

# Publish and Subscribe

## Documento de arquitectura

Agesic  
Gerencia de proyectos

### Historial de Revisiones

Fecha	Versión	Descripción	Autor	Aprobado Por
04/03/11	1.0	Versión inicial	Guzmán Llambías	

Nombre actual del archivo: AGESIC-Plataforma-Publish&Subscribe - Arquitectura.odt

Andes 1365 piso 7°  
Montevideo – Uruguay  
Tel./Fax: (+598) 2901.2929\*  
Email: [contacto@agesic.gub.uy](mailto:contacto@agesic.gub.uy)

[www.agesic.gub.uy](http://www.agesic.gub.uy)

## Índice de contenido

1.Introducción.....	3
1.1. Propósito.....	3
1.2.Público objetivo.....	3
1.3.Glosario.....	4
2.Requerimientos relevantes.....	5
2.1.Proveedor de novedades.....	5
2.2.Suscripción.....	5
2.3.Envío de actualizaciones.....	5
2.4.Consumo de novedades perdidas.....	6
3.Vista general del sistema.....	7
4.Vista de componentes.....	9
4.1.Módulo de ruteo de novedades.....	9
4.2.Módulo de trazabilidad.....	13
4.3.Módulo de proveedores y suscriptores.....	13
4.4.Aplicación Web de administración.....	14
5.Vista de servicios.....	15
5.1.Servicio de notificación de novedades.....	15
5.2.Servicio de recepción de novedades.....	17

# 1. Introducción

La Plataforma de Gobierno Electrónico (PGE) es un conjunto de componentes de software y hardware integrados con el propósito de proveer una infraestructura tecnológica base para el desarrollo de la estrategia de Gobierno Electrónico del Estado Uruguayo.

Desde el punto de vista puramente tecnológico, la PGE apunta a implementar una plataforma de tipo SOA a nivel de Estado, que fomente la interoperabilidad entre organismos y entes públicos, donde clientes y servicios intercambian información vía el uso de Web Services.

Actualmente, la plataforma provee un conjunto de soluciones a la integración de aplicaciones tecnológicamente heterogéneas, destacándose el uso de estándares para la comunicación, mecanismos de seguridad avanzados, ruteo inteligente, transformación y validación de mensajes.

En este documento, se presenta la arquitectura de un nuevo servicio de la plataforma denominado Publish and Subscribe con el propósito de ofrecer un servicio de novedades basado en suscripciones para el uso de los organismos públicos.

En la sección 2 se presentan los requerimientos relevantes a la arquitectura. En la sección 3, se presenta una descripción global del servicio Publish/Subscribe, mientras que las secciones 4 y 5, presentan las vistas de componentes y servicios.

## 1.1. Propósito

El propósito de este documento es presentar la arquitectura del servicio Publish/Subscribe de la Plataforma de Interoperabilidad de Gobierno Electrónico.

## 1.2. Público objetivo

El público objetivo de este documento incluye a arquitectos de software, desarrolladores, administradores y cualquier otra persona que esté interesada en la estructura interna del servicio.

## 1.3. Glosario

**Productor:** Es el sistema encargado de producir novedades en la Plataforma.

**Suscriptor:** Es el sistema que desea recibir novedades acerca de un determinado tópico.

**Tópico:** Se utilizará el concepto de tópico para identificar el canal al cual se subscriben los consumidores de novedades, un suscriptor puede estar asociado a múltiples tópicos y un productor puede generar mensajes también en múltiples tópicos. En la interfaz Web del sistema se podrán dar de alta los tópicos y administrar su asociación con productores y consumidores.

**Piggybacking:** es una técnica de transmisión de datos bidireccional en la Capa de red (Modelo OSI). Con esta técnica, en vez de enviar ACK en un paquete individual, éste es incluido dentro del próximo paquete a enviar.

## 2. Requerimientos relevantes

A continuación se describen los requerimientos relevantes para el diseño de la arquitectura que debe cumplir el servicio de Publish and Subscribe de la Plataforma.

### 2.1. Proveedor de novedades

La plataforma debe proveer los mecanismos necesarios para recibir las novedades de sus proveedores.

### 2.2. Suscripción

Para realizar la suscripción se necesita una interfaz en la plataforma de manera que los administradores de la misma puedan agregar o eliminar proveedores y suscriptores. Para esto los suscriptores deberán proporcionar un servicio donde la plataforma hará push de las novedades.

### 2.3. Envío de actualizaciones

1. La aplicación proveedora enviara a la plataforma la novedad deseada.
2. El Middleware recibe la novedad e intenta enviarla a cada uno de los suscriptores.
3. En caso de que el suscriptor reciba el dato, se guarda la traza en un log
4. En caso de que el suscriptor este fuera de línea, entonces el middleware marca la novedad como perdida y guardara la información de la novedad y suscriptor que fallaron para que este la consuma en un futuro.

## 2.4. Consumo de novedades perdidas

Se provee de un Web Service para que cada suscriptor pueda consultar sus novedades perdidas. En caso de tenerlas, luego de consumir la novedad, se borra la referencia de la base de datos y guarda la traza en un log.

### 3. Vista general del sistema

El sistema Publish/Subscribe es un servicio provisto por la Plataforma de Interoperabilidad con el propósito de ofrecer mecanismos sencillos para la notificación de eventos y novedades de un organismo a varios otros organismos interesados.

El sistema Publish/Subscribe está basado en mecanismos de comunicación de tipo broadcast en donde un productor desea notificar cierta información a varios interesados.

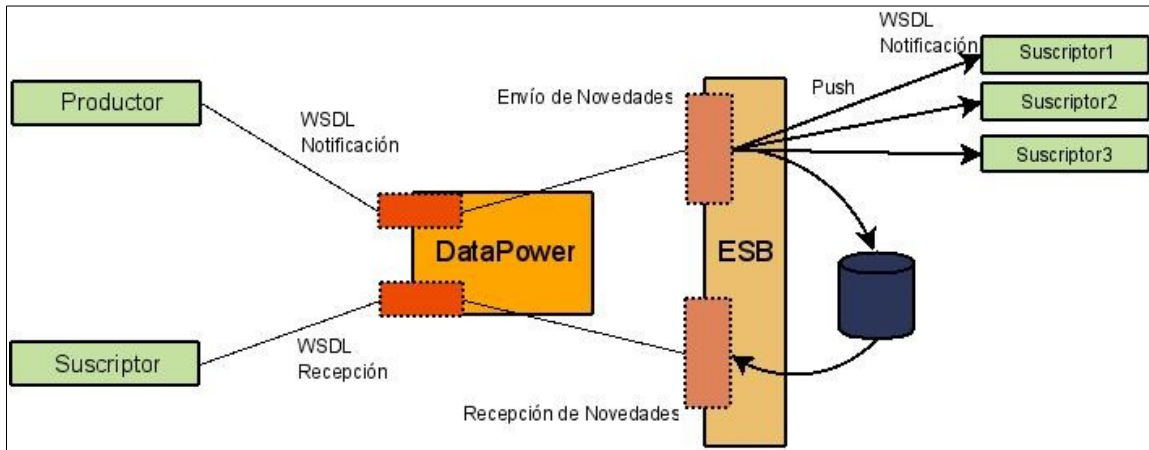
Dentro del contexto de Publish/Subscribe se denomina productor a aquel sistema que produce la información y suscriptor a aquel sistema interesado en recibirla. Los productores comunican la información a los suscriptores por intermedio de canales de comunicación denominados tópicos. Cada vez que un productor envía una novedad a un tópico, el canal se encarga de retransmitirlo a cada uno de los suscriptores interesados.

Actualmente, existen dos mecanismos de entrega de novedades: push y pull. El mecanismo push consiste en que los suscriptores definen previamente dónde recibir las novedades. Cada vez que el servicio recibe una novedad, esta se reenviará al destino definido por el suscriptor. Por otro lado, el mecanismo pull consiste en que el suscriptor consulta a la Plataforma en busca de novedades. Cada vez que el servicio recibe una novedad, esta se almacena y queda a la espera que el suscriptor lo consulte y la retire.

A modo de ejemplo (ficticio), la Dirección General de Registro Civil (DGREC) decidió utilizar el servicio de Publish/Subscribe de la Plataforma para poder comunicar a varios interesados las personas fallecidas y para ello, solicitó la creación de un tópico denominado *fallecidos* dónde poder publicar las novedades referidas a este tema. Por otro lado, la Intendencia de Montevideo (IMM), el Banco de Previsión Social (BPS) y Contaduría General de la Nación (CGN) están muy interesados en recibir estas notificaciones, por lo que decidieron suscribirse al tópico. La IMM y el BPS decidieron utilizar la modalidad push y publicaron un servicio dónde recibir las notificaciones. Por otro lado, la CGN utiliza la modalidad pull y una vez al día consulta a la Plataforma por novedades pendientes. De esta forma, cada vez que la DGREC publica una novedad en el tópico fallecidos, esta es reenviada de forma automática a los servicios de IMM y BPS específicos a este propósito, mientras que las notificaciones para CGN serán almacenadas hasta su posterior retiro.

La figura 1 presenta de forma gráfica como se integra el servicio de Publish/Subscribe a la Plataforma de Interoperabilidad, así como también la interacción entre los actores involucrados. Como se puede ver en la figura, el servicio está alojado en el ESB y está accesible a consumidores y productores

por intermedio de dos servicios publicados en Datapower: Servicio de novedades y Servicio de recepción de novedades. Los productores envían las novedades por intermedio del servicio Envío de Novedades, el cual las reenvía a los suscriptores en modalidad push. Por otro lado, los suscriptores en modalidad pull se comunican con el servicio de recepción de novedades por novedades pendientes.



**Figura 1: Servicio de Publish/Subscribe integrado a la Plataforma de Interoperabilidad**



## 4. Vista de componentes

En esta sección se describen a alto nivel los componentes del sistema y en las siguientes subsecciones se describen cada uno de ellos en detalle:

- El Módulo de Ruteo de Novedades es un componente con lógica orientada a procesamiento de mensajes que resuelve el ruteo de las Novedades.
- El Módulo de Proveedores y Suscriptores actúa como repositorio y gestor de la información requerida para realizar el ruteo, incluyendo la descripción de cuáles son los proveedores y los Web Services de los distintos suscriptores.
- El Módulo de Trazabilidad almacena las trazas de las Novedades y brinda servicios de recuperación y búsqueda.
- Aplicación Web para Administración

### 4.1. Módulo de ruteo de novedades

#### 4.1.1. Descripción funcional

El Módulo de Ruteo de Novedades es responsable del ruteo de las novedades recibidas por el sistema y cuenta con dos modalidades de entrega: push y pull.

En la modalidad pull, el responsable de verificar que no haya novedades pendientes y retirarlas, es el suscriptor. En esta modalidad el sistema recibe las novedades y las almacena a la espera que los suscriptores lo consulten y las retiren.

Por otro lado, en la modalidad de entrega push, el responsable de la entrega es el sistema y cada vez que recibe una novedad, esta la reenvía inmediatamente al destino predefinido por el suscriptor.

El módulo de ruteo implementa estas dos modalidades de entrega mediante dos servicios:

- servicio de ruteo: este componente es el encargado del servicio de entrega en modalidad “push”. Si algún destino opera en modalidad pull, la novedad es enviada al Servicio de Entrega Diferida. Si ocurre algún error en la entrega, se envía la novedad al servicio de novedades diferidas para su almacenamiento a la espera de que el suscriptor acceda a demanda.
- servicio de novedades diferidas: este componente es el encargado del servicio de entrega en modalidad “pull”.

Se recomienda utilizar el servicio de novedades diferidas cuando:

1. El suscriptor no ofrece el servicio de recepción directa de novedades debido a por ejemplo, restricciones en su infraestructura que no le permiten tener una conexión dedicada o un ancho de banda adecuado.
2. El suscriptor ofrece el servicio de recepción directa de novedades, pero por algún problema el envío no se puede realizar.

La información de destinos es obtenida consultando al Módulo de Proveedores y Suscriptores. Cuando las novedades son entregadas este módulo notifica al Módulo de Trazas.

## **4.1.2. Características técnicas**

El Módulo de Ruteo implementa dos mecanismos de ruteo: push y pull.

En el mecanismo de tipo "push", la novedad es recibida y luego enviada por la Plataforma a los suscriptores destino. Se puede decir que se utiliza un modelo "store and forward", es decir, que se acepta la novedad, se canaliza al servicio que la debe entregar y se invoca a los servicios de los suscriptores destino.

En el mecanismo de tipo "pull", el mensaje recibido es almacenado para ser entregado posteriormente a los suscriptores, a través de la operación "Recibir Novedades". Se puede decir que la entrega es a demanda del suscriptor destino. Esta operativa es implementada por el Servicio de Entrega Diferida.

Ambos mecanismos son implementados por el componente de ruteo siendo configurables para cada Suscriptor.

#### 4.1.2.1. Modelo de Ejecución

En esta sección se describen en detalle los modelos de ruteo propuestos para el sistema, presentando primero el modelo de trabajo "push".

En este modelo se plantea el uso de mecanismos de mensajería persistente y transaccional disponibles en JBoss ESB. El funcionamiento del módulo en este modelo es el siguiente:

1. El productor envía una novedad al sistema mediante una invocación al servicio "Recepción de Novedades".
1. El sistema recibe la novedad
2. El sistema almacena la novedad para ser procesada posteriormente. Por ejemplo, mediante colas de mensajes.
3. El sistema responde al proveedor, confirmando la correcta recepción de la novedad.
4. El sistema invoca los Web Services de los suscriptores para entregar la novedad. Si no se puede realizar la entrega esta es almacenada para su posterior procesamiento.
5. Los suscriptores responden indicando la correcta recepción de la misma.

En el modo pull de entrega, la novedad se almacena a la espera que los suscriptores se comuniquen con el sistema para obtener sus novedades. El funcionamiento del módulo con este modelo es el siguiente:

2. El productor envía una novedad al sistema mediante una invocación al servicio "Recepción de Novedades".
3. El sistema recibe la novedad.
4. El sistema almacena la novedad.
5. El sistema responde al proveedor confirmando la correcta recepción de la novedad.
6. El Suscriptor envía una consulta al sistema en busca de novedades pendientes utilizando el servicio "Recibir Novedades". En esta consulta, el sistema envía el identificador (ackId) de la última novedad recibida. Si no recibió novedades hasta el momento, no envía ningún dato.
7. El sistema verifica el identificador (ackId). Si no es vacío, borra la novedad con tal identificador del sistema y comunica al Módulo de Trazas la recepción exitosa.

8. El sistema recupera la próxima novedad pendiente del almacén de novedades.
9. El sistema responde al pedido del suscriptor con la novedad recuperada en el paso 8.

#### **4.1.2.2. Entrega de las notificaciones en modalidad push**

En la modalidad "push" el servicio de ruteo invoca un Web Service predefinido por el suscriptor para la entrega de la novedad. Si esta invocación ocurre con éxito y sin errores se considera como una correcta recepción del mensaje. El destino podrá procesarlo en el momento o utilizar mecanismos asincrónicos con persistencia para su procesamiento (que es lo recomendado). En caso que ocurra un error durante la invocación, el mensaje será almacenado para su posterior consumo en modalidad "pull".

Es posible que el suscriptor reciba novedades duplicadas, lo cual puede ocurrir debido a que el sistema no recibió la confirmación de recepción. En este caso, el suscriptor puede utilizar el encabezado `wsa:MessageID` de WS-Addressing para identificar mensajes repetidos y descartarlo.

#### **4.1.2.3. Entrega de la Novedad en el modo "pull"**

En este modo, la novedad se mantiene en espera hasta que el suscriptor la solicite mediante un llamado al Web Service correspondiente. Para confirmar la recepción del envío, se utiliza el mecanismo de *piggy backing*, en donde el suscriptor confirma la recepción de la novedad en el próximo pedido, sin enviar mensajes específicos para esa tarea. De esta forma, se minimiza el tráfico entre el sistema y los suscriptores.

Al recibir la confirmación, el servicio notifica al módulo de trazabilidad que se ha realizado el envío y se elimina el registro correspondiente de la base de datos.

El servicio es una aplicación basada en JMS y el uso de la Base de Datos Relacional.

## 4.2. Módulo de trazabilidad

### 4.2.1. Descripción funcional

El Módulo de Trazabilidad es responsable del almacenamiento y recuperación de trazas. El almacenamiento se resuelve mediante un Servicio de Registro, un servicio del ESB al que se accede mediante JMS.

Las notificaciones para el Registro de Trazas provienen del Servicio de Ruteo o del Servicio de Entrega Diferida. Las trazas son almacenadas en Base de Datos Relacional.

El Servicio de Consulta de Trazas es un WebService que ofrece la posibilidad de buscar trazas (por fecha, por proveedor y por suscriptor). El acceso a este servicio se realizará desde la aplicación web de administración.

## 4.3. Módulo de proveedores y suscriptores

### 4.3.1. Descripción funcional

Este módulo gestiona el catálogo de Proveedores y Suscriptores y sus distintos servicios Web donde se entregan las novedades de cada suscriptor.

Ofrece servicios interactivos de gestión implementados en la aplicación web de administración, permitiendo realizar las operaciones de alta y mantenimiento.

Como repositorio de información se utiliza una Base de Datos Relacional.

### 4.3.2. Características técnicas

Este módulo realiza el manejo de los catálogos de Proveedores y Suscriptores utilizando la base de datos relacional, los datos para asociar a cada una de estas entidades son:

- Proveedor: Nombre, Descripción, Clave de Acceso
- Suscriptor: Nombre, Descripción, Modo ("push"/"pull"), Endpoint (dirección del WebService para el caso "push")

La clave de acceso del proveedor es opcional, en caso de estar especificada debe ser incluida en el momento del envío de la novedad, como medida de seguridad para evitar el envío de novedades en nombre de otro proveedor.

Este módulo ofrece un WebService con las operaciones necesarias para el mantenimiento de las entidades ya mencionadas, así como para su consulta desde los demás módulos del sistema.

## **4.4. Aplicación Web de administración**

### **4.4.1. Descripción funcional**

Es una aplicación web que interactúa con los módulos de Ruteo y de Proveedores y Suscriptores, permite:

- El mantenimiento (altas, bajas y modificaciones) de proveedores.
- El mantenimiento de suscriptores.
- El mantenimiento de suscripciones (la asociación proveedor-suscriptor).
- La visualización y búsqueda de trazas de entrega de novedades.

El acceso a esta aplicación estará restringida a un usuario administrador, el manejo y declaración de este usuario se hará por fuera de este sistema, utilizando los mecanismos disponibles en JBoss (JAAS desde archivo de propiedades, base de datos, LDAP, etc.).

## **5. Vista de servicios**

En esta sección se describe los servicios Web provistos por el sistema Publish/Subscribe.

### **5.1. Servicio de notificación de novedades**

#### **5.1.1. Descripción general**

Este servicio es utilizado por los productores de novedades para crear una notificación en la plataforma.

#### **5.1.2. Definiciones técnicas**

Con el objetivo de ofrecer una solución suficientemente genérica se plantean las siguientes definiciones técnicas:

- Para cada tópico se debe definir una interfaz de Web Service (WSDL) que deberá ser implementada por el suscriptor en modalidad "push". La única restricción es que el WSDL no defina una respuesta, es decir la respuesta debe contener un "body" vacío.
- El sistema de Publish/Subscribe no utiliza este ni ningún WSDL para su configuración.
- El servicio de novedades aceptará cualquier mensaje SOAP, el cual almacenará y ruteará a los diferentes destinatarios.
- El mensaje SOAP a enviar deberá contener en su cabezal la siguiente información necesaria para el ruteo de las novedades a los suscriptores:
  - identificador del productor del mensaje
  - tópico del mensaje
- Si se desea que la plataforma realice validaciones de acuerdo al WSDL que especifica la estructura de la novedad, éste puede ser especificado en la Plataforma de Interoperabilidad como un ws proxy.

La figura 2 presenta un ejemplo de un mensaje SOAP enviado a este servicio, mientras que la figura 3 presenta un ejemplo de un mensaje SOAP de respuesta.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:not="http://noticias.agesic.gub.uy/" xmlns:ps="http://ps.agesic.gub.uy">
  <soapenv:Header>
    <ps:producer>P1</ps:producer>
    <ps:topic>C</ps:topic>
  </soapenv:Header>
  <soapenv:Body>
    <not:noticia>
      <not:personas>
        <not:persona>
          <not:nombre>Pepe</not:nombre>
          <not:apellido>Gomez</not:apellido>
          <not:fechaNacimiento>2011-03-02T17:57:00.0Z</not:fechaNacimiento>
          <not:ci>1.111.111-1</not:ci>
        </not:persona>
      </not:personas>
    </not:noticia>
  </soapenv:Body>
</soapenv:Envelope>
```

**Figura 2 – Ejemplo de mensaje SOAP para la publicación de una novedad**

```
<soapenv:Envelope
xmlns:ps="http://ps.agesic.gub.uy"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <ps:notificationId>17</ps:notificationId>
  </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
```

**Figura 3 – Ejemplo de mensaje SOAP en respuesta a la publicación de una novedad**



### **5.1.3. Contrato del servicio**

Como se mencionó en la sección 5.1.2. este servicio no tiene definido un contrato específico y acepta cualquier mensaje SOAP que reciba con la única restricción que tenga en su cabezal la información de ruteo necesaria (tópico e identificador del mensaje).

## **5.2. Servicio de recepción de novedades**

### **5.2.1. Descripción general**

Este servicio es utilizado por los suscriptores en modalidad “pull” para obtener sus novedades. Los clientes se comunican con este servicio cada vez que desean obtener sus novedades pendientes.

### **5.2.2. Definiciones técnicas**

Este servicio sigue el esquema de diseño de Servicio Sincrónico de la PGE y para su correcto uso deben enviarse mensajes de acuerdo al siguiente protocolo de comunicación:

1. El cliente manda un mensaje con ack vacío para iniciar el consumo de mensajes.
2. El servicio identifica la primer notificación que no tenga marca de ack y la responde junto con un ackld.
3. El cliente manda otro pedido con el ack recibido en el paso 2.
4. Si todas las notificaciones tienen marca de ack, el servicio envía un mensaje con el body vacío indicando que no hay más notificaciones pendientes. De lo contrario, ir al paso 2.

Las figuras 4 y 5 presentan ejemplos de mensajes de entrada y salida de este servicio.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ps="http://ps.agesic.gub.uy">
  <soapenv:Header/>
  <soapenv:Body>
    <ps:NotificationRequest>
      <subscriber>C1</subscriber>
      <topic>C</topic>
      <ackNotificationId/>
    </ps:NotificationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Figura 4 – Ejemplo de mensaje SOAP para solicitar novedades en modalidad pull**

```
<soapenv:Envelope
xmlns:ps="http://ps.agesic.gub.uy"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <ps:producer>P1</ps:producer>
    <ps:notificationId>17</ps:notificationId>
  </soapenv:Header>
  <soapenv:Body>
    <not:noticia xmlns:not="http://noticias.agesic.gub.uy/">
      <not:personas>
        <not:persona>
          <not:nombre>Pepe</not:nombre>
          <not:apellido>Gomez</not:apellido>
          <not:fechaNacimiento>2011-03-02T17:57:00Z</not:fechaNacimiento>
          <not:ci>1.111.111-1</not:ci>
        </not:persona>
      </not:personas>
    </not:noticia>
  </soapenv:Body>
</soapenv:Envelope>
```

**Figura 5 – Ejemplo de mensaje SOAP de respuesta a la solicitud de novedades pendientes en modalidad pull**

### 5.2.3. Contrato del servicio

La tabla 1 presenta una descripción del contrato del servicio presentando sus operaciones, datos de entrada y salida.

Operación	Entradas	Salidas
Recibir Novedad	RequestNotification: <ul style="list-style-type: none"> <li>• Suscriptor: identificador del suscriptor</li> <li>• Tópico: identificador del tópico</li> <li>• AckID (opcional): indica la última notificación recibida</li> </ul>	Primera notificación sin marca de ack (*)

*Tabla 1 - Contrato del servicio de recepción de novedades*

(\*) La estructura del cuerpo del mensaje depende del tópico al que está suscripto el suscriptor. El cabezal del mensaje debe tener dos elementos relativos a la información de ruteo: producer (origen de la notificación, es el identificador del productor del mensaje) y notificationId (Identificador de la notificación).