LATINIA

Documento LIMSP© Formación

Formación de desarrollo para Plataformas AE y SDP

Uso de trama JSON unificada para AE y SDP

Índice

1.1 Introducción a plataformas LIMSP© SDP, AE e INF

1.1.1 Plataforma LIMSP-SDP

La Plataforma LIMSP-SDP es una sistema SW de infraestructura multicanal que permite que un mensaje genérico enviado a dicha plataforma pueda ser entregado a su destinatario por diferentes canales (SMS, EMAIL, Push Notifications, Twiter, …)

La plataforma provee también de múltiples herramientas de gestión, estadísticas, etc..., así como de un sistema de plantillas que es el que permite que dicho mensaje pueda ser adaptado a los diferentes canales.

1.1.2 Plataforma LIMSP-AE

La Plataforma LIMSP-AE es una sistema SW de infraestructura que se conecta al BUS de eventos (operaciones bancarias de la entidad) y decide para todas las operaciones que le llegan cuales han de notificarse y cuales no (y por tanto deben descartarse).

La plataforma es un analizador de eventos en tiempo real que en base a unas reglas predefinidas decide que es lo que debe ser notificado. Típicamente la mayor parte de los eventos son descartados. Los eventos que deben ser notificados son enviados al SDP para que este gestione su entrega al destinatario final.

1.1.3 Sistema informacional LIMSP-INF

Se llama sistema informacional (LIMSP-INF) a un sistema SW de infraestructura que provee la información de los usuarios (clientes finales de la entidad) cargada sobre el propio sistema, para las plataformas AE y SDP.

A este sistema informacional se le cargan los datos de usuario que las plataformas necesitan para su uso, como puedan ser puntos de contacto (direcciones gsm, email, etc…), datos útiles para comunicar los mensajes (nombre, genero –para referirnos como “Sr.” o “Sra.” –), datos para el análisis de los eventos (¿es un usuario VIP?, ¿A qué segmento pertenece?, ¿Qué lista de productos del banco tiene contratados?, o bien los datos que la misma persona a introduciendo desde la banca online para configurar su servicio (Quiero que me avisen solo si el monto de la operación es mayor de 300)…

Otro tipo de datos que se cargan en el sistema informacional son los referentes a las Apps para comunicaciones de Push Notifications (PNS), como por ejemplo que Apps del banco tiene instaladas, para que sistemas (Android, Apple…), y cuáles son los tokens de comunicación para las Push con los proveedores.

Todos esos datos pueden ser utilizados tanto por AE como por SDP para hacer su trabajo.

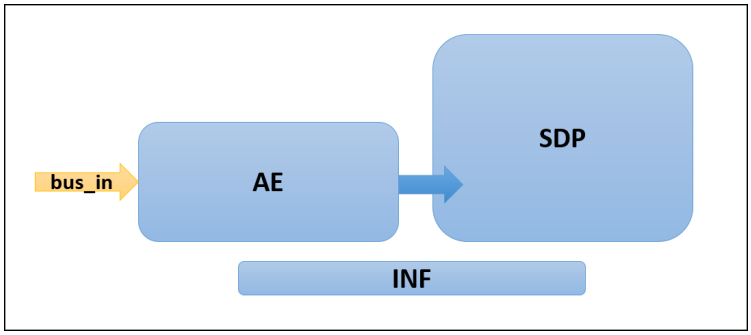
1.2 Entrada unificada de eventos y mensajes

1.2.1 Comprensión de la entrada unificada

El sistema de plataformas AE y SDP permite una única entrada unificada de eventos y mensajes.

- Llamamos **evento** a una operación bancaria concreta que entra en AE y que hay que analizar para identificar si debe ser o no comunicada.
- Llamamos **alerta** a una operación bancaria que entró como un evento de AE y que este tomo la decisión de comunicarla.
- Llamamos **notificación** a una comunicación concreta que se envía sobre SDP o AE y que se debe cursar su salida hacia el proveedor de mensajería y destinatario correspondiente.
- Llamamos **mensaje** a cualquier trama enviada sobre AE o SDP que viaja por las colas de la plataforma. El concepto mensaje es de un carácter más técnico (ej: mensaje JSON, XML sobre una cola JMS).

La entrada unificada se puede ver simplificada con un esquema como el siguiente:



En el esquema se puede ver como hay una única entrada al sistema que hemos llamado "**bus_in**", de donde partirá todos los eventos y mensajes, ya sean dirigidos hacia AE o hacia SDP.

Esta entrada unificada tiene también un modelo unificado de trama, que consiste en un formato JSON. El formato de trama JSON es lo suficientemente flexible como para poder soportar más o menos campos o bloques de campos en función de las necesidades de análisis o comunicación.

1.2.2 Provisión para la entrada unificada

Ante todo, cuando se está diseñando un servicio para enviar notificaciones al cliente, es necesario plantearse alguna pregunta:

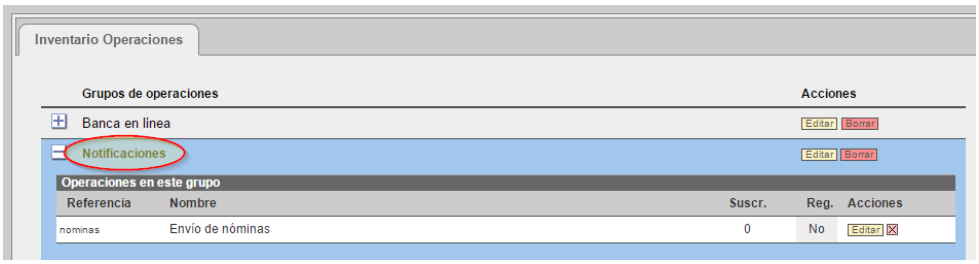
- ¿Siempre que se produzca una operación para ese servicio hay que comunicarla?
- ¿O bien la comunicación depende de si se cumplen ciertas reglas?

Poniendo ejemplos para una respuesta afirmativa a la primera pregunta, podrían ser servicios de notificación de nóminas, alta de tarjetas, etc… Cualquiera en la que generemos una operación y esa deba comunicarse si o si a su destinatario.

Como ejemplos para una respuesta afirmativa a la segunda pregunta, podrían ser servicios de suscripción a notificaciones de alertas de uso de la tarjeta de crédito, retiro de efectivo, etc… Cualquiera en la que generamos una operación y esta debe ser analizada por el sistema para decidir si debe comunicarse o no.

En cualquier caso, para poder hacer uso de la entrada unificada de eventos y mensajes para AE y SDP es necesario realizar acciones de provisión adicionales:

- Crear los grupos de operaciones bancarias adecuados en la herramienta de ***Catálogo de operaciones***.
- Dentro de cada grupo crear las diferentes operaciones, estableciendo un código de operación/servicio y una descripción para cada uno de ellos.
- Entre los grupos de operaciones, y solo a efectos organizativos, crear un grupo llamado "**Notificaciones**". Dentro de este grupo se crearán las operaciones que no requieran análisis, esto es, aquellas que se generan para ser siempre comunicadas, y eso no depende de ninguna regla.



- Para cada operación del grupo de "**Notificaciones**", dirigirse a la herramienta de ***Reglas de suscripción de alertas***, y establecer una propiedad de servicio llamada "`op_type`" y con valor "`alert`".

Con esta propiedad `op_type=alert` le estamos diciendo a las operaciones declaradas (independientemente del grupo en el que estén) que deben dejar pasar los eventos convirtiéndolos siempre en alertas hacia el SDP sin analizar sus datos.

- El resto de operaciones que no sean para enviar alertas directamente al SDP necesitarán reglas de análisis de AE.

1.3 Formato de mensajes JSON, para LIMSP© AE

La estructura básica para construir un evento de AE es un JSON con la siguiente estructura:

Ejemplo1:

```
{
	"header": {
		"refService": "NOMINA",
		"keyValue": "47857684577J"
	},
	"data": {
		"amount": 153.05
		"concepto": "Nómina febrero"
	}
}
```

Se genera un evento donde se indica un código de servicio de nómina, así como la clave principal del usuario que está realizando la operación.

Como puede verse en el ejemplo anterior, la estructura del evento es:

- "**header**": Es el bloque que define la cabecera del evento. En general este bloque se utiliza para condicionar el tratamiento que la Plataforma realizará con el evento enviado.
- "**data**": Es el bloque que define los datos del evento. En general este bloque se utiliza para informar de los datos de negocio o comunicación con el cliente.

1.3.1 Uso de la cabecera "header"

En la cabecera del evento debe establecerse un conjunto de campos predefinidos que condicionan la forma en la que la plataforma deberá procesarlo. Estos campos pueden ser:

- "**id**": Opcional. De existir debe ser un identificador alfanumérico de hasta 24 caracteres que identifique en la entidad a la transacción que se ha generado. Su uso es para efectos de trazabilidad y consulta posterior del estado del mensaje.
- "**refCompany**": Opcional. Es la referencia o código de la organización que contendrá todos los datos de los clientes/usuarios finales. De no establecerse se tomará una organización configurada por defecto.
- "**refService**": Es la referencia o código de operación bancaria que origina el evento. Puede tratarse de cualquier código definido en la organización, por ejemplo el de una compra con tarjeta de crédito en comercio, el código de retiro de efectivo en cajero, el de una transferencia entre cuentas, etc… Todos estos códigos deben estar predefinidos en la herramienta de ***Catálogo de operaciones*** de la plataforma.
- "**keyName**": Opcional. De existir, su valor deberá ser el nombre de una clave secundaria de cliente/usuario. Las claves secundarias de usuario son una forma alternativa de identificar a cada

usuario. Ejemplos de claves alternativas para identificar a usuarios pueden ser su número de tarjeta de crédito (pan), su número GSM, su dirección de email, su cuenta en Twitter, o quizás alguna que hubiera podido ser clave principal pero que finalmente no se utilizó como tal. Ejemplos del valor de este campo podrían ser "*pan*", "*gsm*", "*email*", "*id_usario*", etc..., y en general cualquier nombre de clave de usuario que declaremos en la plataforma como tal. Este parámetro debe establecerse en conjunto con "keyValue".

Si no existe este parámetro se asumirá que el evento hace referencia a la clave principal de usuario (refUser). La clave principal del usuario puede ser un ID de cliente del banco, un número documento nacional de identidad, etc… En general debería ser un código que identifique **inequívocamente** y de **forma permanente** a un único cliente/usuario. El valor de esta clave deberá encontrarse en "keyValue".

- "**`keyValue`**": Su valor deberá ser el valor de la clave principal o secundaria de cliente/usuario especificada por el campo "keyName". Por ejemplo sería el ID del cliente, su número de documento de identificación, el número de tarjeta para la clave "*pan*", el número de teléfono GSM para la clave "*gsm*", la dirección de email para el nombre de clave "*email*", etc… Este parámetro debe establecerse en conjunto con "keyName".

1.3.2 Uso del bloque de datos "data"

El bloque "data" se utiliza para establecer el bloque de datos de la operación bancaria, siendo estos tratados de forma genérica por la plataforma, ya sea porque van a procesarse por reglas de negocio configuradas a medida de los datos, o porque son datos que se quieren comunicar en las notificaciones finales a usuarios/clientes.

La plataforma no predefine que datos pueden enviarse en la operación bancaria, si bien los diferentes campos posibles deben ser declarados previamente en la herramienta de Gestión de tramas, indicando en cada trama bancaria que datos puede contener si estos van a ser útiles para su procesado en la plataforma.

Datos típicos de este bloque son:

- Fecha y hora en la que se produce la operación.
- Importe/monto de la operación.
- Número de tarjeta o de cuenta donde se realiza la operación
- País donde se produce la operación
- Identificador de comercio donde se produce la operación

El nombre y formato interno que tiene cada campo del bloque depende totalmente de cómo se declaren en la definición de la trama bancaria a utilizar y de su uso en las diferentes reglas de negocio que se creen en la plataforma.

Se muestra otro ejemplo de un posible evento.

Ejemplo2:

```
{
        "header": {
                "id": "254365t4",
                "refCompany": "MIBANCO",
                "refService": "MOV_COM",
                "keyName": "pan",
                "keyValue": "4632522753245795"
```

```
        },

        "data": {
                "amount": 153.05,
                "mcc": 5001,
                "fecha": "20150914",
                "hora": "17:36:23"
        }
}
```

Ejemplo de un evento de medios de pago que llega con la clave de cliente "pan" y contiene el identificador de transacción interna de la plataforma de medios de pago.

1.4 Formato de mensajes JSON, para LIMSP© SDP

La estructura básica para construir una notificación de SDP es un JSON con la siguiente estructura:

Ejemplo 3:

```
{
        "header": {
                "id": "t9n4n4ncn3r",
                "refCompany": "MIBANCO",
                "refService": "OTP",
                "keyName": "gsm",
                "keyValue": "+123456789"
        },
        "data": {
                "codigoOtp": "768954A"
        }
}
```

Ejemplo de envío de una notificación con un código OTP.

Como puede verse en el ejemplo anterior, la estructura del evento es:

- "**header**": Es el bloque que define la cabecera del evento. En general este bloque se utiliza para condicionar el tratamiento que la Plataforma realizará con el evento enviado.
- "**data**": Es el bloque que define los datos del evento. En general este bloque se utiliza para informar de los datos de negocio o comunicación con el cliente. El formato es idéntico al utilizado en LIMSP© AE.

1.4.1 Uso de la cabecera "header"

En la cabecera del evento debe establecerse un conjunto de campos predefinidos que condicionan la forma en la que la plataforma deberá procesarlo. Estos campos pueden ser:

- "**`id`**": Debe ser un identificador alfanumérico de hasta 24 caracteres que identifique en la entidad el mensaje que se está generando. Su uso es para efectos de trazabilidad y consulta posterior del estado del mensaje.

- "**`refCompany`**": Opcional. Es la referencia o código de la organización que contendrá todos los datos de los clientes/usuarios finales. De no establecerse se tomará una organización configurada por defecto. Es equivalente al uso de la propiedad "`loginEnterprise`".

- "**`refService`**": Es el servicio que el banco ofrece a sus usuarios/clientes y es la base de la plataforma sobre la que esta contabiliza el envío del mensaje. En la plataforma SDP a este servicio se le suele llamar también contrato. Para SDP se utiliza como equivalencia de `refContract`, si bien en AE se refiere al servicio de alertas sobre el que se establecen las reglas y al que se suscriben los usuarios/clientes.

- "**`keyName`**": Opcional. De existir, su valor deberá ser el nombre de una clave secundaria de cliente/usuario. Las claves secundarias de usuario son una forma alternativa de identificar a cada usuario. Ejemplos de claves alternativas para identificar a usuarios pueden ser su número de tarjeta de crédito (pan), su número GSM, su dirección de email, su cuenta en Twitter, o quizás alguna que hubiera podido ser clave principal pero que finalmente no se utilizó como tal. Ejemplos del valor de este campo podrían ser "*pan*", "*gsm*", "*email*", "*id_usario*", etc..., y en general cualquier nombre de clave de usuario que declaremos en la plataforma como tal. Este parámetro debe establecerse en conjunto con "keyValue".

  Si no existe este parámetro se asumirá que el evento hace referencia a la clave principal de usuario (refUser). La clave principal del usuario puede ser un ID de cliente del banco, un número documento nacional de identidad, etc… En general debería ser un código que identifique **inequívocamente** y de **forma permanente** a un único cliente/usuario. El valor de esta clave deberá encontrarse en "keyValue".

- "**`keyValue`**": Opcional. Su valor deberá ser el valor de la clave principal o secundaria de cliente/usuario especificada por el campo "keyName". Por ejemplo sería el ID del cliente, su número de documento de identificación, el número de tarjeta para la clave "*pan*", el número de teléfono GSM para la clave "*gsm*", la dirección de email para el nombre de clave "*email*", etc… Este parámetro debe establecerse en conjunto con "keyName".

- "**`refProduct`**": Opcional. Es la referencia de la aplicación o producto backend generador de mensajes, y al que ha de pertenecer el contrato. Se recomienda no utilizar este parámetro en el JSON, y en su lugar establecer su valor como login de autentificación de la aplicación en caso de uso de Adaptador WS, y en caso de JMS añadir además la propiedad JMS "`refProduct`" con su valor en los mensajes en la cola.

1.4.2 Uso del bloque de datos "data" con estructura anidada

El bloque "data" se utiliza para establecer el bloque de datos del mensaje a comunicar, siendo estos datos tratados de forma genérica por la plataforma, y generalmente utilizados en las plantillas.

La plataforma no predefine que datos pueden enviarse.

Datos típicos de este bloque son:

- Fecha y hora en la que se produce la operación.
- Importe/monto de la operación.

- Número de tarjeta o de cuenta donde se realiza la operación
- Nombre de la persona que realiza la operación
- Nombre del comercio donde se produce la operación

El nombre y formato interno que debe tener cada campo del bloque depende totalmente de cómo se deban utilizar en las plantillas.

El contenido de cada uno de los elementos del bloque puede ser un valor textual o un conjunto de datos, ya sea en forma de lista o de conjunto de valores.

Cuando los datos forman una lista, a cada uno de los elementos de los datos se le aplica una transformación añadiéndole un índice con su posición en la lista. Esta transformación permite mantener la compatibilidad con el formato de datos anterior.

Cuando uno de los elementos del bloque contiene un conjunto de datos los nombres se expanden estableciendo el nombre completo con los elementos padre.

Ejemplo de JSON con estructuras de datos complejas

Ejemplo4:

```
{
	"header": {
		"id": "254365t4",
		"refCompany": "MIBANCO",
		"refService": "MOV_COM",
		"keyName": "pan",
		"keyValue": "4632522753245795"
	},
	"data": {
		"amount": 153.05,
		"mcc": 5001,
		"fecha": "20150914",
		"hora": "17:36:23",
		"movimientos": [
			{
				"num": "1",
				"concepto": "Gasolinera",
				"monto": 73.05
			},
			{
				"num": "2",
				"concepto": "Restaurante",
				"monto": 80.00
			}
		],
		"tarjeta": {
			"pan": "4632 5227 5324 5795",
			"titular": "Luis Aguilar",
```

```
                    "expiración": "12/04/2020"
            }
      }
}
```

Cada uno de los elementos de la lista "movimientos" se traducirá en una estructura como la siguiente:

```
{
      "header": {
            "id": "254365t4",
            "refCompany": "MIBANCO",
            "refService": "MOV_COM",
            "keyName": "pan",
            "keyValue": "4632522753245795"
      },
      "data": {
            "amount": 153.05,
            "mcc": 5001,
            "fecha": "20150914",
            "hora": "17:36:23",
            "movimientos.1.num": "1",
            "movimientos.1.concepto": "Gasolinera",
            "movimientos.1.monto": 73.05,
            "movimientos.2.num": "2",
            "movimientos.2.concepto": "Restaurante",
            "movimientos.2.monto": 80.00,
            "tarjeta.pan": "4632 5227 5324 5795",
            "tarjeta.titular": "Luis Aguilar",
            "tarjeta.expiración": "12/04/2020"
      }
}
```

Ejemplo de un evento de medios de pago que llega con la clave de cliente "pan" y contiene el identificador de transacción interna de la plataforma de medios de pago.

En este caso se muestran los nombres de los datos para pasar de una estructura de datos en forma de árbol a una estructura de datos plana.

Las dos estructuras anteriores son equivalentes, pero hay que considerar que son de uso exclusivo para las plantillas de LIMSP© SDP, ya que LIMSP© AE actualmente (versión 2.5) no puede gestionar estructuras anidadas o expandidas de propiedades.

1.5 Formato de mensajes JSON extendido, para LIMSP© SDP

La estructura para construir una notificación de SDP extendida es un JSON con la siguiente estructura:

Ejemplo5:

```
{
	"header": {
		"id": "0656hg34d4",
		"refCompany": "MIBANCO",
		"refService": "CAMPAIGN1",
		"keyName": "gsm",
		"keyValue": "+123456789",
		"channels": "sms,pns,email",
		"maxChannels": 1,
		"billable": "false",
		"refLang": "es_ES",
		"refApp": "myAppPns"
	},
	 "info": {
		"refAlias": "MOV2VOD",
		"idOpDest": "3411",
		"msgsource": "on-line"
	},
	"data": {
		"nombre": "Fernando",
		"concepto": "Depósito 100"
	},
	"addresses": [{
		"className": "phone",
		"ref": "+123456789"
		},
		{
		"className": "email",
		"type": "to",
		"ref": "aa@bb.cc"
		}
	],
	"contents": [
		{
			"id": "12345",
			"type": "application/pdf",
			"encoding": "base64",
			"name": "deposito-100.pdf",
			"size": 500,
			"value":
"8927r82y1h8yf892yr71238yrh2819ry129rh8219fgh21gfh21gfh217gfh412f08921ghf429h78f",
```

```
                    "access": "private"
                }
        ],
        "mAppData" : [
        {
                "name": "folder",
                "type": "text",
                "attributes": {
                    "store": "true"
                },
                "value": "promociones"
          }
        ]
}
```

Ejemplo de envío de una notificación multi-canal originada por una campaña de promoción.

Como puede verse en el ejemplo, en la estructura del evento se expande el bloque "header" con mucha más información, y se añade además nuevos bloques, con lo que los bloques disponibles son:

- "**header**": Es el bloque que define la cabecera del mensaje. En general este bloque se utiliza para condicionar el tratamiento que la Plataforma realizará con el mensaje enviado.
- "**info**": Permite añadir datos a la cabecera del mensaje. Pueden ser elementos ya conocidos por la plataforma o elementos personalizables para la aplicación. Se establecen en el INFO del mensaje.
- "**data**": Es el bloque que define los datos del envío. Estos datos son utilizados por las plantillas para componer el contenido del mensaje que finalmente hay que entregar al destinatario.
- "**adresses**": Establece una lista con las direcciones (endpoint) del destinatario para cada posible canal por el que se le pueda enviar contenidos (excepto PNS).
- "**contents**": Identifica una lista de contenidos a enviar al destinatario. Son contenidos que se envían sin que sea necesario el uso de plantillas. Puede tratarse por ejemplo de ficheros adjuntos de un email.
- "**mAppData**": Para Push notifications (PNS). Permite enviar un conjunto de parámetros nombre = valor a la M-App.

1.5.1 Uso de la cabecera "header"

En la cabecera del formato extendido pueden establecerse un conjunto de campos predefinidos que condicionan la forma en la que la plataforma deberá procesar los mensajes. Estos campos pueden ser:

- "**id**": Debe ser un identificador alfanumérico de hasta 24 caracteres que identifique en la entidad el mensaje que se está generando. Su uso es para efectos de trazabilidad y consulta posterior del estado del mensaje.

- "**refCompany**": Opcional. Es la referencia o código de la organización que contendrá todos los datos de los clientes/usuarios finales. De no establecerse se tomará una organización configurada por defecto. Es equivalente al uso de la propiedad "`loginEnterprise`".

- "**refService**": Para SDP se utiliza como equivalencia de `refContract`, si bien en AE se refiere al servicio de alertas sobre el que se establecen las reglas y al que se suscriben los usuarios/clientes.

- "**refContract**": Es el servicio que el banco ofrece a sus usuarios/clientes y es la base de la plataforma sobre la que esta contabiliza el envío del mensaje. En la plataforma SDP a este servicio se le suele llamar también contrato.

- "**keyName**": Opcional. De existir, su valor deberá ser el nombre de una clave secundaria de cliente/usuario. Las claves secundarias de usuario son una forma alternativa de identificar a cada usuario. Ejemplos de claves alternativas para identificar a usuarios pueden ser su número de tarjeta de crédito (pan), su número GSM, su dirección de email, su cuenta en Twitter, o quizás alguna que hubiera podido ser clave principal pero que finalmente no se utilizó como tal. Ejemplos del valor de este campo podrían ser "*pan*", "*gsm*", "*email*", "*id_usario*", etc..., y en general cualquier nombre de clave de usuario que declaremos en la plataforma como tal. Este parámetro debe establecerse en conjunto con "keyValue".

  Si no existe este parámetro se asumirá que el evento hace referencia a la clave principal de usuario (refUser). La clave principal del usuario puede ser un ID de cliente del banco, un número documento nacional de identidad, etc… En general debería ser un código que identifique **inequívocamente** y de **forma permanente** a un único cliente/usuario. El valor de esta clave deberá encontrarse en "keyValue".

- "**keyValue**": Opcional. Su valor deberá ser el valor de la clave principal o secundaria de cliente/usuario especificada por el campo "keyName". Por ejemplo sería el ID del cliente, su número de documento de identificación, el número de tarjeta para la clave "*pan*", el número de teléfono GSM para la clave "*gsm*", la dirección de email para el nombre de clave "*email*", etc… Este parámetro debe establecerse en conjunto con "keyName".

- "**refProduct**": Opcional. Es la referencia de la aplicación o producto backend generador de mensajes, y al que ha de pertenecer el contrato. Se recomienda no utilizar este parámetro en el JSON, y en su lugar establecer su valor como login de autentificación de la aplicación en caso de uso de Adaptador WS, y en caso de JMS añadir además la propiedad JMS "`refProduct`" con su valor en los mensajes en la cola.

- "**refMsgLabel**": Opcional. Es la referencia de la etiqueta a utilizar o a crear. Se utiliza para segmentar o diferenciar varios mensajes que comparten la misma etiqueta a nivel de estadísticas.

- "**pnsRefApp**": Opcional. Es la referencia de la M-App a la cual se quiere enviar una notificación Push (PNS).

- "**maxChannels**": Opcional. Es el número máximo de canales por los que debe enviarse el mensaje. Los canales concretos por los que puede salir estarán determinado por las cláusulas del contrato SDP, las propiedades del mismo, o por la propiedad "`channels`" y el bloque "`addresses`" de este mismo JSON.

- "**channels**": Opcional. Es la lista de canales (en un contenido de referencias separadas por comas) por las que la aplicación espera poder enviar el mensaje. Por ejemplo "`sms,email,pns`"…

- "**billable**": Opcional. Es para indicar con "`true`" o con "`false`" si el mensaje enviado debe ser facturado al usuario/cliente final o no.

- "**refLang**": Opcional. Es el código para el idioma localizado, en formato ISO estándar `[ISO-639]_[ISO-3166]`, por ejemplo "`es_ES`", o "`es_MX`".

- "**deliveryReport**": Opcional. Es para indicar con "`true`" o con "`false`" si la plataforma debe solicitar estados de para el mensaje enviado al operador/proveedor, o no.

- "**tsDelivery**": Opcional. Es para indicar el timestamp (con el clásico formato de los sistemas UNIX/Linux y Java, en milisegundos desde el 01/01/1970 a las 00:00:00:000) del momento a partir del que puede entregarse el mensaje a su destinatario. Esta funcionalidad está sujeta a licencia.

- "**tsDeliveryEnd**": Opcional. Es para indicar el timestamp (con el clásico formato de los sistemas UNIX/Linux y Java, en milisegundos desde el 01/01/1970 a las 00:00:00:000) del momento final hasta el que puede entregarse el mensaje a su destinatario. Este parámetro junto a "tsDelivery" identifica un intervalo horario de la entrega del mensaje. Esta funcionalidad está sujeta a licencia.

- "**tsExpire**": Opcional. Es para indicar el timestamp (con el clásico formato de los sistemas UNIX/Linux y Java, en milisegundos desde el 01/01/1970 a las 00:00:00:000) del momento en que caduca la entrega del mensaje. Esto es que si no se ha entregado el mensaje (el móvil estaba apagado, mensaje retenido, etc…) ya no se va a entregar, y en caso de que tuviera establecidos reintentos a días posteriores estos no irían más allá de este momento establecido.

- "**tfDelivery**": Opcional. El mismo concepto que "tsDelivery" pero en formato ISO-8601.

- "**tfDeliveryEnd**": Opcional. El mismo concepto que "tsDeliveryEnd" pero en formato ISO-8601.

- "**tfExpire**": Opcional. El mismo concepto que "tsExpire" pero en formato ISO-8601.

- "**subject**": Opcional. Es el campo subject o asunto de los envíos de email.

1.5.2 Uso del bloque "info"

El bloque opcional "**info**" permite añadir elementos adicionales a la cabecera del mensaje cuyo uso es menos habitual, o elementos personalizables.

Los elementos personalizables viajan con el mensaje por el BUS de la plataforma, se pueden utilizar en determinados puntos, pero no se almacenan en la base de datos para su consulta posterior en estadísticas.

Los elementos de la cabecera con un uso más residual suelen ser utilizados para funcionalidades de la plataforma que en el pasado tenían una utilidad pero que actualmente están prácticamente en desuso. Por ejemplo el alias para el envío de mensajes push, los perfiles o forzar la operadora de salida para los mensajes SMS.

1.5.3 Uso del bloque de contenidos "addresses"

El bloque opcional "**addresses**" se utiliza para establecer la lista de las direcciones (o endpoints) de los destinatarios del mensaje.

La recomendación es, en lugar de utilizar este bloque, identificar adecuadamente al usuario en el envío y que estas direcciones de destino sean obtenidas por la propia plataforma a partir de los datos de usuario registrados en el sistema informacional.

Cada elemento de este bloque podrá tener los campos:

- "**ref**": Es la dirección del destinatario.

- "**className**": Es la clase de dirección del destinatario. Posibles valores "phone" para SMS, "email" para Email, "tweet" para Twitter. No se permite el uso de tokens PNS.

- "**type**": Indica el tipo de dirección. Posibles valores: "to" como caso habitual para cualquier clase de contenido, y "cc" y "bcc" para copia y copia oculta en el uso de clase "email".

- "**alias**": Opcionalmente es el nombre (que se utiliza solo para Email) del destinatario que queremos que aparezca en el envío del email, en lugar de su dirección de correo.

1.5.4 Uso del bloque de contenidos "contents"

El bloque opcional "**contents**" se utiliza para establecer los contenidos concretos que se quiere enviar en cada mensaje. Cada elemento de este bloque podrá tener los siguientes campos:

- "**id**": Opcional. Indica un ID de contenido previamente cargado en la plataforma. Se utiliza cuando se quieren enviar contenidos muy grandes o el mismo contenido a muchos usuarios. En estos casos se carga previamente el contenido en la plataforma, con lo que esta devuelve un "idContent". Con este se pueden enviar los mensajes haciendo referencia a dicho contenido ya cargado previamente. De no establecer este campo deben establecerse el contenido directamente en el campo "content".
- "**value**": Opcional. Si no se estableció un "id" entonces es necesario enviar directamente el contenido en este campo. Si se trata de un contenido textual puede introducirse directamente, pero si este contenido puede tener caracteres especiales o ser contenido binario, entonces debe codificarse en base64 e indicar esto en el campo "encoding".
- "**type**": Indica el content-type del contenido a enviar. Debe ser uno de los MIME Types estándares y comúnmente utilizados.
- "**encoding**": Opcional. De establecer este parámetro opcional su valor ha de ser "base64", e indicará que el contenido del campo "content" esta codificado en Base64.
- "**name**": Opcional. Si se trata del envío de un fichero adjunto, debe establecerse en este campo cual será el nombre del fichero que aparecerá para este adjunto.
- "**size**": Opcional. Sirve para indicar el tamaño del contenido a enviar.
- "**access**": Opcional. Sirve para indicar si se trata de un contenido de acceso público (por defecto, si no se establece nada) o privado (para el caso de PNS). Los valores posibles valores son "public" o "private".
- "**refTemplate**": Referencia de la plantilla de contenido a utilizar.

1.5.5 Uso del bloque de datos para la M-App "mAppData"

El opcional bloque "**mAppData**" se utiliza para establecer los datos que deben entregarse a la M-App en el envío de Push Notifications.

Su estructura interna es:

- "**name**": Nombre del campo o dato a entregar a la M-App.
- "**value**": Valor del campo o dato a entregar a la M-App.
- "**type**": Tipo del dato a entregar a la M-App. Típicamente "text".
- "**attributes**": Será una lista de posibles atributos del campo para la M-App. Los posibles usos de estos atributos están delimitados por la plataforma.
    - "**store**": Indicará con valor "true" que el dato debe guardarse en BD para que este dato pueda ser recuperado por la M-App aunque nunca le acabe llegando la comunicación Push. Por defecto se considera "false", con lo que el dato no se guarda en la plataforma.