# Mensajes de vortex

## Requests HTTP

* mensajeVortex:String  
  Es un parámetro post que tiene en formato JSON un objeto de tipo VortexWrapper
* RESPONSE:String  
  En la respuesta se devuelve en formato JSON un objeto tipo VortexWrapper

### Wrapper HTTP de mensajes vortex

* sessionId: Long  
   Es el ID de sesión HTTP brindado por el server. Inicialmente debería ser -1 al ser enviado para solicitar un nuevo valor.
* mensajes: List< MensajeVortex>  
  Una lista de mensajes MensajeVortex para enviar a la red, o la lista de mensajes recibidos en la response

## Mensajes vortex

* identificación: IdVortex  
  Es un identificador que intenta ser único en la red para el mensaje. Si bien no es posible garantizar que 2 mensajes distintos circulen con el mismo ID en el mismo momento, se busca hacerlo muy difícil.
* tagsDestino: List<String>  
  El conjunto de Tags aplicados al mensaje que permiten determinar a quién va dirigido
* contenido: ContenidoVortex  
  El contenido del mensaje que tiene valor semántico para el que lo reciba (puede ser también un metamensaje)

### ID de un mensaje vortex

* numeroDeSecuencia: Long  
  Es un número consecutivo que el cliente debe incrementar con cada mensaje distinto que envía
* idDelEmisor: String  
  Es una cadena arbitraria que intenta identificar al emisor. No puede asegurarse su exclusividad, pero mientras más rara mejor (actualmente implementada con números aleatoreos).
* hashDelContenido: String  
  Es un checksum de comprobación para verificar que el mensaje se recibió sin alteraciones. Actualmente no está implementado y se manda cualquier valor que permita discriminar un mensaje de otro. En un futuro se implementaría con un prefijo, dos puntos y el valor, sin espacios. Por ejemplo: “CRC8:AB”, “CRC32:AABBCCDD”,”SHA1: 2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12”  
  El nodo que recibe el mensaje debe verificar el hash si tiene el algoritmo con el cual está realizado, si no debe ignorarlo
* timestamp:Long  
  Marca un momento en el tiempo desde el punto de vista del emisor. Permite discriminar los mensajes de emisores idénticos que mandan los mismos mensajes en el mismo orden. (sería mucha coincidencia que además coincidiera el tiempo reloj en que los mandan).

### Contenido Vortex

* tipoContenido:String  
  Un texto que indica qué esperar del atributo valor del contenido. Este valor permite saber cómo parsear el valor
* valor:String  
  En formato texto cualquier contenido que sea de valor para la aplicación. Es dependiente de la aplicación concreta, pero normalmente será un objeto en formato JSON

## Metamensajes Vortex

Los metamensajes vortex, son objetos en formato JSON que se mandan como valor dentro de un MensajeVortex. Cada metamensaje tiene asociado un tipo de contenido. Y todos se envían con el mismo tag: “CHE”.

### SuscribirATags

* Acción: Registra en la red los tags de los mensajes que nos interesa recibir.
* tipoContenido: “vortex.metamensaje. SubscribirATags”
* Formato del valor: {tags:[“TAG1”,”TAG2”]}

### DesuscribirDeTags

* Acción: Desregistra de la red los tags de los mensajes que ya no nos interesa recibir.
* tipoContenido: “vortex.metamensaje. DesuscribirDeTags”
* Formato del valor: {tags:[“TAG1”,”TAG2”]}

### RechazarSuscripcion

* Acción: Desregistra de la red los tags de los mensajes que ya no nos interesa recibir.
* tipoContenido: “vortex.metamensaje. DesuscribirDeTags”
* Formato del valor: {tags:[“TAG1”,”TAG2”]}