonado)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de aplicación | String | SI |
| codigoOrigen | Código origen | String | SI |
| codigoOrigenInt | Código origen interto | String | SI |
| uniAdmin | Unidad Administrativa | String | NO |
| servicio | Servicio | String | NO |
| codigoAbonado | Código abonado | String | SI |

**Retorno:**

* XML con la lista de notificaciones, esquema de devolución:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="numero\_notificaciones" type="xs:string" />

<xs:element name="identificador" type="xs:string" />

<xs:element name="numero\_abonado" type="xs:string" />

<xs:element name="codigo\_abonado" type="xs:string" />

<xs:element name="codigo\_origen" type="xs:string"/>

<xs:element name="codigo\_origen\_interno" type="xs:string"/>

<xs:element name="fecha\_error" type="xs:string"/>

<xs:element name="fecha\_lectura" type="xs:string"/>

<xs:element name="fecha\_puesta\_disposicion" type="xs:string"/>

<xs:element name="fecha\_rechazo" type="xs:string"/>

<xs:element name="asunto" type="xs:string"/>

<xs:element name="cuerpo" type="xs:string"/>

<xs:element name="borrado" type="xs:string"/>

<xs:element name="fecha\_borrado" type="xs:string"/>

<xs:element name="codigo\_estado" type="xs:string"/>

<xs:element name="dsestado\_postal" type="xs:string"/>

<xs:element name="numero\_aplicacion" type="xs:string"/>

<xs:element name="cifrado" type="xs:string"/>

<xs:element name="referencia\_aportacion" type="xs:string"/>

<xs:element name="cddocum" type="xs:string"/>

<xs:element name="codarchivo" type="xs:string"/>

<xs:element name="notificacion">

<xs:complexType>

<xs:sequence>

<xs:element ref="identificador"/>

<xs:element ref="codigo\_origen"/>

<xs:element ref="codigo\_origen\_interno"/>

<xs:element ref="numero\_abonado"/>

<xs:element ref="codigo\_abonado"/>

<xs:element ref="fecha\_puesta\_disposicion"/>

<xs:element ref="fecha\_lectura"/>

<xs:element ref="fecha\_rechazo"/>

<xs:element ref="fecha\_error"/>

<xs:element ref="asunto"/>

<xs:element ref="cuerpo"/>

<xs:element ref="borrado"/>

<xs:element ref="fecha\_borrado"/>

<xs:element ref="codigo\_estado"/>

<xs:element ref=" dsestado\_postal "/>

<xs:element ref="numero\_aplicacion"/>

<xs:element ref="cifrado"/>

<xs:element ref="referencia\_aportacion"/>

<xs:element ref="cddocum"/>

<xs:element ref="codarchivo"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="notificaciones">

<xs:complexType>

<xs:sequence>

<xs:element ref="notificacion" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Excepciones:** ICMException. Excepción en caso de que se produzca un error interno.

|  |
| --- |
| **enviarNotificacionGfic** |

Procedimiento que envía una notificación y devuelve el número de notificación y el estado de la misma, si incluye adjunto, no se envía como tal, se envían los parámetros que identifican al adjunto.

*String* **enviarNotificacionGfic** *(String codigoApp, String codigoOrigen, String codigoOrigenInt, String cifrada, String refAportacion, String uniAdmin, String servicio, String asunto, String cuerpo, String codigoAbonado, String [] numerosReferencia, String [] codigosArchivo, String [] nombresAdjunto, String cdProcedimiento)*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicación | String | SÍ |
| codigoOrigen | Referencia | String | SI |
| codigoOrigenInt | Referencia interna | String | SI |
|  |  |  |  |
| cifrada | Cifrado del cuerpo del mensaje: Valores posibles:  true/false o 1/0 | String | SI |
| refAportacion | Nº de Referencia de la aportación | String | NO |
| uniAdmin (\*) | Descripción Unidad Administrativa | String | SI |
| servicio | Descripción servicio | String | SI |
| asunto | Descripción del asunto | String | SI |
| cuerpo | Cuerpo de la notificación | String | NO |
| codigoAbonado | DUE del interesado | String | SI |
| numerosReferencia | Números de referencia de los adjuntos | String [] | NO |
| codigosArchivo | Códigos de archivo de los adjuntos | String [] | NO |
| nombresAdjunto | Nombre de los adjuntos | String [] | NO |
| cdProcedimiento | Código del procedimiento | String | NO |

**Comentarios:**

El nombre de cada adjunto debe contener la extensión.

Los parámetros de codigo origen son textos libres.

(\*) En el parámetro de Unidad Administriva se puede añadir el procedimiento administrativo de IPAE, concatenado el código de PA con la cadena “\_@@@\_”.

**Retorno:**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<xs:element name="enviarNotificacion">

<xs:complexType>

<xs:attribute name="nmnotificacion" type="xs:string"/>

<xs:attribute name="cdestadonotificacion" type="xs:string"/>

<xs:attribute name="dsestadonotificacion" type="xs:string"/>

</xs:complexType>

</xs:element>

</xs:schema>

**Ejemplos de los tipos de respuestas del servicio *enviarNotificacion****:*

* **1-PUESTA A DISPOSICION (OK)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion>XXXXXX</nmnotificacion>

<cdestadonotificacion>**1**</cdestadonotificacion>

<dsestadonotificacion>**PUESTA A DISPOSICION**</dsestadonotificacion>

</enviarNotificacion>

* **5-ERROR ENTREGA (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion> XXXXXX </nmnotificacion>

<cdestadonotificacion>**5**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENTREGA**</dsestadonotificacion>

</enviarNotificacion>

* **0-No ha sido aceptada la DEH del interesado (NIF), en estos momentos no es posible enviar la Notificación. (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion></nmnotificacion>

<cdestadonotificacion>**0**</cdestadonotificacion>

<dsestadonotificacion>**No ha sido aceptada la DEH del interesado XXXXXXXXX, en estos momentos no es posible enviar la Notificación.**</dsestadonotificacion>

</enviarNotificacion>

* **9-ERROR ENVIO: (Descripción del error) (ERROR)**

<enviarNotificacion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.madrid.org/enviarNotificacion.xsd">

<nmnotificacion></nmnotificacion>

<cdestadonotificacion>**9**</cdestadonotificacion>

<dsestadonotificacion>**ERROR ENVIO: Se ha producido un error …** </dsestadonotificacion>

</enviarNotificacion>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

* **10-ERROR ENVIO: (Descripción del error) (ERROR)**

<soapenv:Fault>

<faultcode>soapenv:Server.userException</faultcode>

<faultstring>java.rmi.RemoteException: 02 02 ERR002 -No existe el Procedimiento Administrativo.</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">icmaplides06.madrid.org</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

**Excepciones:** ICMException. Excepción en caso de que la aplicación no exista o se produzca un error interno.

|  |
| --- |
| **datadoOrganismo** |

Método que realiza la recepción del **Datado** de un envío*.*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| organismo\_emisor | Tipo compuesto que contendrá tanto el código DIR3 del organismo como su nombre. | Tipo  Organismo  Emisor | SI |
| codigo\_dir3 | Código DIR3 del organismo emisor. | String | SI |
| nombre | Nombre del organismo emisor. | String | SI |
| identificador\_destinatario | Identificador del envío devuelto en la respuesta de altaEnvio() | String | SI |
| resultado | Estado de la entrega del comunicado que se corresponderá con uno de los estados de Notific@. | String | SI |
| fecha | Fecha en la que se produjo el intento de entrega. | dateTime | SI |
| receptor | Datos del receptor del envío. | tipo\_persona | SI |
| nif | NIF del receptor del envío. | string | SI |
| nombre | Nombre del receptor del envío. | string | SI |
| modo | Indica quién ha originado el datado. Valores posibles: correos\_cie, electrónico, sede, carpeta. | string | SI |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| **codigo\_respuesta** | string | Código de tres dígitos asociado al resultado. |
| **texto\_respuesta** | string | Descripción de la operación. |

|  |
| --- |
| **certificacionOrganismo** |

Método que realiza la recepción de la **Certificación** de un envío*.*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | |
| organismo\_emisor | Sólo el código DIR3 del Organismo. | | String |
| identificador\_destinatario | Identificador del envío devuelto en la respuesta de altaEnvio() | | String |
| acuse\_o\_sobre | Indica si es un acuse de recibo o un sobre escaneado. | | String |
| certificacion | Contiene el base64 del PDF con el contenido. | | String |
| hash\_sha1 | Codificación base64 del sha1 del contenido binario del documento | | String |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| codigo\_respuesta | string | Código de tres dígitos asociado al resultado. |
| texto\_respuesta | string | Descripción de la operación. |

|  |
| --- |
| **sincronizarEnvio** |

Método que realiza la recepción del **Datado y Certificado** de un envío*.*

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| organismoEmisor | Código DIR3 del organismo emisor del envío. | String | SI |
| identificador | Identificador público del envío que se está tratando de sincronizar. | String | SI |
| tipoEntrega | Indica si se está enviando un datado, una certificación o ambos casos.  1- Datado  2- Datado y certificacion | Integer | SI |
| modoNotificacion | Modo en que se ha realizado la entrega.  1- Sede  2- Funcionario Habilitado  3- Postal  4- DEH  5- Carpeta | Integer | SI |
| estado | Estado en que se encuentra el envío. Incluyendo los diferentes estados postales. | String | SI |
| fechaEstado | Fecha en la que se produce el cambio de estado. | String | SI |
| receptor | Receptor de la notificación. | Receptor | SI |
| acusePDF | Documento que certifica la entrega, en formato PDF. | Acuse | NO |
| acuseXML | Documento que certifica la entrega, en el formato XML definido para la Prueba Electrónica de entrega. | Acuse | NO |
| opcionesSincronizarEnvio | Opciones relacionadas con el envío de información de la entrega. | Opciones | NO |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| codigoRespuesta | string | Código de tres dígitos asociado al resultado. |
| descripcionRespuesta | string | Descripción de la operación. |
| identificador | string | Identificador enviado en la petición |

|  |
| --- |
| **localizaNotificaciones** |

Método que realiza la recuperación de notificaciones que están pendientes de acceso, dirigidas a la CM.

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicaión. | String | SI |
| nifTitular | NIF de la CM. | String | SI |
| nifDestinatario | NIF de la CM. | String | SI |
| fechaDesde | Fecha desde la cual se realiza la búsqueda. | String | NO |
| fechaHasta | Fecha hasta la cual se realiza la búsqueda. | String | NO |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| cdRespuesta | string | Código de tres dígitos asociado al resultado. |
| dsRespuesta | string | Descripción de la operación. |
| numNotificaciones | string | Número de notificaciones recuperadas. |

|  |
| --- |
| **peticionNotificacion** |

Realiza la comparecencia de las nofificaciones de la CM y recupera todos los documentos asociados a ellas.

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| codigoApp | Código de la aplicaión. | String | SI |
| cdidentificador | Identificador del envío | String | SI |
| codigoOrigen | Código que identifica el origen del envio | String | SI |
| nifReceptor | NIF de la CM.  NIF de la persona compareciente de la notificación | String | SI |
| nombreReceptor | Nombre de la CM. Nombre de la persona compareciente de la notificación. | String | SI |
| evento | Para las notificaciones implicará la aceptación de la notificación a efectos jurídicos. | String | SI |
| concepto | Concepto de la notificación | String | SI |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| cdRespuesta | string | Código de tres dígitos asociado al resultado. |
| dsRespuesta | string | Descripción de la operación. |
| cdidentificador | string | Identificador del envío. |

|  |
| --- |
| **enviarNotificacionUnidadRegistro** |

Realiza el envío en registro del documento de una notificación a otra unidad de registro.

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| cdApp | Código de la aplicaión. | String | SI |
| usuario | Usuario. | String | SI |
| cddocum | Código de documento. | String | SI |
| unidadorig | Unidad adminitrativa de origen. | String | SI |
| unidaddest | Unidad adminitrativa de destino. | String | SI |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| cdRespuesta | string | Código de tres dígitos asociado al resultado. |
| dsRespuesta | string | Descripción de la operación. |

|  |
| --- |
| **recepcionNotificacionUnidadRegistro** |

Realiza la recepción del documento de una notificación en la unidad de registro de destino.

**Parámetros de entrada:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parámetro** | **Descripción** | **Tipo** | **Obligatorio** |
| cdApp | Código de la aplicaión. | String | SI |
| usuario | Usuario. | String | SI |
| cddocum | Código de documento. | String | SI |
| unidaddest | Unidad adminitrativa de destino. | String | SI |

**Retorno:**

|  |  |  |
| --- | --- | --- |
| **Parámetros** | **Tipo** | **Descripción** |
| cdRespuesta | string | Código de tres dígitos asociado al resultado. |
| dsRespuesta | string | Descripción de la operación. |

**ESTRUCTURAS DE DATOS**

Todos los parámetros utilizados en NOTE\_WS se pasan por medio de tipos primitivos, sin utilizar clases complejas ni agrupaciones.

**RESPONSABLE TECNICO (ante dudas técnicas en la implementación y uso del webservice)**

Unidad de Aplicaciones Comunes y Administración Electrónica

**RESPONSABLE FUNCIONAL (ante dudas funcionales en cuanto al funcionamiento del webservice o del sistema de información a acceder)**

Unidad de Aplicaciones Comunes y Administración Electrónica

**OBSERVACIONES**

## Integración de una aplicación con el WS

**Normalmente, el flujo que se seguirá será el siguiente:**

* **Envío de Notificaciones Telemáticas**

1. Antes de proceder al envío de una Notificación Administrativa al Ciudadano o Empresa administrativamente debería realizarse el apunte en el registro de salida de la CM (ver Servicio Registro – Registro de Salida).
2. Previo al envío de la Notificación, deberá comprobarse que el interesado a quién se dirigirá la notificación está registrado en el Servicio de Notificaciones Telemáticas, por lo que en primer lugar, se llama al método **existeNIF** del Sistema NOTE\_WS.

En la respuesta de este método se comprueba:

* 1. Si existe el interesado (*<existeNIF>****1****</existeNIF>…)*. Se debe comprobar además si está autorizado para recibir notificaciones.
     1. Si está autorizado (…<itAutorizado>**S**</itAutorizado>) se le puede enviar la notificación.
     2. Si no está autorizado (…<itAutorizado>**N**</itAutorizado>) aún no se le pueden enviar notificaciones, el interesado debe aceptar su DEH para poder enviarle notificaciones.
  2. Si no existe el interesado (<existeNIF>**0**</existeNIF>…) podrá realizar un ***preAltaAbonado*** del mismo.

Dependiendo del medio del prealta que se informe, el interesado estará autorizado para recibir o no notificaciones, antes de que el interesado acepte la DEH. Para saber este dato, debe llamar de nuevo al método ***existeNIF*** y comprobar si se le pueden enviar notificaciones.

* 1. Si existe el interesado, pero está incativo por unificación (<existeNIF>**4**</existeNIF>…), no podrá enviarle notificaciones.

1. A continuación, si el interesado existe y está autorizado para enviarle notificaciones, se utiliza el servicio que dispone el Sistema NOTE\_WS para el envío de notificaciones ***enviarNotificacion***
2. Respecto a la integración con la plataforma Notific@, si la aplicación está integrada, el envío de notificaciones también se enviará a dicha plataforma, la notificación también estará disponible en el PAG (Carpeta Ciudadana).

* **Envío de Notificaciones POSTALES**
* Antes de proceder al envío de una Notificación Administrativa al Ciudadano o Empresa administrativamente debería realizarse el apunte en el registro de salida de la CM (ver Servicio Registro – Registro de Salida).
* Previo al envío de la Notificación, deberá comprobarse que el interesado a quién se dirigirá la notificación está registrado en el Servicio de Notificaciones Telemáticas, por lo que en primer lugar, se llama al método **existeNIFPostal** del Sistema NOTE\_WS. Para que en caso de que no exista, se le dé de alta

En la respuesta de este método :

* + Si existe el interesado o puede darse de alta (*<existeNIF>****1****</existeNIF>…)*.
  + Si no puede darse de alta el interesado (<existeNIF>**0**</existeNIF>…)

Aunque esté preparado para enviársele notificaciones postales, esto no habilita el envío de notificaciones telemáticas, para ello, consultar el punto anterior.

* A continuación, si el interesado existe, se utiliza el servicio que dispone el Sistema NOTE\_WS para el envío de notificaciones postal ***enviarNotificacionPostal.*** Para poder dar de alta es necesario que el procedimiento esté dado de alta en la tabla NOTE\_CONTRATO, informando de DIR3, código de contrato postal y código de facturación.
* Respecto a la integración con la plataforma Notific@, si la aplicación está integrada, el envío de notificaciones también se enviará a dicha plataforma, la notificación también estará disponible en el PAG (Carpeta Ciudadana). También le aparecerá en NOTE\_PUBLIC.
* **Recepción de Acuse de Recibo**

El Sistema NOTE ofrece servicios para consultar el estado de las Notificaciones Telemáticas enviadas desde un entorno de Aplicación. Los posibles estados de una notificación son:

1. – Puesta a disposición – Notificaciones aún no recepcionadas por el interesado
2. – Leídas – Notificaciones leídas por el interesado
3. – Rechazadas – Notificaciones rechazadas por el interesado
4. – Rechazadas automáticamente – Notificaciones rechazadas automáticamente una vez vencido el plazo legal de 10 días sin recepción por parte del interesado. También son aquellas que no se pudieron entregar vía postal ( fallecido, dirección incorrecta…)

Así, las Aplicaciones deberán considerar un proceso automático de consulta periódica del estado de las Notificaciones enviadas, a partir del cual proceder al tratamiento de gestión que corresponda en cada caso.

* Se describe a continuación la secuencia de utilización de los servicios NOTE:

1. Utilización del servicio ***obtenerNotificacionesPendientesTotalAcuse*** del Sistema NOTE\_WS con el que se obtendrá información del estado de las notificaciones enviadas. Si la notificación no está en estado de puesta a disposición (esté leída, rechazada, rechazada automáticamente) tendrá informados los campos cddocum y codarchivo *(…<cddocum>03180012349</cddocum> y <codarchivo>5527237</codarchivo>)* que corresponden con el número registro y el código de archivo del acuse de recibo correspondiente. En este no se recupera el documento de acuse de recibo, se obtendrá en la siguiente llamada. En caso de que haya sido rechazada automáticamente postal , por no poder entregar la carta al destinatario, devuelve informado el campo dsestado\_postal, indicando el motivo.
2. En los casos en los que las notificaciones hayan cambiado de estado (Leídas, Rechazadas, Rechazadas automáticamente) podrá obtener el acuse de recibo (pdf en base64) correspondiente de las mismas utilizando el servicio ***obtenerNotificacion.*** En caso de que haya sido rechazada automáticamente postal , por no poder entregar la carta al destinatario, devuelve informado el campo dsestado\_postal, indicando el motivo.
3. Una vez tratada la información en la gestión propia de los entornos de las aplicaciones, se utilizará el servicio ***eliminarNotificacionesPendientes*** para evitar obtener de nuevo esta información en las sucesivas siguientes llamadas al servicio *obtenerNotificacionesPendientesTotalAcuse*.

* Si se utilizan los servicios a través de wrapper de base de datos (DBA\_WS), la secuencia de utilización de los servicios NOTE es similar a la anteriormente descrita:

1. En primer lugar, utilización del servicio ***OBTENERNOTIFICACIONESPENDACUSE*** con el que se obtendrá información del estado de las notificaciones enviadas.

En este proceso, se insertarán en la tabla de NOTE (WS\_NOTE\_NOTIFPENDIENTES) las notificaciones correspondientes. Si la notificación ha cambiado de estado (Leídas, Rechazadas, Rechazadas automáticamente) estarán informados los campos NOTE\_CDDOCUM y NOTE\_CODARCHIVO, correspondientes al acuse de recibo de la notificación. En caso de que haya sido rechazada automáticamente postal , por no poder entregar la carta al destinatario, devuelve informado el campo dsestado\_postal, indicando el motivo.

1. Una vez haya gestionada la información del servicio previo y antes de invocar al siguiente servicio, deberá realizar la eliminación de los registros de la tabla WS\_NOTE\_NOTIFPENDIENTES.
2. En los casos en los que las notificaciones hayan cambiado de estado (Leídas, Rechazadas, Rechazadas automáticamente) podrá obtener el acuse de recibo de las mismas utilizando el servicio ***OBTENERNOTIFICACION***. En caso de que haya sido rechazada automáticamente postal , por no poder entregar la carta al destinatario, devuelve informado el campo dsestado\_postal, indicando el motivo.
3. Una vez tratada la información en la gestión propia de los entornos de las aplicaciones, se utilizará el servicio ***ELIMINARNOTIFICACIONESPEND*** para evitar obtener de nuevo esta información en las sucesivas siguientes llamadas al servicio ***OBTENERNOTIFICACIONESPENDACUSE***.

**SEGURIDAD**

|  |
| --- |
|  |

Acceso público internet.

|  |
| --- |
|  |

Acceso público intranet.

|  |
| --- |
| **X** |

Securizado: Acceso solo desde aplicaciones de ICM instaladas en los entornos de Internet e Intranet.

|  |
| --- |
|  |

Acceso privado con certificado digital: Los clientes del webservice tienen que identificarse mediante certificado digital.

|  |
| --- |
|  |

Acceso privado con firma de mensaje SOAP: Los mensajes SOAP tienen que ir firmados con un certificado digital.