



## Plataforma de Vehículo Conectado DGT 3.0

Definición de protocolos e interfaces para la recogida de notificaciones de la señal V16 v0.3

Autor: UTE KPMG ASESORES S.L., PONS SEGURIDAD VIAL S.L, e  
INSPIDE S.L. UTE LEY 18/1982

Última modificación: 25/06/19



## Control de versiones

Versión	Fecha	Editor	Descripción / Comentarios
0.1	25/06/2019	Dir. Técnico	Creación del documento
0.2	15/07/2019	Dir. Técnico	Modificación de interfaces
0.3	22/07/2019	Dir. Técnico	Adaptación del modelo final

Las revisiones se consolidarán en una versión por el editor del documento, una vez consensuada con el resto de actores. El contacto con el editor es: José Gómez Castaño [ [jgcasta@inspide.com](mailto:jgcasta@inspide.com) ]



## 1. Objetivo

El presente documento se elabora con el objetivo de definir la arquitectura, interfaces de comunicación y modelos de datos para la integración de notificaciones procedentes de dispositivos IoT que implementen la señal V16.

## 2. Definiciones

- DGT 3.0: Plataforma de la DGT encargada de la recogida y distribución de información relacionada con seguridad vial.
- Proveedor de información V16 (PV16): Plataforma encargada de proporcionar notificaciones relacionadas con la señal V16 a DGT3.0.

## 3. Documentación de referencia

- Instruccion-18-V-132-Revision-senal-V16-2.pdf.
- Orden PCI/810/2018, de 27 de julio.
- JSON <https://tools.ietf.org/html/rfc6902>.
- Real Decreto 1071/2007 de 27 de julio, por el que se regula el sistema geodésico de referencia oficial en España.
- Geographic information — Well known text representation of coordinate reference systems [<https://www.opengeospatial.org/standards/wkt-crs> ].
- ISO 8601 Date and Time Formats [ <https://www.iso.org/iso-8601-date-and-time-format.html> ].

## 4. Principios generales

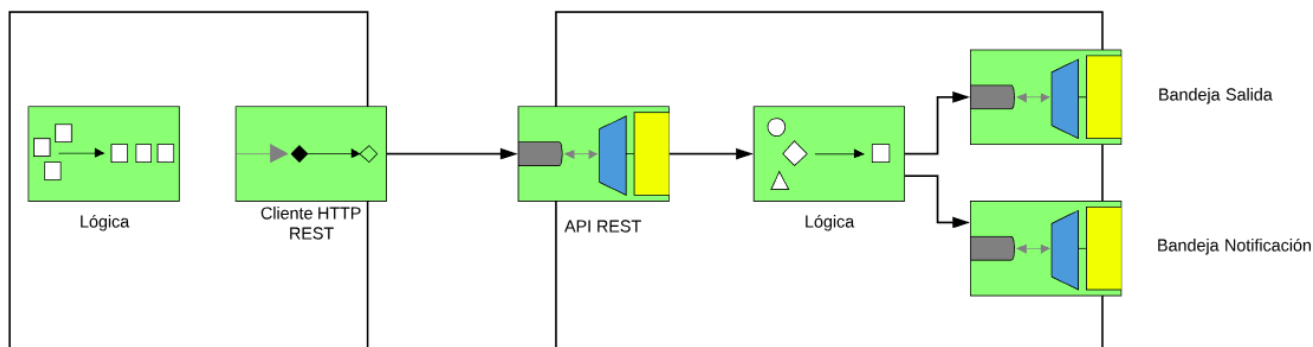
### 4.1. Arquitectura

La arquitectura se encuadra dentro del esquema general planteado en la plataforma DGT3.0 para recepción de notificaciones externas, concretamente la Bandeja de Notificación. La comunicación entre el PV16 y DGT3.0 se realizará nube a nube, no permitiéndose conexiones directas desde los dispositivos IoT que proporcionan la señal V16. La plataforma V16 será responsable de anonimizar al dispositivo origen y de dar la calidad requerida a la información contenida en el mensaje. La plataforma expone un interfaz mediante un API REST que permite una fácil integración de cualquier cliente en la misma.



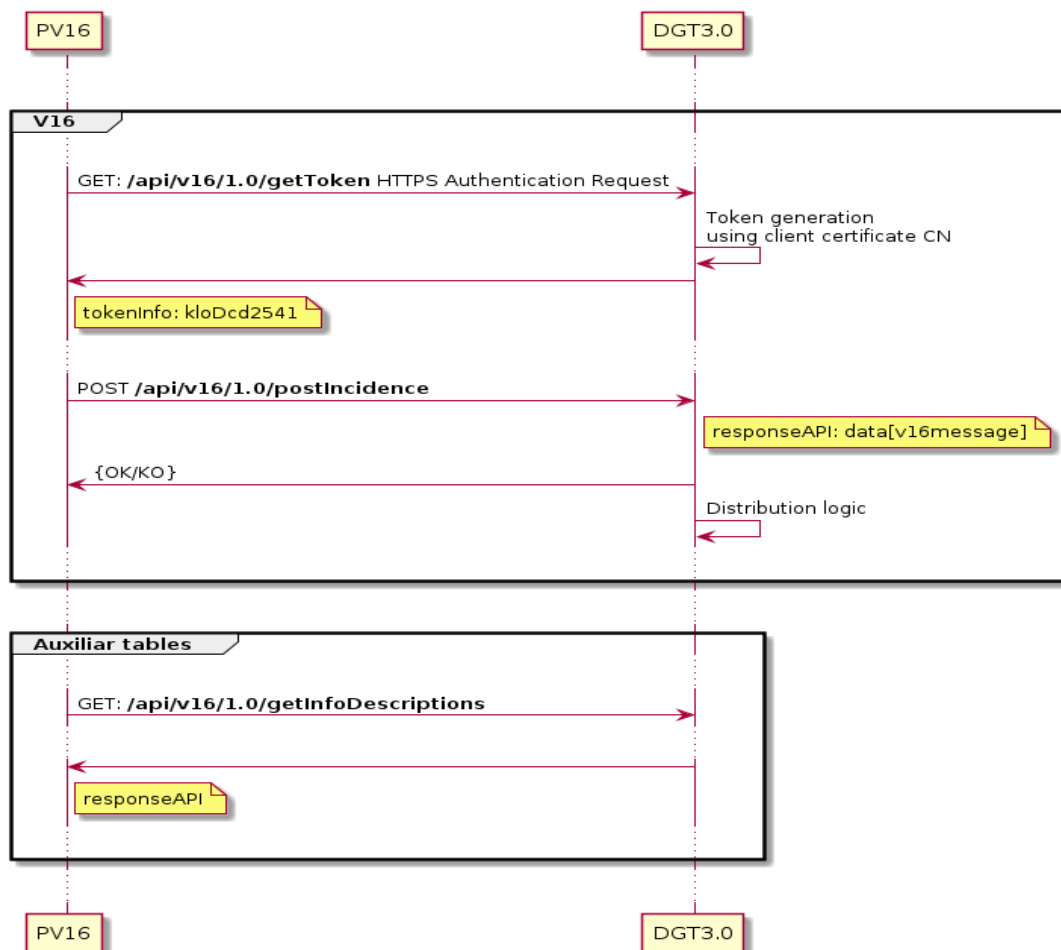
Proveedor V16

Plataforma DGT 3.0



## 4.2. Flujo de información

El flujo de mensajes que se intercambian entre el PV16 y la plataforma se muestra en el siguiente diagrama de secuencia.





Para llevar a cabo las operaciones del API es necesario obtener un token de sesión que caducará de forma aleatoria a lo largo de la misma. La operación que permite obtenerlo es **/api/v16/1.0/getToken** mediante el método GET. El formato de los mensajes se detalla en el apartado Formato de Datos correspondiente al mensaje *tokenInfo*. Para la generación del token es necesario que el cliente está dado de alta en la plataforma y se identifique mediante el certificado X509 proporcionado por la misma.

Una vez obtenido el token correspondiente, se puede invocar a la operación a la que se le envía el mensaje con la información de la incidencia **/api/v16/1.0/postincidence**. Mediante una operación POST se remitirá el mensaje y la plataforma responderá con un mensaje de tipo *responseAPI*, informando del resultado de la recepción del mensaje.

### 4.3. Formato de datos

Todos los mensajes intercambiados entre cada una de las partes que intervienen en la solución estarán en formato JSON [ <https://tools.ietf.org/html/rfc6902> ]. A continuación, se detallan el esquema de los mensajes y un ejemplo de cada uno

#### 4.3.1. **linfoToken**, respuesta a la solicitud de un token

Ejemplo tokenInfo

```
{
  "infoCode": 0,
  "infoDesc": "OK",
  "data": [
    {
      "token": "28a9e96167a6ee0f84bb9c46e8a3b381032f7d9de59ce882539db044e4ee691f"
    }
  ]
}
```

El esquema al que responde este mensaje es

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "tokenInfo",
  "description": "Token response",
  "version": "1.0",
  "properties": {
    "infoCode": {
      "type": "integer"
    },
    "infoDesc": {
      "type": "string"
    },
    "data": {
      "type": "array",
```



```
    "items": [
      {
        "type": "object",
        "properties": {
          "token": {
            "type": "string"
          }
        },
        "required": [
          "token"
        ]
      }
    ]
  },
  "required": [
    "infoCode",
    "infoDesc",
    "data"
  ]
}
```

#### 4.3.2. responseAPI Mensaje de respuesta a un mensaje

##### Ejemplo responseAPI.json

```
{
  "infoCode": 0,
  "infoDesc": "OK",
  "data": []
}
```

El esquema al que responde este mensaje es:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "responseAPI",
  "description": "Generic response v16 API",
  "version": "1.0",
  "properties": {
    "infoCode": {
      "type": "integer"
    },
    "infoDesc": {
      "type": "string"
    },
    "data": {
      "type": "array",
      "items": {}
    }
  }
}
```



```
},
"required": [
  "infoCode",
  "infoDesc",
  "data"
]
}
```

Dentro del elemento “data” de este mensaje se debe incluir el mensaje v16.

```
{
  "actionID": "",
  "token": "",
  "detectionTime": "",
  "eventPosition": "",
  "deviceEventType": "",
  "deviceEventTypeValue": 0,
  "informationQuality": 0,
  "heading": 0,
  "stationType": 0,
  "eventSpeed": 0,
  "ambientTemperature": 0,
  "lanePosition": 0,
  "use": 0
}
```

Los atributos de este mensaje se describen en la siguiente tabla:

Atributo	Significado
actionID	Identificador del evento anonimizando al dispositivo de origen
token	Token generado para plataforma cliente durante la autenticación
detectionTime	Timestamp del instante en el que ocurre el evento, en formato ISO 8601 YYYY-MM-DDTHH:MM:SSZ
eventPosition	Posición del vehículo en formato WKT ISO 19111
deviceEventType	Tipo de evento con la plataforma
deviceEventTypeValue	Valor correspondiente al tipo de evento. 1 – activación; 2 – permanece encendido
informationQuality	Precisión GPS estimada en metros
heading	Orientación del vehículo en grados desde el Norte hacia el Este
stationType	Modo de transporte
eventSpeed	Velocidad, en nuestro caso 0, ya que el vehículo debe estar en reposo



ambientTemperature	Temperatura ambiente
lanePosition	Carril en el que se encuentra el vehículo. 0 - arcén
use	Uso correspondiente al modo de transporte

Los valores de cada uno de los atributos se pueden consultar en la operación correspondiente al API de las tablas auxiliares.

#### 4.4. Protocolos de comunicación

Para permitir un envío simple y seguro de las notificaciones desde la nube del PV16, se utiliza una conexión HTTPS. De esta manera, cada mensaje enviado será procesado y se devolverá al cliente PV16 la confirmación correspondiente.

#### 4.5. Posicionamiento

De acuerdo con el Real Decreto 1071/2007 de 27 de julio, por el que se regula el sistema geodésico de referencia oficial en España, para indicar la posición en longitud y latitud de un evento, se dará en coordenadas grados sexagesimales, con 5 decimales, en sistema de referencia ERTS89, EPSG4326, correspondiente a WGS 84.

#### 4.6. Escala de tiempo y sincronización

Un aspecto crítico tener en consideración en todas las partes software y hardware del sistema, es la sincronización entre los diferentes elementos.

Se utilizará la escala de tiempo UTC, al estar ser un patrón internacional de hora, independiente de zonas horarias.

La sincronización entre dispositivos se realizará usando el protocolo NTP, al ser el protocolo estándar de distribución de hora, utilizando por servidores stratum 1 en España en el Real Instituto y Observatorio de la Armada para servir de patrón nacional de la unidad básica de tiempo (\*). Además, se listan a continuación otras posibles fuentes

URL	IP	Ubicación
hora.roa.es	150.214.94.5	Real Instituto y Observatorio de la Armada, San Fernando (Cádiz).
minuto.roa.es	150.214.94.10	Real Instituto y Observatorio de la Armada, San Fernando (Cádiz)
hora.rediris.es	130.206.3.166	RedIRIS, Madrid
ntp.cesca.cat	84.88.0.106	CESCA, Barcelona

(\*)

<http://www.armada.mde.es/ArmadaPortal/page/Portal/ArmadaEspañola/cienciaobservatorio/prefLang-es/06Hora>





## 4.7. Seguridad

Las comunicaciones estarán cifradas en todos los puntos mediante protocolo criptográfico TLS basado en certificados x.509v3. La autenticación de clientes se realizará mediante el uso de certificados x.509v3 otorgados por la CA de DGT 3.0 a cada una de las partes. Será por tanto de aplicación la Ley 59/2003 de 19 de diciembre de firma electrónica.

El certificado de cliente se otorgará cada responsable de la nube cliente, una vez cumplimentado el formulario de petición correspondiente. XXXXXXXXXXXX aquí la referencia al formulario XXXXXXXXXXXX. Este certificado permitirá realizar la autenticación y autorización del cliente.

Para asegurar el origen de las comunicaciones, se implementará además un filtrado a nivel 3 OSI, para evitar accesos desde orígenes no autorizados. Se recabará de los participantes el direccionamiento IP desde la que se llevarán a cabo los envíos de mensajes al broker MQTT.

## 5. Mensajes a intercambiar

Los mensajes que se reciben en la plataforma están destinados a encapsular la información mínima necesaria para informar de la activación de la señal V16 en un instante y localización determinadas. El formato de los mensajes será JSON.

## 6. Interfaz API REST

Este interfaz es responsabilidad de DGT 3.0 para la recepción de datos de manera síncrona descritos más arriba. Este permite peticiones GET y POST. Su funcionalidad permite la recepción de los datos de incidencia proporcionados desde el PV16.

La documentación detallada del API REST estará disponible en la url expuesta por DGT 3.0 en formato SWAGGER 2.0. Esto permitirá no solo la consulta actualizada de dicho API, sino facilitar la generación automática de clientes de dicho API.

Las conexiones de los clientes se llevarán a cabo siguiendo las premisas que se especifican en el apartado de seguridad, usando protocolo HTTPS para conseguir que todas las comunicaciones sean cifradas. Los clientes tendrán que tener instalado un Certificado Digital x509 v3, emitido por la CA de la plataforma DGT 3.0, válido para la empresa u organismo cliente que realiza la conexión.

Los endpoint y operaciones básicas que expondrá este API tendrán la siguiente sintaxis:

```
baseURL entorno preproduccion: BASE_URL_PRE a definir
baseURL entorno producción:    BASE_URL_PRO a definir
baseURN:                        /api/v16/version/operacion
baseURN documentación:         /swagger-ui.html
```



Al estar basado en SWAGGER la publicación del API, es posible utilizar la utilidad swagger-codegen-cli para generar una aplicación completa, a partir del fichero JSON obtenido de la descarga BASE\_URL/v2/api-docs (requerido certificado)

Un ejemplo de generación de cliente se encuentra descrito a continuación:

```
java -jar swagger-codegen-cli-2.4.1.jar generate \
  -i v16api_1.0.json \
  --api-package es.xxxxxx.dgt30.v16api.client.api \
  --model-package es.xxxxxx.dgt30.v16api.client.model \
  --invoker-package es.xxxxxx.dgt30.v16api.client.invoker \
  --group-id es.xxxxxx \
  --artifact-id es.xxxxxx.dgt30.v16api \
  --artifact-version 0.0.1-SNAPSHOT \
  -l java \
  --library resttemplate \
  -o DGT30v16apiClient
```

## 6.1. Obtención de información auxiliar

Permite la descarga de informaciones auxiliares y básicas necesarias para la interpretación de mensajes o del entorno del sistema de información semafórica. En principio se identifican las siguientes, sin perjuicio de que vayan apareciendo más a medida que se necesiten. Dentro del elemento data de un mensaje responseAPI se incluye el objeto correspondiente, así por ejemplo para infoDescriptions

```
{
  "infoCode": 0,
  "infoDesc": "OK",
  "data": [
    {
      "infoCode": 0,
      "infoDesc": "OK"
    },
    {
      "infoCode": 99,
      "infoDesc": "error general"
    }
  ]
}
```

### Esquema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
```



```
"title": "getInfoDescriptions",
"description": "Info descriptions in response v16 API messages",
"version": "1.0",
"properties": {
  "infoCode": {
    "type": "integer"
  },
  "infoDesc": {
    "type": "string"
  },
  "data": {
    "type": "array",
    "items": [
      {
        "type": "object",
        "properties": {
          "infoCode": {
            "type": "integer"
          },
          "infoDesc": {
            "type": "string"
          }
        }
      }
    ],
    "required": [
      "infoCode",
      "infoDesc"
    ]
  }
},
"required": [
  "infoCode",
  "infoDesc",
  "data"
]
}
```

**URI:** baseUrl/api/v16/1.0/getInfoDescriptions

**Método:** GET.

**Utilidad:** Permite obtener el listado de la codificación de informaciones.

```
{
  "infoCode": 0,
  "infoDesc": "OK"
}
```

**URI:** baseUrl/api/v16/1.0/getDeviceEventTypes

**Método:** GET.

**Utilidad:** Permite obtener el listado de la codificación de los tipos de eventos que son el origen de la comunicación.



```
{
  "deviceTypeId": 1,
  "deviceTypeDesc": "Device Type description 1"
}
```

**URI:** baseUrl/api/v16/1.0/getDeviceEventTypeValues/{deviceTypeId}

**Método:** GET.

**Utilidad:** Permite obtener el listado de la codificación de los tipos de los valores correspondientes a un deviceType concreto.

```
{
  "eventTypeValueId": 1,
  "eventTypeValueDesc": "Device Type Value description 1"
}
```

**URI:** baseUrl/api/v16/1.0/getStationTypes

**Método:** GET.

**Utilidad:** Permite obtener el listado de la codificación de los tipos de modos de transporte.

```
{
  "stationTypeId": 1,
  "stationTypeDesc": "Station type description 1"
}
```

**URI:** baseUrl/api/v16/1.0/getUses

**Método:** GET.

**Utilidad:** Permite obtener el listado de la codificación de los usos.

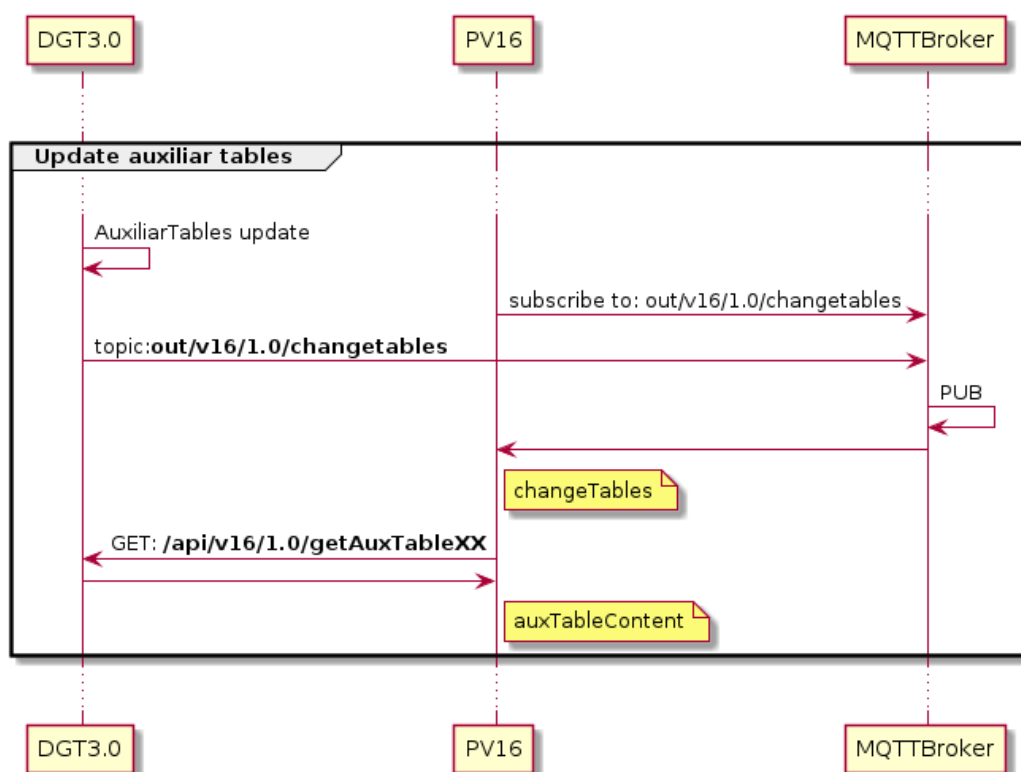
```
{
  "useId": 1,
  "useDesc": "Use description 1"
}
```

## 6.2. Cambios en las tablas auxiliares

Para notificar a los clientes PV16 de cambios en el contenido de las tablas auxiliares, se enviará un mensaje a un tópico MQTT al que se podrán suscribir los interesados.

Tópico MQTT out/v16/1.0/changetables

El flujo de notificación y obtención de la tabla correspondiente se muestra en la siguiente figura



### 6.3. Envío de evento de activación o desactivación de V16

Permite el envío de un mensaje indicando la activación o desactivación de una señal V16. Este mensaje irá insertado en el elemento “data” del mensaje responseAPI, descrito anteriormente en el apartado 4.3.2.,

**URI:** baseUrl/api/v16/1.0/postincidence

**Método:** POST

El mensaje enviado será compuesto como en el ejemplo:

```
{
  "actionID": "1234",
  "token": "550e8400e29b41d4a71644665545478e",
  "detectionTime": "2019-07-22T09:59:00Z",
  "eventPosition": "POINT(-3.52351 40.53256)",
  "deviceEventType": "z0",
  "deviceEventTypeValue": 1,
  "informationQuality": 5,
  "heading": 45,
  "stationType": 7,
  "eventSpeed": 0,
  "ambientTemperature": 10,
```



```
    "lanePosition": 0,  
    "use": 0  
}
```

### Esquema del mensaje:

```
{  
  "$schema": "http://json-schema.org/draft-04/schema#",  
  "type": "object",  
  "title": "v16message",  
  "description": "v16 API message content",  
  "version": "1.0",  
  "properties": {  
    "actionID": {  
      "type": "string"  
    },  
    "token": {  
      "type": "string"  
    },  
    "detectionTime": {  
      "type": "string"  
    },  
    "eventPosition": {  
      "type": "string"  
    },  
    "deviceEventType": {  
      "type": "string"  
    },  
    "deviceEventTypeValue": {  
      "type": "integer"  
    },  
    "informationQuality": {  
      "type": "integer"  
    },  
    "heading": {  
      "type": "integer"  
    },  
    "stationType": {  
      "type": "integer"  
    },  
    "eventSpeed": {  
      "type": "integer"  
    },  
    "ambientTemperature": {  
      "type": "integer"  
    },  
    "lanePosition": {  
      "type": "integer"  
    },  
    "use": {  
      "type": "integer"  
    }  
  }  
}
```



```
},
"required": [
  "actionID",
  "token",
  "detectionTime",
  "eventPosition",
  "deviceEventType",
  "deviceEventTypeValue",
  "informationQuality",
  "heading",
  "stationType",
  "eventSpeed",
  "ambientTemperature",
  "lanePosition",
  "use"
]
}
```

## 7. Requisitos

La plataforma cliente debe cumplir una serie de requisitos con objeto de tener una información de calidad de esta:

- Se debe enviar un mensaje de activación por cada evento.
- Se debe enviar un mensaje cada 60 segundos durante el tiempo que esté activado.
- Se requiere una precisión de 5 metros en posicionamiento.
- Se requiere enviar la precisión estimada en la posición.
- Desde el instante de activación hasta el primer envío a la plataforma no deben excederse los 60 segundos.