



MANUAL DE USO DEL SERVICIO DE MENSAJERÍA

Métodos Web que permiten al usuario de mensajería de valenciaportpcs.net, realizar procedimientos como la conexión y validación, así como el procedimiento de intercambio de mensajes

ÍNDICE

1 // INTRODUCCIÓN	3
1.1 // CONTROL DE CAMBIOS	3
1.2 // OBJETO.....	3
1.3 // ALCANCE	3
1.4 // CONTENIDOS	4
1.5 // ABREVIATURAS Y ACRÓNIMOS	4
2 // APIS MENSAJERÍA VALENCIAPORTPCS	5
3 // API REST.....	6
4 // SERVICIO WEB SOAP	7
4.1 // SOBRE EL TRATAMIENTO DE ERRORES	7
4.2 // DEFINICIONES	7
4.3 // DESCRIPCIÓN DEL SERVICIO WEB SOAP DE LOGIN	7
4.4 // MÉTODOS DEL SERVICIO WEB SOAP DE LOGIN	7
4.4.1. Parámetros.....	7
4.4.2. Valor de retorno	7
4.4.3. Excepciones.....	8
4.5 // DESCRIPCIÓN DEL SERVICIO WEB SOAP DE MENSAJERÍA	8
4.6 // MÉTODOS DEL SERVICIO WEB SOAP DE MENSAJERÍA	10
4.6.1. Método Upload.....	10
4.6.2. Método UploadZippedMessage	11
4.6.3. Método UploadZippedFile	11
4.6.4. Método ListMessages	11
4.6.5. Método ListMessagesByService.....	12
4.6.6. Método ListMessagesByMessageType	13
4.6.7. Método ListMessagesByDate	14
4.6.8. Método Download	15
4.6.9. Método DownloadZippedMessage	15
4.6.10. Método DownloadZippedFile	16

1 // Introducción

1.1 // Control de Cambios

La siguiente tabla recoge los capítulos sobre los que se han realizado modificaciones respecto a la anterior versión del documento.

Los cambios concretos realizados, quedan indicados a lo largo del documento en color rojo.

Versión	Partes que Cambian	Descripción del Cambio
09 Junio 2011	--	Version Inicial
10 Junio 2011	Adición servicio Ferrocarril	Adición servicio Ferrocarril
29 Marzo 2012	4.1 // Descripción del servicio 4.2.4 // Método ListMessages 4.2.5 // Método ListMessagesByService 4.2.6 // Método ListMessagesByMessageType 4.2.7 // Método ListMessagesByDate	Nuevo valor posible "All" para el parámetro "Pending" El parámetro "Pending" se ignora Nuevo método ListMessagesByDate
10 Noviembre 2015	4.1 // Descripción del servicio	Eliminación COARRIs RAILW y adición mensajes servicio Escalas
14 Julio 2022	Adición mensajería API REST	Adición mensajería API REST

1.2 // Objeto

El objeto del presente documento es mostrar a los clientes del portal valenciaportpcs.net el modo de utilizar el API de mensajería, ya sea éste SOAP o REST, y que permite el intercambio (envío y recepción) de información con el portal.

1.3 // Alcance

El documento está orientado al cliente final de mensajería de valenciaportpcs.net, facilitándole la integración con una serie de métodos Web que le permitan realizar el procedimiento de conexión y validación de usuario en el portal, y el posterior procedimiento de intercambio de mensajes.

1.4 // Contenidos

El presente documento está estructurado en diferentes capítulos según los siguientes puntos:

- Capítulo 1 Introducción
- Capítulo 2 APIs de mensajería disponibles en Valenciaportpcs.net
- Capítulo 3 Funcionamiento API REST
- Capítulo 4 Funcionamiento Servicio Web SOAP

1.5 // Abreviaturas y Acrónimos

Término	Significado
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
API	Application Programming Interface

2 // APIs Mensajería ValenciaportPCS

Para el intercambio de mensajes dentro del portal valenciaportpcs.net existe la posibilidad de usar servicios web bajo protocolo SOAP o bien APIs REST, ofreciendo ambos una funcionalidad similar.

Para el envío de mensajería **mediante API REST**, detallada en el punto 3 de esta misma guía, se hará uso de OAuth para la autenticación y del propio API de mensajería para el intercambio de mensajes.

Para la comunicación bajo **protocolo SOAP**, detallada en el punto 4 de esta misma guía, se hará uso, de cara a la autenticación, de un web service específico de Login y de un segundo web service, de mensajería, para el envío de los intercambios.

Se recomienda el uso de API REST para desarrollos nuevos, al ofrecer un mayor abanico de tipos de mensajes con los que se puede trabajar (en concreto mensajes de los servicios de Declaraciones Sumarias, Escalas y Mercancías Peligrosas, no disponibles con los servicios web SOAP).

3 // API REST

Para el uso de la funcionalidad expuesta por el API de mensajería es necesario disponer previamente de un token de autenticación. Éste se obtendrá mediante llamada a un servicio de OAuth.

Todos los detalles sobre la obtención de este token, así como de la funcionalidad ofrecida por el propio API de mensajería, se encuentran en:

Documentación de la API REST de mensajería en Producción:

<https://api.valenciaportpcs.net/messaging/swagger>

Documentación de la API REST de mensajería en Test:

<https://testapi.valenciaportpcs.net/messaging/swagger>

El API REST de mensajería ofrece la posibilidad de trabajar con dos buzones, el buzón por defecto "default" (parámetro box en todos los métodos expuestos por el API), para los siguientes servicios:

1. Transporte Terrestre
2. Reservas de Carga
3. Instrucciones de Embarque
4. Listas de Carga/Descarga
5. Listas de Pasajeros y Declaración de Residuos de la Escala
6. Servicio de transmisión del Peso Verificado
7. Track&Trace

Y, en el caso de intercambios de los servicios de Declaraciones Sumarias, Escalas y Mercancías Peligrosas, el parámetro box (buzón) APVINCOMING para el envío hacia ValenciaportPCS y APVOUTGOING para recibir desde ValenciaportPCS las respuestas.

Los mensajes de estos tres últimos servicios de ValenciaportPCS sólo podrán ser transmitidos a través de API REST, NO estando disponibles bajo los servicios web SOAP que veremos a continuación.

4 // Servicio Web SOAP

4.1 // Sobre el tratamiento de errores

El portal valenciaportpcs.net se ha desarrollado con tecnología .NET, empleando el protocolo SOAP (Simple Object Access Protocol) para sus servicios Web (Web Services). En este entorno, los errores se gestionan mediante excepciones. Cualquier error producido en el sistema, genera una excepción SOAP que contiene toda la información sobre él. Todos los métodos detallados en este documento, siempre que se genere un error o excepción no controlada, devolverán una excepción SOAP, que deberá ser tratada por la aplicación que los utilice. En ciertos casos también se devuelven como excepciones errores controlados, en cuyo caso se describen en la sección dedicada a cada método particular.

4.2 // Definiciones

- **Sesión:** Conexión abierta por un usuario en el portal.
- **GUID:** Identificador Único Global de un mensaje en el sistema.

4.3 // Descripción del servicio web SOAP de Login

El servicio Web de acceso (Login) permite a la aplicación cliente abrir y cerrar sesiones contra el servidor, obteniendo un ticket de sesión necesario en la invocación de los diferentes métodos del servicio Web de mensajería. Para poder realizar estas acciones, valenciaportpcs.net pone a disposición de sus usuarios los métodos mostrados en el siguiente punto.

Se accede a este servicio en Producción mediante la URL:

<https://www.valenciaportpcs.net/services/login.asmx>

Se accede a este servicio en entorno de Test mediante la URL:

<https://test.valenciaportpcs.net/services/login.asmx>

4.4 // Métodos del servicio web SOAP de Login

Es el método que permite abrir una sesión de un usuario en el portal. La sintaxis del método es la siguiente:

```
Login (UserLogin As String, Password As String,  
OrganizationCode As String) As UserSession
```

4.4.1. Parámetros

- **UserLogin:** Nombre del usuario registrado en valenciaportpcs.net.
- **Password:** Clave (contraseña) del usuario indicado en UserLogin.
- **OrganizationCode:** Código de la organización a la que pertenece el usuario indicado en UserLogin.

4.4.2. Valor de retorno

- **UserSession:** Información sobre la sesión del usuario que ha accedido con éxito al portal.

El valor de retorno `UserSession` es una estructura, encapsulada en un XML, que devuelve toda la información sobre el usuario y su organización registrada en `valenciaportpcs.net`. Se puede acceder al contenido completo de dicha estructura abriendo el servicio Web de Login en un navegador; no obstante, el dato más relevante de dicha estructura es el elemento `TicketGUID`, que identifica unívocamente la sesión que el portal ha abierto para este usuario, y que es necesario para utilizar el resto de métodos de mensajería descritos a continuación.

4.4.3. Excepciones

Si usuario y/o contraseña no son válidos, este método Web devuelve una excepción informando de que el acceso no ha podido realizarse.

4.5 // Descripción del servicio web SOAP de mensajería

El servicio Web de Mensajería permite a la aplicación cliente consultar, descargar y enviar mensajes al portal `valenciaportpcs.net`. Para poder realizar estas acciones, el portal pone a disposición de sus usuarios los métodos mostrados en el siguiente punto.

Algunos de los parámetros que hacen referencia a códigos que se pasan a los métodos de este servicio son "Listas de Constantes". En esos casos, será necesario seleccionar uno de los valores proporcionados por dichas listas, o el método Web devolverá un error.

Las "Listas de Constantes" utilizadas, son las siguientes:

TransportService.ServiceCodes: Lista de valores posibles para el parámetro `ServiceCode`.

- **SHIPP**: Servicio de Instrucciones de Embarque (Shipping Instructions).
- **TRACK**: Servicio de Seguimiento (Track & Trace).
- **TRANS**: Servicio de Transporte Terrestre
- **BKING**: Servicio de Reservas de Carga (Booking)
- **STEIN**: Servicio de Instrucciones a Terminales
- **RAILW**: Servicio de Ferrocarril
- **BERTH**: Servicio de Escalas.

TransportService.MessageTypes: Lista de valores posibles para el parámetro `MessageType`.

Mensajes del Servicio de Instrucciones de Embarque:

- **IFTMIN**: Instrucción de Embarque (Shipping Instruction).
- **CONTRL**: Confirmación de recepción del mensaje por parte del sistema receptor.
- **APERAK**: Respuesta de aceptación o rechazo del contenido del mensaje.

Mensajes del Servicio de Seguimiento:

- **IFTSTA**: Mensaje de Seguimiento (Track & Trace)

Mensajes del Servicio de Ferrocarril:

- **COPRAR**: Lista de contenedores a cargar/descargar por la terminal

- **CONTRL:** Confirmación de recepción del mensaje por parte del sistema receptor.

Mensajes del Servicio de Transporte Terrestre (formato valenciaportpcs.net):

- **DUTv2:** Documento Único de Transporte
- **ReleaseOrderv2:** Orden de Entrega
- **AcceptanceOrderv2:** Orden de Admisión
- **InlandTransportDetailsv2:** Asignación de los Datos de Transporte
- **ReleaseConfirmationv2:** Confirmación de Entrega
- **AcceptanceConfirmationv2:** Confirmación de Admisión
- **Acknowledgementv2:** Confirmación de recepción (respuesta de aceptación o rechazo)

Mensajes del Servicio de Reservas de Carga:

- **IFTMBF:** Reserva de carga (Booking).
- **CONTRL:** Confirmación de recepción del mensaje por parte del sistema receptor.
- **IFTMBC:** Confirmación de reserva de carga (Booking Confirmation).

Mensajes del Servicio de Instrucciones a Terminales (formato valenciaportpcs.net):

- **COPLIS:** Lista de contenedores del consignatario de mercancía
- **COPORD:** Lista de carga/descarga del consignatario de buque
- **COARRI:** Confirmación de carga/descarga
- **CONTRL:** Confirmación de recepción del mensaje por parte del sistema receptor.
- **APERAK:** Respuesta de aceptación o rechazo del contenido del mensaje.

Mensajes del Servicio de Instrucciones a Terminales (formato antiguo SIC):

- **CONTAINERLIST:** Lista de carga/descarga del consignatario de mercancía
- **CONTAINERLISTORDER:** Lista de contenedores del consignatario de buque
- **CONTAINERLISTREPORT:** Confirmación de carga/descarga
- **RESPONSEMESSAGE:** Confirmación de recepción del mensaje

Mensajes del Servicio de Escalas:

- **PAXLST:** Lista de pasajeros y Lista de Tripulantes.
- **WASDIS:** Notificación de Residuos.

TransportService.FileTypes: Lista de valores posibles para el parámetro FileType.

- **XML:** Formato XML
- **X12:** Formato ANSI X12
- **FP:** Formato fichero plano
- **EDI:** Formato EDIFACT

TransportService.MessageDirection: Lista de valores posibles para el parámetro [In].

- **Out:** Envío de mensajes al portal desde el usuario.
- **In:** Recepción de mensajes del portal por el usuario.

TransportService.MessageStates: Lista de valores posibles para el parámetro Pending.

- **Pending:** Mensajes que se encuentran pendientes de que el usuario realice su descarga.
- **NoPending:** Mensajes que se encuentran en el portal pero que ya han sido descargados por el usuario.
- **All:** No filtrar por estado. Listar tanto mensajes que se encuentran pendientes de descargar como mensajes que ya han sido descargados por el usuario.

Se accede a este servicio en Producción mediante la URL:

<https://www.valenciaportpcs.net/services/transportservice.asmx>

Se accede a este servicio en entorno de Test mediante la URL:

<https://test.valenciaportpcs.net/services/transportservice.asmx>

Los mensajes de los servicios de Declaraciones Sumarias, Escalas y Mercancías peligrosas sólo podrán ser transmitidos a través de API REST, NO estando disponibles bajo los servicios web SOAP.

4.6 // Métodos del servicio web SOAP de mensajería

A continuación se enumeran y describen los diferentes métodos Web que oferta el Servicio de Mensajería de valenciaportpcs.net. En todos ellos es necesario trasladar un ticket de sesión para poder utilizarlos, de forma que el paso previo a invocar cualquiera de estos métodos es haber realizado una llamada exitosa al método Login descrito en el capítulo anterior.

4.6.1. Método Upload

Este método permite enviar un mensaje al portal. La sintaxis del método es la siguiente:

Upload (ByteArray () As Byte, SessionTicket As String) As Boolean

4.6.1.1. Parámetros

- **ByteArray:** Cadena de bytes con el contenido del mensaje.
- **SessionTicket:** Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método Login).

4.6.1.2. Valor de retorno

- **Boolean:** Si el envío se ha realizado correctamente se devolverá true, en caso contrario false.

4.6.1.3. Excepciones

Si el mensaje no se ha podido enviar al portal, por ejemplo por no ser de un tipo identificado, este método Web devuelve una excepción informando de que el envío no ha podido realizarse.

4.6.2. Método UploadZippedMessage

Este método permite enviar un mensaje comprimido (mediante el algoritmo de compresión ZIP) al portal. Nótese que este método **no** espera un fichero con contenido, sino directamente una cadena de bytes comprimida, de forma que el resultado de la descompresión sea el mensaje en sí. La sintaxis del método es la siguiente:

```
UploadZippedMessage (ByteArray() As Byte, SessionTicket As String)
As Boolean
```

4.6.2.1. Parámetros

- **ByteArray**: Cadena de bytes con el contenido comprimido del mensaje.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.2.2. Valor de retorno

- **Boolean**: Si el envío se ha realizado correctamente se devolverá true, en caso contrario false.

4.6.2.3. Excepciones

Si el mensaje no se ha podido enviar al portal, por ejemplo por no ser de un tipo identificado, este método Web devuelve una excepción informando de que el envío no ha podido realizarse.

4.6.3. Método UploadZippedFile

Este método permite enviar al portal un fichero comprimido (mediante el algoritmo de compresión ZIP) con un único mensaje en su interior. Nótese que este método espera un fichero comprimido, cuyo contenido sea un único mensaje comprimido. La sintaxis del método es la siguiente:

```
UploadZippedFile (ByteArray() As Byte, SessionTicket As String)
As Boolean
```

4.6.3.1. Parámetros

- **ByteArray**: Cadena de bytes con el contenido comprimido del mensaje.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.3.2. Valor de retorno

- **Boolean**: Si el envío se ha realizado correctamente se devolverá true, en caso contrario false.

4.6.3.3. Excepciones

Si el mensaje no se ha podido enviar al portal, por ejemplo por no ser de un tipo identificado, este método Web devuelve una excepción informando de que el envío no ha podido realizarse.

4.6.4. Método ListMessages

Este método devuelve una estructura con todos los mensajes de la organización pendientes de descargar filtrando únicamente por su dirección (si es de envío o de recepción). La sintaxis del método es la siguiente:

ListMessages (Pending As MessageStates, In As MessageDirection, SessionTicket As String) As MessageData

4.6.4.1. Parámetros

- **Pending:** Este parámetro debe ser siempre el valor "Pending", cualquier otro valor posible se ignora. La búsqueda incluirá sólo los mensajes pendientes de descargar por el usuario.
- **In:** Indica si la búsqueda debe incluir los mensajes entrantes o salientes. Nota: De momento únicamente se devuelven los mensajes salientes, es decir, cuyo destinatario es la organización conectada.
- **SessionTicket:** Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.4.2. Valor de retorno

- **MessageData:** Lista de mensajes que cumplen el criterio de búsqueda indicado en In. Esta lista podría estar vacía.

Cada registro de la estructura MessageData contiene la siguiente información sobre cada mensaje:

- **MessageGUID** – Identificador único del mensaje
- **MessageName** – Nombre del mensaje con el siguiente formato: Servicio-Tipo_de_Mensaje-Emisor-Receptor-Número_de_mensaje-FechaHora
- **ServiceCode** – Código del servicio al que pertenece el mensaje
- **MessageType** – Tipo de mensaje
- **FileType** – Formato del mensaje (plano, xml, edi, etc.)
- **MessageNumber** – Número de mensaje
- **MessageRecipient** – Destinatario del mensaje
- **MessageGenerationDate** – Fecha en la que se generó el mensaje
- **DocumentNumber** – Número de documento asociado al mensaje
- **SenderReference** – Referencia del emisor para el documento asociado al mensaje

4.6.5. Método ListMessagesByService

Este método devuelve una estructura con todos los mensajes de la organización pendientes de descargar filtrando por el código de servicio, y su dirección (si es de envío o de recepción). La sintaxis del método es la siguiente:

ListMessagesByService (ServiceCode As ServiceCodes, Pending As MessageStates, In As MessageDirection, SessionTicket As String) As MessageData

4.6.5.1. Parámetros

- **ServiceCode:** Código de servicio de los mensajes a descargar.
- **Pending:** Este parámetro debe ser siempre el valor "Pending", cualquier otro valor posible se ignora. La búsqueda incluirá sólo los mensajes pendientes de descargar por el usuario.
- **In:** Indica si la búsqueda debe incluir los mensajes entrantes o salientes. Nota: De momento únicamente se devuelven los mensajes salientes, es decir, cuyo destinatario es la organización conectada.
- **SessionTicket:** Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.5.2. Valor de retorno

- **MessageData:** Lista de mensajes que cumplen los criterios de búsqueda indicados en ServiceCode e In. Esta lista podría estar vacía.

Cada registro de la estructura MessageData contiene la siguiente información sobre cada mensaje:

- **MessageGUID** – Identificador único del mensaje
- **MessageName** – Nombre del mensaje con el siguiente formato: Servicio-Tipo_de_Mensaje-Emisor-Receptor-Número_de_mensaje-FechaHora
- **ServiceCode** – Código del servicio al que pertenece el mensaje
- **MessageType** – Tipo de mensaje
- **FileType** – Formato del mensaje (plano, xml, edi, etc.)
- **MessageNumber** – Número de mensaje
- **MessageRecipient** – Destinatario del mensaje
- **MessageGenerationDate** – Fecha en la que se generó el mensaje
- **DocumentNumber** – Número de documento asociado al mensaje
- **SenderReference** – Referencia del emisor para el documento asociado al mensaje

4.6.6. Método ListMessagesByMessageType

Este método devuelve una estructura con todos los mensajes de la organización pendientes de descargar filtrando por el tipo de mensaje, el código de servicio y su dirección (si es de envío o de recepción). La sintaxis del método es la siguiente:

```
ListMessagesByMessageType (MessageType As MessageTypes,
ServiceCode As ServiceCodes, Pending As MessageStates, In As
MessageDirection, SessionTicket As String) As MessageData
```

4.6.6.1. Parámetros

- **MessageType:** Código del tipo de mensajes a descargar.
- **ServiceCode:** Código de servicio de los mensajes a descargar.
- **Pending:** Este parámetro debe ser siempre el valor "Pending", cualquier otro valor posible se ignora. La búsqueda incluirá sólo los mensajes pendientes de descargar por el usuario.
- **In:** Indica si la búsqueda debe incluir los mensajes entrantes o salientes. Nota: De momento únicamente se devuelven los mensajes salientes, es decir, cuyo destinatario es la organización conectada.
- **SessionTicket:** Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.6.2. Valor de retorno

- **MessageData:** Lista de mensajes que cumplen los criterios de búsqueda indicados en MessageType, ServiceCode e In. Esta lista podría estar vacía.

Cada registro de la estructura MessageData contiene la siguiente información sobre cada mensaje:

- **MessageGUID** – Identificador único del mensaje
- **MessageName** – Nombre del mensaje con el siguiente formato: Servicio-Tipo_de_Mensaje-Emisor-Receptor-Número_de_mensaje-FechaHora
- **ServiceCode** – Código del servicio al que pertenece el mensaje

- **MessageType** – Tipo de mensaje
- **FileType** – Formato del mensaje (plano, xml, edi, etc.)
- **MessageNumber** – Número de mensaje
- **MessageRecipient** – Destinatario del mensaje
- **MessageGenerationDate** – Fecha en la que se generó el mensaje
- **DocumentNumber** – Número de documento asociado al mensaje
- **SenderReference** – Referencia del emisor para el documento asociado al mensaje

4.6.7. Método ListMessagesByDate

Este método devuelve una estructura con los mensajes de la organización filtrando por la fecha del mensaje, el tipo de mensaje, el código de servicio, el estado del mensaje (si está pendiente de descarga o no) y su dirección (si es de envío o de recepción). La sintaxis del método es la siguiente:

```
ListMessagesByDate(FromDate As String, ToDate As String,
MessageType As String, ServiceCode As String, Pending As String,
In As String, SessionTicket As String) As String
```

4.6.7.1. Parámetros

- **FromDate**: Fecha más temprana de generación del mensaje (inclusive). Para evitar ambigüedades debe usarse el formato "AAAA-MM-DD" para indicar solo fecha o "AAAA-MM-DDThh:mm:ss" donde "T" es un carácter fijo separador de los campos de fecha y hora. Si la cadena está vacía o es nula no se filtra por fecha inicial.
- **ToDate**: Fecha más tardía de generación del mensaje (exclusive). Para evitar ambigüedades debe usarse el formato "AAAA-MM-DD" para indicar solo fecha o "AAAA-MM-DDThh:mm:ss" donde "T" es un carácter fijo separador de los campos de fecha y hora. Si la cadena está vacía o es nula no se filtra por fecha final.
- **MessageType**: Código del tipo de mensajes a descargar. Si la cadena está vacía o es nula no se filtra por tipo de mensaje.
- **ServiceCode**: Código de servicio de los mensajes a descargar. Si la cadena está vacía o es nula no se filtra por servicio.
- **Pending**: Indica si la búsqueda debe incluir los mensajes pendientes, los no pendientes o todos. Es obligatorio indicar uno de los valores posibles.
- **In**: Indica si la búsqueda debe incluir los mensajes entrantes o salientes. Es obligatorio indicar uno de los valores posibles. Nota: De momento únicamente se devuelven los mensajes salientes, es decir, cuyo destinatario es la organización conectada.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.7.2. Valor de retorno

- **String**: XML con la lista de mensajes que cumplen los criterios de búsqueda indicados en FromDate, ToDate, MessageType, ServiceCode, Pending e In. Esta lista podría estar vacía. El número de registros está limitado (en principio a los 1000 primeros mensajes generados que cumplen los criterios), para listar más mensajes volver a llamar al método con una fecha u hora posterior en el parámetro FromDate.

El nodo raíz del XML devuelto será “MessageData” y contendrá un elemento “Messages” por cada mensaje encontrado. Cada elemento “Messages” contendrá los siguientes elementos con información sobre el mensaje:

- **MessageGUID** – Identificador único del mensaje
- **MessageName** – Nombre del mensaje con el siguiente formato: Servicio-Tipo_de_Mensaje-Emisor-Receptor-Número_de_mensaje-FechaHora
- **ServiceCode** – Código del servicio al que pertenece el mensaje
- **MessageType** – Tipo de mensaje
- **FileType** – Formato del mensaje (plano, xml, edi, etc.)
- **MessageNumber** – Número de mensaje
- **MessageRecipient** – Destinatario del mensaje
- **MessageGenerationDate** – Fecha en la que se generó el mensaje
- **DocumentNumber** – Número de documento asociado al mensaje
- **SenderReference** – Referencia del emisor para el documento asociado al mensaje

4.6.8. Método Download

Este método permite descargar un mensaje pendiente del portal. La sintaxis del método es la siguiente:

```
Download (MessageGUID As String, SessionTicket As String) As
ByteArray ()
```

4.6.8.1. Parámetros

- **MessageGUID**: Identificador único del mensaje, obtenido a partir de cualquiera de los métodos de listado descritos anteriormente.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.8.2. Valor de retorno

- **ByteArray**: Cadena de bytes con el contenido del mensaje.

4.6.8.3. Excepciones

Si el mensaje no se ha podido descargar del portal, por ejemplo por no encontrarse a partir del identificador indicado, este método Web devuelve una excepción informando que la descarga no ha podido realizarse.

4.6.9. Método DownloadZippedMessage

Este método permite descargar en formato comprimido un mensaje pendiente del portal. La sintaxis del método es la siguiente:

```
DownloadZippedMessage (MessageGUID As String, SessionTicket As
String) As ByteArray ()
```

4.6.9.1. Parámetros

- **MessageGUID**: Identificador único del mensaje, obtenido a partir de cualquiera de los métodos de listado descritos anteriormente.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.9.2. Valor de retorno

- **ByteArray**: Cadena de bytes con el contenido comprimido del mensaje.

4.6.9.3. Excepciones

Si el mensaje no se ha podido descargar del portal, por ejemplo por no encontrarse a partir del identificador indicado, este método Web devuelve una excepción informando que la descarga no ha podido realizarse.

4.6.10. Método DownloadZippedFile

Este método permite descargar en formato comprimido un fichero que contendrá un único mensaje pendiente del portal. La sintaxis del método es la siguiente:

```
DownloadZippedFile (MessageGUID As String, SessionTicket As String) As ByteArray()
```

4.6.10.1. Parámetros

- **MessageGUID**: Identificador único del mensaje, obtenido a partir de cualquiera de los métodos de listado descritos anteriormente.
- **SessionTicket**: Código que identifica de forma única la sesión que ha iniciado el usuario en el portal (obtenido de los datos que devuelve el método **Login**).

4.6.10.2. Valor de retorno

- **ByteArray**: Cadena de bytes con el contenido del fichero comprimido, en cuyo interior estará el mensaje.

4.6.10.3. Excepciones

Si el mensaje no se ha podido descargar del portal, por ejemplo por no encontrarse a partir del identificador indicado, este método Web devuelve una excepción informando que la descarga no ha podido realizarse.



valenciaport  **pcs.net**
P o r t C o m m u n i t y S y s t e m

Centro de Atención a Usuarios
Avenida Muelle del Turia, s/n
46024 Valencia
Tel.: 902 88 44 24
R.C.I.: 10 001
www.valenciaportpcs.net
cau@valenciaportpcs.net