

HISTORIAL DE VERSIONES

Versión	Fecha	Cambios
3.2.2.5	05/02/2018	<ul style="list-style-type: none"> Inclusión del abonado “Persona Jurídica” identificado por su NIF. Inclusión de nuevas validaciones para los códigos DIR3 y RPA
3.0.0	12/12/16	<ul style="list-style-type: none"> Se incluyen los siguientes métodos en la interfaz de la API (MCSN): enviarRemesaEni(RemesaDocMetadatos remesaDocMetadatos) y enviarRemesaEni(RemesaEniDoc remesa) Se realiza la validación del documento ENI haciendo uso de la librería conversor-eni.jar Incluidas clases de remesa y notificación necesarios para implementar la lógica de la nueva operación de servicio web
2.8.15	15/02/16	<ul style="list-style-type: none"> El componente de configura de forma dinámica mediante el objeto <i>Configuration</i>.
2.4.0	20/11/14	<ul style="list-style-type: none"> Se añade una nueva operación denominada “obtenerInfAbonadoSuscrito(String identificadorAbonado, int codServicio)” que permite obtener los datos de un abonado cuyo identificador se pasa como parámetro si está dado de alta en el servicio que se pasa como parámetro.
2.3.0	07/05/12	<ul style="list-style-type: none"> Se añade una nueva operación denominada “obtenerInfCompletaRemesas(int[])” que permite obtener el registro de salida en @ries de la puesta a disposición de las notificaciones de dicha remesa. El email es obligatorio en el alta de abonados.
2.2.0	13/03/12	<ul style="list-style-type: none"> Se añade información complementaria acerca de la identificación de los abonados en la plataforma. Se actualiza la normativa vigente.
2.0.3	05/07/10	<ul style="list-style-type: none"> Se hace compatible el API de Notific@ con la versión de Axis 1.4. Este paquete permite al API generar las solicitudes SOAP y enviarlas al servicio Web de Notific@. Para aquellos integradores que dispongan de una versión anterior del API de Notific@, es importante actualizar todas las librerías del directorio lib.
2.0.2	05/10/05	<ul style="list-style-type: none"> Se hace compatible el API de Notific@ con la versión del Axis 1.2.1. Este paquete permite al API generar las solicitudes SOAP y enviarlas al servicio Web de Notific@. Para aquellos integradores que dispongan de una versión anterior del API de Notific@, es importante actualizar todas las librerías del directorio lib. Se resuelve un error en la librería itext.jar que

		<p>imposibilitaba la carga de esta, en un servidor de aplicaciones. Esta librería es la encargada de realizar la firma y cifrado de PDFs.</p>
2.0.1	21/06/05	<ul style="list-style-type: none"> • Se resuelve un error que había en el API y que imposibilitaba la conexión HTTPS contra el servidor de Notific@ desde clientes LINUX. • Se añade un nuevo método constructor al cliente <i>MCSN</i> que permite pasar como parámetro una instancia de un <i>properties</i> que contiene los parámetros de configuración del fichero (<i>mcs.properties</i>), en vez de pasarle la ruta completa del archivo. Para más información ver el Javadoc
2.0	25/05/05	<ul style="list-style-type: none"> • Se añade una nueva funcionalidad que permite la creación y el envío de notificaciones cifradas con el certificado X509 del destinatario y firmadas con el certificado de componente asociado a la Entidad Emisora, todo ello embebido dentro de uno o varios PDFs. <ul style="list-style-type: none"> ○ Aparece una nueva clase denominada CriptoAbonado que extiende de Abonado y que permite indicar el certificado del usuario. <i>notificaciones.cliente.estructuras.CriptoAbonado</i> ○ Nueva clase que representa a una notificación cifrada <i>notificaciones.cliente.estructuras.NotificacionCifrada</i>. ○ Se añade un ejemplo de envío de notificación cifrada (<i>EnviarNotificacionCifrada.java</i>). • Nueva funcionalidad para la obtención de certificados digitales de usuario con el fin de cifrar notificaciones. Este certificado se obtiene cuando el usuario se autentica en el acceso al sistema de notificaciones. <i>CertificadoInf[] solicitarCertificadoAbonados(String[] anagramasAbons)</i>. • Cambio en el archivo de configuración del API (<i>mcsn.properties</i>). Se añade un nuevo método de obtención de la contraseña que protege a la clave privada del certificado de componte que firma las solicitudes. Anteriormente se indicaba en el fichero de configuración directamente la contraseña, ahora se permite además indicar, si se desea, una clase a partir de la cual se obtiene de una forma segura dicha contraseña. • Se ha puesto en <i>deprecated</i> la clase que permite obtener el Anagrama fiscal Largo (GeneradorIDUsuario del paquete <i>notificaciones.common.util</i>). A partir de ahora se deberá obtener del servidor de autenticación de @Firma. • Al dar de alta un usuario en un servicio con los atributos teléfono móvil y/o correo electrónico rellenos, el sistema al recibir la solicitud entenderá que este usuario desea recibir avisos SMS y/o correo electrónico.

		Para más información consultar el documento API de Entidades Emisoras v2.0.
1.4	28/01/05	<ul style="list-style-type: none"> • Se añade una nueva funcionalidad. Solicitud de comprobación del estado de un conjunto de usuarios en un servicio. Ver Javadoc de la clase MCSN • Se amplia documentación Javadoc acerca de los posibles estados de una Notificación. • Se han puesto <i>deprecated</i> algunas constantes de la clase <i>notificaciones.common.util.EstadoEstadoNotificacion</i>, en particular: <ul style="list-style-type: none"> ○ PD_Y_SIN_NOTIFICAR: Notificación puesta a disposición e imposible envío de aviso al Abonado, por correo y/o SMS ○ PD_Y_NOTIFICADO: Notificación puesta a disposición y avisado al Abonado por SMS ○ PD_Y_ENVIADA_NOTIFICACION: Notificación puesta a disposición y enviado aviso por Correo al abonado. ○ PD_E_IMPOSIBLE_NOTIFICAR: Notificación puesta a disposición e imposible de enviar aviso al Abonado por Correo y/o SMS <p>Para más información consultar el documento API de Entidades Emisoras v1.4.</p>