

## TABLA DE CONTENIDO

<b>Tabla de figuras .....</b>	<b>2</b>
<b>1 Resumen ejecutivo .....</b>	<b>4</b>
<b>2 Introducción .....</b>	<b>5</b>
<b>2.1 Objetivo .....</b>	<b>5</b>
<b>2.2 Justificación .....</b>	<b>5</b>
<b>2.3 Alcance.....</b>	<b>5</b>
<b>3 Estado del arte.....</b>	<b>7</b>
<b>4 Metodología .....</b>	<b>8</b>
<b>4.1 Decodificación de mensajes tipo SIGMET y AIRMET. ....</b>	<b>9</b>
4.1.1 Primera línea del mensaje .....	11
4.1.2 Nombre de la FIR/UIR/CTA .....	11
4.1.3 Fenómeno observado o pronosticado .....	11
4.1.4 Lugar.....	11
4.1.5 Nivel .....	24
4.1.6 Movimiento o movimiento previsto .....	25
4.1.7 Cambios de intensidad.....	25
4.1.8 Fenómeno .....	26
4.1.9 Consideraciones adicionales .....	28
<b>4.2 Implementación del sistema automático de decodificación .....</b>	<b>29</b>
4.2.1 Implementación de mensajes tipo SIGMET .....	30
4.2.2 Implementación de mensajes tipo AIRMET .....	33
4.2.3 Implementación de mensajes tipo METAR.....	34
4.2.4 Implementación de mensajes tipo TAF.....	34
4.2.5 Implementación de mensajes tipo Avisos de Aeródromo.....	35
<b>5 Resultados.....</b>	<b>36</b>
<b>6 Conclusiones.....</b>	<b>46</b>
<b>6.1 Limitaciones .....</b>	<b>46</b>
<b>6.2 Recomendaciones.....</b>	<b>46</b>
<b>6.3 Trabajo futuro .....</b>	<b>47</b>
<b>7 Bibliografía y referencias.....</b>	<b>48</b>
<b>8 Siglas y acrónimos.....</b>	<b>49</b>

## TABLA DE FIGURAS

Figura 1. Simbología de fenómenos meteorológicos utilizada por la AEMet.....	5
Figura 2. Archivos del proyecto .....	8
Figura 3. Página web correspondiente a la decodificación por parte del usuario de mensajes tipo SIGMET y AIRMET con el pertinente aviso de precaución de la única atención a mensajes elaborados a partir de normativa OACI .....	9
Figura 4. Limitaciones de tráfico de la licencia estándar de Google Maps usada bajo Javascript .....	10
Figura 5. Lista de fenómenos en mensajes tipo SIGMET, extraído de la Guía SIGMET de la AEMet .....	26
Figura 6. Lista de fenómenos en mensajes tipo AIRMET, extraído de la Guía AIRMET de la AEMet .....	27
Figura 7. Página web correspondiente a la decodificación automática de mensajes tipo SIGMET .....	29
Figura 8. Ejemplo de fichero cuyo contenido es el propio mensaje, en este caso se trata de un mensaje tipo AIRMET.....	30
Figura 9. Esquema correspondiente a la obtención del listado de SIGMET y el contenido de un determinado mensaje.....	30
Figura 10. Esquema correspondiente a la obtención del listado de AIRMET y el contenido de un determinado mensaje.....	33
Figura 11. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo METAR .....	34
Figura 12. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo TAF .....	34
Figura 13. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo Aviso de Aeródromo .....	35
Figura 14. Consola web en la que se miden los diferentes parámetros en la ejecución de la herramienta .....	36
Figura 15. Ejemplo de AIRMET con localización dada a través de un punto .....	37
Figura 16. Ejemplo de AIRMET con localización dada a través de una línea de longitud y una dirección.....	38
Figura 17. Ejemplo de SIGMET con localización dada a través de una línea por dos puntos y una dirección.....	39
Figura 18. Ejemplo de SIGMET con localización dada a través de direcciones respecto a dos líneas de latitud y longitud .....	40
Figura 19. Ejemplo de SIGMET con localización dada a través de un polígono .....	41

Figura 20. Ejemplo de METAR.....	42
Figura 21. Ejemplo de TAF .....	43
Figura 22. Ejemplo de Aviso de Aeródromo .....	44
Figura 23. Cancelación de un AIRMET .....	45

# 1 RESUMEN EJECUTIVO

---

En el presente documento se explica el trabajo desarrollado desde enero de 2016 hasta agosto del mismo año realizado por Miguel González Calvo en colaboración con la Agencia Estatal de Meteorología y con el Doctor Miguel Ángel Pelacho Aja, jefe del Servicio de Aplicaciones Aeronáuticas de la Agencia.

El trabajo consiste en el desarrollo de una página web que brinda al usuario la posibilidad de descodificar mensajes tipo SIGMET y AIRMET introducidos por él mismo, así como consultar y descodificar los mensajes tipo SIGMET y AIRMET vigentes a nivel mundial a excepción de zonas donde los criterios de elaboración no se corresponden con los definidos por la OACI; y mensajes tipo METAR, TAF y Avisos de Aeródromo en los principales aeródromos del territorio nacional.

Se explican los elementos presentes en los mensajes tipo SIGMET y AIRMET (a excepción de los de ciclones tropicales y cenizas volcánicas cuya descodificación no se ha desarrollado) que se han debido estudiar detalladamente para poder desarrollar el código referido a la descodificación de los mismos.

Se explica de igual manera la implementación de los sistemas de descodificación automática con el código desarrollado para la descodificación individual de mensajes tipo SIGMET y AIRMET y el código cedido por la Agencia desarrollado por Karina Melo Anginoni para la descodificación individual de mensajes tipo METAR, TAF y Avisos de Aeródromo.

Se analizan asimismo los resultados obtenidos del proceso de creación de la herramienta y las conclusiones obtenidas a partir del mencionado análisis a través de las limitaciones encontradas, recomendaciones y trabajo futuro a desarrollar.

## 2 INTRODUCCIÓN

### 2.1 OBJETIVO

Se pretende diseñar una herramienta que permita la descodificación de mensajes SIGMET y AIRMET.

### 2.2 JUSTIFICACIÓN

Dada la importancia de esta tipología de mensajes en el ámbito aeronáutico, se precisa que la información se presente de forma clara, visual y concisa al público objetivo, que en este caso se define principalmente como los tripulantes de aviación civil.

### 2.3 ALCANCE

En un inicio el proyecto se limitaba a la descodificación de SIGMET y AIRMET, mientras que con el avance del mismo se propuso la posibilidad de mostrarlos gráficamente en un mapa gracias a la simbología de fenómenos desarrollada por la Agencia Estatal de Meteorología (de aquí en adelante AEMet).


























 Tormentas	 Llovizna	
 Ciclón tropical	 Lluvia	
 Fuerte línea de turbonada	 Nieve	
 Turbulencia moderada	 Chubasco	 Granizo
 Turbulencia fuerte	 Ventisca alta de nieve	
 Ondas orográficas	 Calima fuerte de arena o polvo	
 Engelamiento moderado en la aeronave	 Tempestad extensa de arena o polvo	
 Engelamiento fuerte en la aeronave	 Calima extensa	
 Niebla extensa	 Neblina extensa	
 Materiales radiactivos en la atmósfera	 Humo extenso	
 Erupción volcánica	 Precipitación engelante	
 Oscurecimiento de las montañas	 Nube de cenizas visible	

Figura 1. Simbología de fenómenos meteorológicos utilizada por la AEMet

Tras el desarrollo de esta herramienta, se convino interesante la posibilidad de incluir mensajes METAR, TAF y avisos de aeródromo aprovechando el trabajo desarrollado por Karina Melo Anginoni, consistente en la exclusiva descodificación de los mensajes.

Por tanto, el alcance del proyecto se establece como la realización, desarrollo y pruebas de una herramienta web de carácter visual basada en mapas interactivos que muestre descodificados los siguientes mensajes: SIGMET (excluyendo los de cenizas volcánicas y ciclones tropicales por su excesiva diversidad de tipologías en su elaboración), AIRMET a nivel mundial; y METAR, TAF y avisos de aeródromo a nivel nacional.

### 3 ESTADO DEL ARTE

---

Existen diversas utilidades para la decodificación y presentación en mapa de esta tipología de mensajes, que se distinguirán diferentemente:

En el caso de los mensajes SIGMET, destaca la desarrollada por la *National Oceanic and Atmospheric Administration*<sup>1</sup>. (NOAA en adelante), pero que no hace uso de una simbología visual y clara para el público objetivo definido. Referente a iniciativas privadas, se encuentra la página web *Skyvector*<sup>2</sup>, aunque presenta similar problemática.

En el caso de los mensajes AIRMET, la herramienta desarrollada por la NOAA también ofrece la posibilidad de visualizar algunos de los mensajes, resultando a todas luces insuficiente puesto que únicamente se presentan en ciertas áreas (principalmente la región de los Estados Unidos, que es el área objetiva definida por la organización).

En el caso de los mensajes METAR y TAF, se vuelven a encontrar la iniciativa pública de la NOAA<sup>3</sup> y la privada también a manos de *Skyvector*<sup>2</sup>. La problemática en este caso surge por la falta de aeródromos peninsulares con no excesiva importancia comercial, y sin embargo pueden ser interesantes para aviación general. Además, en el caso de *Skyvector*, no se presenta una decodificación del mensaje sino la presentación del propio código.

---

<sup>1</sup> <http://www.aviationweather.gov/sigmet>

<sup>2</sup> <https://skyvector.com>

<sup>3</sup> <http://www.aviationweather.gov/metar> y <http://www.aviationweather.gov/taf>

## 4 METODOLOGÍA

La totalidad del proyecto fue desarrollada en un sistema bajo *OS X El Capitán 10.11.5* actuando de servidor gracias a *Apache* para *Unix v2.4.18*.

El control de versiones se realizó gracias a *git v2.7.4* con una réplica del repositorio en *Github*<sup>4</sup>.

En un inicio se valoraron las diferentes posibilidades para el desarrollo de esta herramienta, que quedan resumidas en: herramienta web, herramienta nativa para PC y herramienta nativa para *Smartphone* o *Tablet*. Por su gran versatilidad y la sencilla posibilidad de inclusión futura de mapas interactivos, se decidió optar por la herramienta web basada en los estándares HTML5, ECMAScript (a través de Javascript y jQuery) y CSS (pese a todavía no ser estándar, dada su actual alta implementación en los navegadores, se incorporaron funcionalidades CSS3).

La parte correspondiente al servidor se programó en PHP principalmente para obtener los mensajes de archivos .txt como se explicará en la parte correspondiente. En la siguiente figura se puede ver la estructura principal del proyecto.

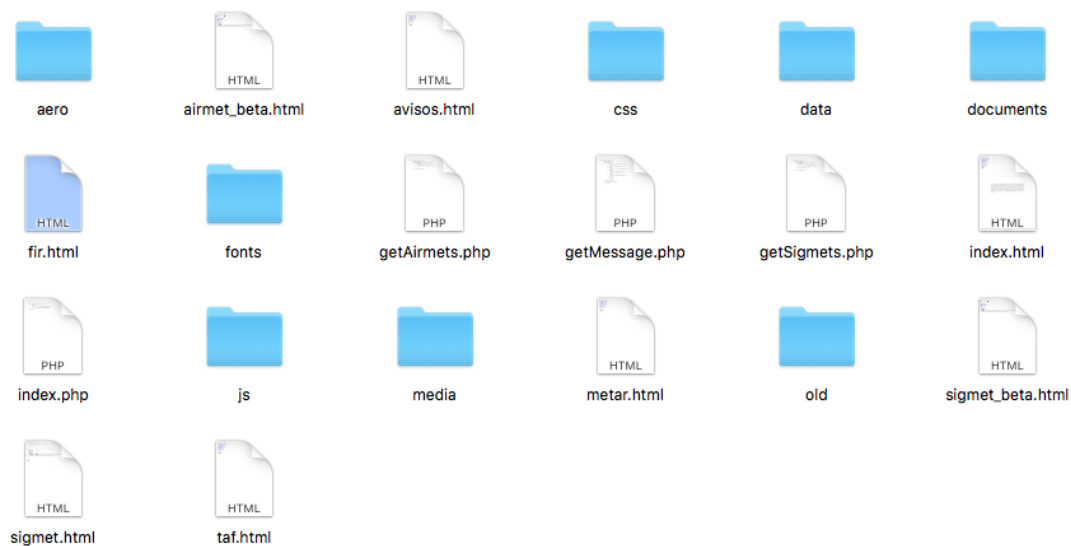


Figura 2. Archivos del proyecto

<sup>4</sup> <https://github.com/Liuz9/Meteo>



## 4.1 DECODIFICACIÓN DE MENSAJES TIPO SIGMET Y AIRMET.

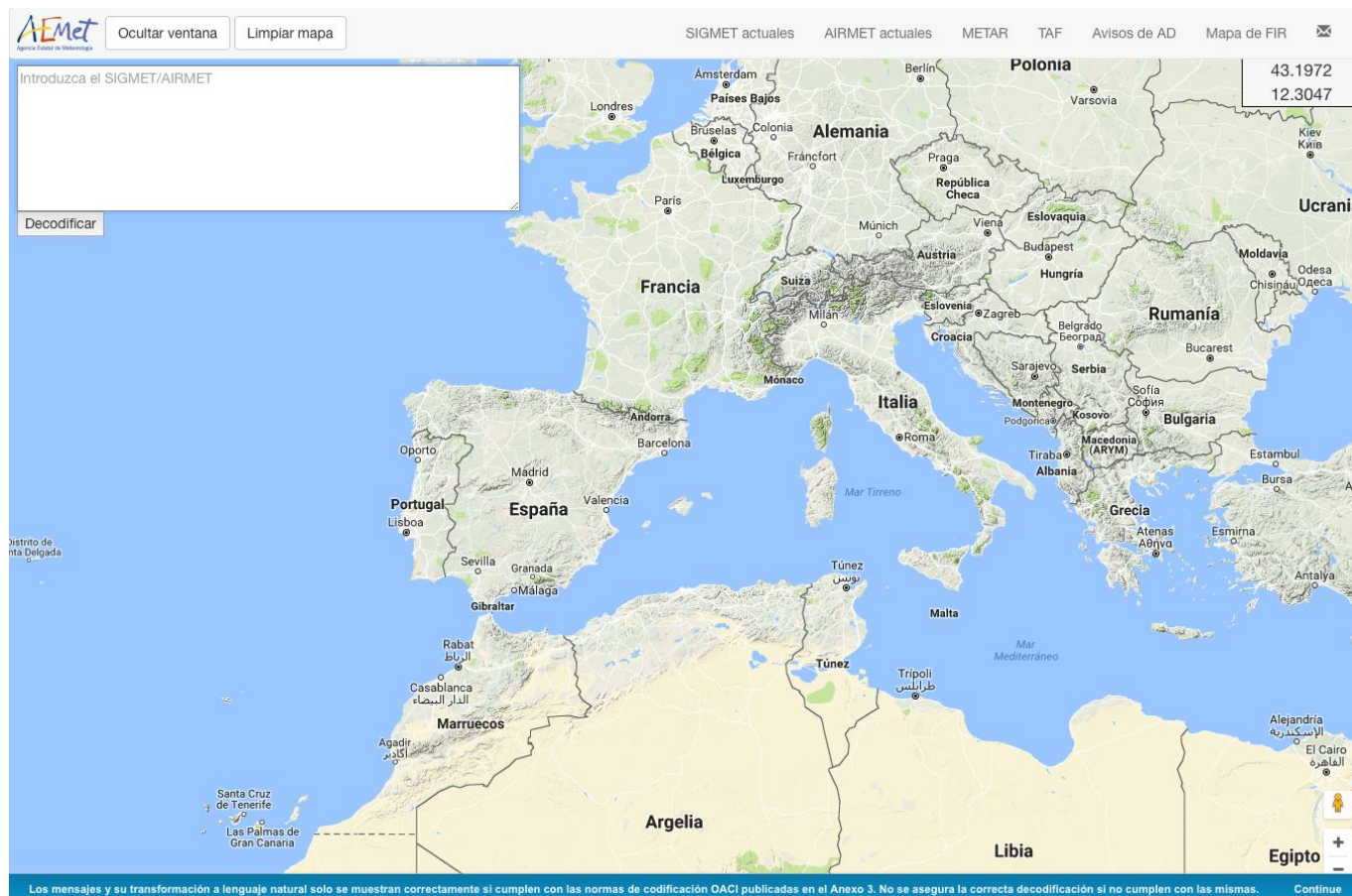


Figura 3. Página web correspondiente a la decodificación por parte del usuario de mensajes tipo SIGMET y AIRMET con el pertinente aviso de precaución de la única atención a mensajes elaborados a partir de normativa OACI

El primer conflicto surgió a la hora de enfrentarse a la propia decodificación de mensajes SIGMET; analizando los diferentes métodos para aproximarse a una solución final se optó por el uso de expresiones regulares para identificar patrones dentro del propio código, utilizando la herramienta online *RegExr*<sup>5</sup> para realizar el desarrollo, probarlo y comprobar que funcionaba correctamente.

Por requerir un tratamiento demasiado exhaustivo y estar fuera del alcance del proyecto, se excluyeron los SIGMET de cenizas volcánicas y ciclones tropicales.

Una vez se logró la decodificación de mensajes tipo SIGMET, se afrontó la referente a mensajes del tipo AIRMET, con similar estructura pero pequeñas peculiaridades y otra colección completamente diferente de fenómenos. Por ello, se aprovechó gran parte del código desarrollado para SIGMET realizando las variaciones oportunas para adaptarlo a mensajes AIRMET.

Es necesario recalcar la inmensa variedad de formatos SIGMET y AIRMET existentes y la falta de acuerdo en su elaboración, incluso dentro de países con una misma norma al respecto.

<sup>5</sup> <http://regexpr.com>

Por ello, esta herramienta se ciñe estrictamente a las normas de elaboración publicadas por la OACI<sup>6</sup>.

A modo de ejemplo, compárense ambos mensajes SIGMET, de los Estados Unidos el primero y de Europa el segundo:

```
SIGMET VICTOR 1 VALID UNTIL 221830
SIGMET
NC MD DC DE VA AND CSTL WTRS
FROM EMI TO 150SE SIE TO 180ESE ECG TO 170SE ECG TO 40WSW LYH TO
EMI
OCNL SEV TURB BTN FL210 AND FL320. DUE TO WNDSHR ASSOC'D WITH
JTST. RPTD BY A321 E170. CONDS ENDG 1830Z.
```

```
LECM SIGMET 1 VALID 221628/221830 LEVA-
LECM MADRID FIR/UIR EMBD TS OBS AT 1615Z WI N4230 W00810 -
N4340 W00650 - N4340 W00550 - N42 W00720 TOP FL450 MOV NE NC=
```

Nótese que el primer mensaje no se ajusta a las recomendaciones OACI (ver tabla de recomendaciones).

Tras la satisfactoria elaboración del código de descodificación, fue planteada la posibilidad de incluir la información descodificada de cada mensaje en un mapa en la propia página web. Concluida interesante esta posibilidad por parte del tutor y alumno, se decidió el desarrollarlo mediante la API de Google Maps<sup>7</sup> que, pese a las limitaciones de tráfico que impone, se creen suficientes para el alcance de esta herramienta.

Límites de uso estándar	
<b>Usuarios de la API estándar:</b> <ul style="list-style-type: none"> <li>Gratis hasta exceder 25 000 cargas de mapas por día durante 90 días consecutivos</li> </ul>	<b>Habilitar la facturación de pago según el uso para desbloquear cuotas mayores:</b> Al exceder los límites de uso gratuito, se facturarán USD 0,50 por 1000 solicitudes adicionales, hasta 1 000 000 diarias.
	<a href="#">HABILITAR FACTURACIÓN</a>

Figura 4. Limitaciones de tráfico de la licencia estándar de Google Maps usada bajo Javascript<sup>8</sup>

Mediante una adecuada implementación de las funcionalidades de la API de Google Maps, se desarrollaron funcionalidades tales como la presentación de SIGMET y AIRMET en puntos, polígonos o líneas y la inclusión del FIR en el que el SIGMET o AIRMET se encontraba como una capa más del mapa a través de sus coordenadas obtenidas a través de una base de datos adecuada.

Para el desarrollo de la descodificación se contempló el usar las Guías SIGMET y AIRMET desarrolladas por el propio SAA de AEMet que contienen en sus anexos resúmenes tabulados de los formatos SIGMET y AIRMET. Sin embargo, en aras de la versatilidad de la aplicación, se

<sup>6</sup> Anexo 3 al Convenio sobre Aviación Civil Internacional: Servicio meteorológico para la navegación aérea internacional. Apéndice 6.

<sup>7</sup> <https://developers.google.com/maps/>

<sup>8</sup> <https://developers.google.com/maps/documentation/javascript/usage>

decide el hecho de usar la tabla A6-1 del Apéndice 6 del Anexo 3 de OACI como se señaló anteriormente.

Tómese en consideración que los mensajes pueden contener o no la cabecera pertinente, por lo que la aplicación la obviará de estar presente por no aportar información adicional relevante a la del propio mensaje.

A continuación se hará un análisis de los elementos de los SIGMET y AIRMET comunes a ambos (i.e. todos salvo los fenómenos):

#### 4.1.1 Primera línea del mensaje

Compuesta por el **indicador de lugar de FIR/CTA, identificación** (SIGMET o AIRMET), **período de validez** e **indicador de lugar de OVM**.

Utilizada la siguiente expresión regular se obtienen los elementos anteriores.

```
/([A-Z]{4}) (AIRMET|SIGMET) (\w{1,3}) VALID (\d{6}\/\d{6}) ([A-Z]{4})-
```

A través de esta primera línea se reconoce si el mensaje es de tipo SIGMET o AIRMET para la posterior decodificación del fenómeno.

#### 4.1.2 Nombre de la FIR/UIR/CTA

Teniendo en cuenta que dependiendo del tipo de mensaje se pueden encontrar FIR, CTA, UIR y FIR/UIR, se desarrolla la siguiente expresión regular para la obtención de la región de espacio aéreo:

```
/\w{4} ((\w|\s){1,200}) (FIR\/UIR|FIR|CTA|UIR) /
```

#### 4.1.3 Fenómeno observado o pronosticado

Atendiendo al origen del mensaje, éste puede haber sido predicho u observado directamente. Además, se puede incluir la hora de predicción u observación en el propio mensaje, siendo ésta opcional. Véase la expresión regular:

```
/(OBS|FCST) ( AT \d{4}Z)?/;
```

#### 4.1.4 Lugar

Se trata del elemento con mayor diversidad del mensaje. Indica la localización del fenómeno a través de diferentes métodos como se ilustra en los siguientes ejemplos:

```
SOFN54 NOFN50 N2020 W07005 N2706 W07306
N48 E010
N OF N1515 AND W OF E13530
W OF E1554
N OF LINE S2520 W11510 – S2520 W12010
WI N6030 E02550 – N6055 E02500 – N6050 E02630
ENTIRE FIR
ENTIRE CTA
```

- **Latitud y longitud de un punto:**

Según terminología OACI:

*Nnn[nn] Wnnn[nn] o Nnn[nn] Ennn[nn] o Snn[nn] Wnnn[nn] o Snn[nn] Ennn[nn]*

Se usa la herramienta del API de Google Maps para construir *Markers*, *google.maps.Marker*. El símbolo utilizado es el correspondiente al fenómeno, que se explicará posteriormente. Además, se añade una ventana de información para incluir la información que ya habrá sido decodificada del mensaje.

Una vez representado el marcador en la posición determinada por la latitud y longitud del punto, se añadirán los eventos correspondientes para que la ventana se abra una vez se coloca el ratón encima del símbolo y permanezca abierta hasta que se cierre deliberadamente. Asimismo, se mostrará el código del mensaje en la parte inferior de la ventana del navegador para poder consultar el mensaje original, con la salvedad de que éste desaparecerá una vez se retire el ratón de encima del símbolo.

Esta estructura de *Marker* se utilizará de igual manera para el resto de formas de suministrar la posición del fenómeno, con salvedades detalladas a continuación en cada una de ellas.

```
function createMarker(location) {
    var icon = {
        url: 'media/symbols/'+sigmet.phenomenon+'.png', // url
        scaledSize: new google.maps.Size(50, 50), // scaled
        size

        origin: new google.maps.Point(0,0), // origin
        anchor: new google.maps.Point(25, 25) // anchor
    };

    var marker = new google.maps.Marker({
        position: location,
        map: map,
        icon: icon,
    });

    var contentString = '<table><tr><th colspan="2"
id="tabletitle">'+sigmet.tipo+'</th></tr><tr><th>Fenómeno</th><td>'+si
gmet.hazard+'</td></tr><tr><th>FIR</th><td>'+sigmet.fir+'</td></tr><tr>
<th>Secuencia</th><td>'+sigmet.number+'</td></tr><tr><th>Comienzo</th>
<td>'+sigmet.begins+'</td></tr><tr><th>Fin</th><td>'+sigmet.ends+'</t
d></tr><tr><th>Base</th><td>'+sigmet.base+'</td></tr><tr><th>Tope</th>
<td>'+sigmet.top+'</td></tr><tr><td
colspan="2">'+sigmet.forecast+'</td></tr><tr><td
colspan="2">'+sigmet.movement+'</td></tr><tr><td
colspan="2">'+sigmet.changes+'</td></tr><tr><td
colspan="2">'+sigmet.position+'</td></tr></table>';

    var infowindow = new google.maps.InfoWindow({
```

```

        content: contentString
    });
    var code = sigmet.code.replace(/[A-Z]{4}-/, function
myFunction(x) {return x+'<br/>';});
    console.log(code);

    marker.addListener('mouseover', function() {
        infowindow.open(map, marker);
        $('#codigo').html(code);
        $('#codigo').show();
    });

    marker.addListener('mouseout', function() {
        $('#codigo').hide();
    });

    marker.addListener('click', function() {
        if(infowindow.isOpen()) {
            infowindow.close();
        }
        else {
            infowindow.open(map, marker);
        }
    });

```

- **A una determinada dirección de una línea de latitud o longitud dada:**

Según terminología OACI:

*Nnn[nn] Wnnn[nn] o Nnn[nn] Ennn[nn] o Snn[nn] Wnnn[nn] o Snn[nn] Ennn[nn] o*

*N OF Nnn[nn] o S OF Nnn[nn] o N OF Snn[nn] o S OF Snn[nn] o [AND]*

*W OF Wnnn[nn] o E OF Wnnn[nn] o W OF Ennn[nn] o E OF Ennn[nn] o*

*[N OF,NE OF,E OF,SE OF,S OF,SW OF,W OF,NW OF] [LINE] Nnn[nn] o Snn[nn] Wnnn[nn] o  
Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]*

Nótese que en estos casos, salvo en el apartado de LINE en el que se construye la línea a través de dos puntos, la línea se construye a través de una latitud o longitud dada.

Nótese también que existe la posibilidad de denotar una región del FIR/UIR/CTA a través de dos líneas y dos direcciones respecto a las mismas.

El *Marker* se colocará en la intersección de ambas en el anterior caso y en el centro de la línea en caso de tratarse de una única recta.

Se usa la herramienta del API de Google Maps para construir líneas: *google.maps.Polyline*.

```
function createPolyline (polyCoords) {
```

```

var polyline = new google.maps.Polyline({
    path: polyCoords,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 2
});
polyline.setMap(map);
}

```

Para poder dibujar la línea (o las líneas) en el mapa con precisión se ha de obtener la intersección entre la misma y la región para delimitar la misma. Con este fin se tuvo que desarrollar una función que devuelven los cortes con una línea de latitud o longitud fija. Esta función carece de gran precisión por usar matemáticas referidas a geometría plana y no esférica. Se optó por esta posibilidad puesto que la resolución de las ecuaciones en geometría esférica requerían demasiado tiempo de ejecución por ser un entorno web, el cual se ralentizaría demasiado en clientes con ordenadores sin excesiva potencia de cálculo.

Esta aproximación, tras desarrollarla e implementarla, se juzgó suficiente como se verá en las figuras subsecuentes referidas a esta tipología de localización del fenómeno.

```

function getCortes(firData, txt) {
    if(firData != null) {

        var cortes = new Array();

        if(txt.charAt(0)=='N' || txt.charAt(0)=='S') {
            var lat = str2coord(txt);

            for(i = 0;
i<firData.geometry.coordinates[0][0].length; i++) {
                p =
[firData.geometry.coordinates[0][0][i][1],
firData.geometry.coordinates[0][0][i][0]];

                if(i==0) {
                    p_1 =
[firData.geometry.coordinates[0][0][firData.geometry.coordinates[0][0]
.length-1][1],
firData.geometry.coordinates[0][0][firData.geometry.coordinates[0][0].
length-1][0]];
                }
                else {
                    p_1 =
[firData.geometry.coordinates[0][0][i-1][1],
firData.geometry.coordinates[0][0][i-1][0]];
                }
            }
        }
    }
}

```

```

        if((p[0]-lat)*(p_1[0]-lat)<0) {
            m = (p[0]-p_1[0])/(p[1]-p_1[1]);
            cortes.push(p_1[1]+(lat-
p_1[0])/m);
        }
        else if(p_1[0]-lat==0) {
            cortes.push(p_1[1]);
        }
    }
}
else if(txt.charAt(0)=='E' || txt.charAt(0)=='W') {
    var long = str2coord(txt);

    for(i = 0;
i<firData.geometry.coordinates[0][0].length; i++) {
        p =
[firData.geometry.coordinates[0][0][i][1],
firData.geometry.coordinates[0][0][i][0]];

        if(i==0) {
            p_1 =
[firData.geometry.coordinates[0][0][firData.geometry.coordinates[0][0]
.length-1][1],
firData.geometry.coordinates[0][0][firData.geometry.coordinates[0][0].
length-1][0]];
        }
        else {
            p_1 =
[firData.geometry.coordinates[0][0][i-1][1],
firData.geometry.coordinates[0][0][i-1][0]];
        }

        if((p[1]-long)*(p_1[1]-long)<0) {
            m = (p[0]-p_1[0])/(p[1]-p_1[1]);
//      cortes.push((p[1]+p_1[1])/2);
            cortes.push(p_1[0]+(long-
p_1[1])*m);
        }
        else if(p_1[1]-long==0) {
            cortes.push(p_1[0]);
        }
    }
}

cortes = cortes.sort(function(a, b){return a-b});
console.log(cortes);

return cortes;
}

```

- **Polígono a través de sus coordenadas:**

Según terminología OACI:

*WI Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – [Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]]*

Se usa la herramienta del API de Google Maps para construir polígonos: *google.maps.Polygon*.

El *Marker* se sitúa en este caso en el centro del polígono usando una función específica del API correspondiente a la herramienta anterior.

```
function createPolygon(polyCoords) {
    var polygon = new google.maps.Polygon({
        paths: polyCoords,
        strokeColor: '#FF0000',
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: '#FF0000',
        fillOpacity: 0.35
    });
    polygon.setMap(map);
}
```

- **FIR entero:**

Según terminología OACI:

*ENTIRE FIR*

Por estar ya representado el FIR entero, lo único que se realiza es colocar el *Marker* en el centro del FIR, información obtenida a través de la propia base de datos de FIR/UIR/CTA.

- **CTA entera:**

Según terminología OACI:

*ENTIRE CTA*

Debido a que la base de datos utilizada no contempla las CTA y ser muy poco frecuente en los mensajes emitidos, se optó por no incluir esta posibilidad.

#### 4.1.4.1 Consideraciones adicionales.

Nótese que en función de en qué orden se realicen las comprobaciones de las expresiones regulares en las operaciones de *if*, se puede llevar a equívoco y obtener una localización incorrecta.

Por la gracia del ejemplo, véase que en la localización *WI N6030 E02550 – N6055 E02500 – N6050 E02630* si se realiza antes el *if* correspondiente con la expresión regular de un punto (a saber: */ (N|S) (\d{2,4}) (W|E) (\d{3,5}) /*) en lugar de de la correspondiente al polígono



(tipo de localización correcta en este ejemplo) se obtendría un falso positivo, ya que no se trataría de un único punto sino de un polígono como se ha indicado.

En aras de una mayor claridad, se incluye el siguiente código con la correspondiente jerarquía desarrollada para que el lector pueda consultarlo en detalle:

```
{
    if(/(N|NE|E|SE|S|SW|W|NW) OF LINE (N|S) (\d{2,4})
(E|W) (\d{3,5}) - (N|S) (\d{2,4}) (E|W) (\d{3,5})/.test(txt)) {
        var res = /(N|NE|E|SE|S|SW|W|NW) OF LINE
(N|S) (\d{2,4}) (E|W) (\d{3,5}) - (N|S) (\d{2,4})
(E|W) (\d{3,5})/.exec(txt);

        var polyCoords = [];
        polyCoords.push(new
google.maps.LatLng(str2coord(res[2]+res[3]),
str2coord(res[4]+res[5])));
        polyCoords.push(new
google.maps.LatLng(str2coord(res[6]+res[7]),
str2coord(res[8]+res[9])));
        createPolyline(polyCoords);

        sigmet.position = 'Al '+cdirection(res[1])+' de
la línea';

        createMarker(google.maps.geometry.spherical.interpolate(polyCoo
rds[0], polyCoords[1], 0.5));
    }
    else if(/ENTIRE FIR/.test(txt)) {
        sigmet.position = 'FIR completo';
        if(firData != null) {
            createMarker(new
google.maps.LatLng(firData.properties.centlat,
firData.properties.centlong));
        }
        else {
            alert('Error recuperando informacion del
FIR');
        }
    }
    else if(/ENTIRE CTA/.test(txt)) {
        if(firData != null) {
            sigmet.position = 'CTA completa';
            createMarker(new
google.maps.LatLng(firData.properties.centlat,
firData.properties.centlong));
        }
        else {
```

```

                                alert('Error recuperando informacion de
la CTA');
                                }
                                }
                                else if(/WI (N|S) (\d{2,4}) (W|E) (\d{3,5}) -
((N|S)\d{2,4} (W|E)\d{3,5}) - ((N|S)\d{2,4} (W|E)\d{3,5}) ( -
((N|S)\d{2,4} (W|E)\d{3,5}))?( - ((N|S)\d{2,4} (W|E)\d{3,5}))?( -
((N|S)\d{2,4} (W|E)\d{3,5}))?( - ((N|S)\d{2,4}
(W|E)\d{3,5}))?/.test(txt)) {
                                var res = /WI (N|S) (\d{2,4}) (W|E) (\d{3,5}) -
(N|S) (\d{2,4}) (W|E) (\d{3,5}) - (N|S) (\d{2,4}) (W|E) (\d{3,5}) ( -
(N|S) (\d{2,4}) (W|E) (\d{3,5}))?( - (N|S) (\d{2,4}) (W|E) (\d{3,5}))?( -
(N|S) (\d{2,4}) (W|E) (\d{3,5}))?( - (N|S) (\d{2,4}
(W|E) (\d{3,5}))?/.exec(txt);

                                var polyCoords = [];
                                polyCoords.push(new
google.maps.LatLng(str2coord(res[1]+res[2]),
str2coord(res[3]+res[4])));
                                polyCoords.push(new
google.maps.LatLng(str2coord(res[5]+res[6]),
str2coord(res[7]+res[8])));
                                polyCoords.push(new
google.maps.LatLng(str2coord(res[9]+res[10]),
str2coord(res[11]+res[12])));

                                i = 13;
                                while(res[i]!=null) {
                                        polyCoords.push(new
google.maps.LatLng(str2coord(res[i+1]+res[i+2]),
str2coord(res[i+3]+res[i+4])));
                                        i+=5;
                                }
                                createPolygon(polyCoords);

                                var bounds = new google.maps.LatLngBounds();
                                var i;
                                for (i = 0; i < polyCoords.length; i++) {
                                        bounds.extend(polyCoords[i]);
                                }
                                createMarker(bounds.getCenter());
                                map.setCenter(bounds.getCenter());
                                }
                                else if(/(N|S) OF (N|S) (\d{2,4}) AND (W|E) OF
(W|E) (\d{3,5})/.test(txt)) {
                                var res = /(N|S) OF (N|S) (\d{2,4}) AND (W|E) OF
(W|E) (\d{3,5})/.exec(txt);

                                if(firData != null) {

```

```

        var lat = str2coord(res[2]+res[3]);
        var long = str2coord(res[5]+res[6]);
        var corteslat = getCortes(firData,
res[2]+res[3]);

        var corteslong = getCortes(firData,
res[5]+res[6]);

        corteslat = corteslat.sort(function(a,
b){return a-b});
        corteslong = corteslong.sort(function(a,
b){return a-b});

        console.log(corteslat);
        console.log(corteslong);

        polyCoords = new Array();

        pto = new google.maps.LatLng(lat, long);
        console.log(pto.lat(), pto.lng());

        if(res[1]=='N' && res[4]=='E') {
            polyCoords.push(new
google.maps.LatLng(lat, long));
            polyCoords.push(new
google.maps.LatLng(corteslong.max(), long));
            createPolyline(polyCoords);

            polyCoords = new Array();
            polyCoords.push(new
google.maps.LatLng(lat, long));
            polyCoords.push(new
google.maps.LatLng(lat, corteslat.max()));
            createPolyline(polyCoords);
        }
        else if(res[1]=='N' && res[4]=='W') {
            polyCoords.push(new
google.maps.LatLng(lat, long));
            polyCoords.push(new
google.maps.LatLng(corteslong.max(), long));
            createPolyline(polyCoords);

            polyCoords = new Array();
            polyCoords.push(new
google.maps.LatLng(lat, corteslat.min()));
            polyCoords.push(new
google.maps.LatLng(lat, long));
            createPolyline(polyCoords);

```

```

        }
        else if (res[1]=='S' && res[4]=='W') {
            polyCoords.push(new
google.maps.LatLng(corteslong.min(), long));
            polyCoords.push(new
google.maps.LatLng(lat, long));

            createPolyline(polyCoords);

            polyCoords = new Array();
            polyCoords.push(new
google.maps.LatLng(lat, corteslat.min()));
            polyCoords.push(new
google.maps.LatLng(lat, long));

            createPolyline(polyCoords);
        }
        else if (res[1]=='S' && res[4]=='E') {
            polyCoords.push(new
google.maps.LatLng(corteslong.min(), long));
            polyCoords.push(new
google.maps.LatLng(lat, long));

            createPolyline(polyCoords);

            polyCoords = new Array();
            polyCoords.push(new
google.maps.LatLng(lat, long));
            polyCoords.push(new
google.maps.LatLng(lat, corteslat.max()));

            createPolyline(polyCoords);
        }

        sigmet.position = 'A1
'+direction(res[1])+' de '+res[2]+res[3]+' y '+direction(res[4])+'
de '+res[5]+res[6];

        createMarker(new
google.maps.LatLng(str2coord(res[2]+res[3]),
str2coord(res[5]+res[6])));
    }
    else {
        alert('Error recuperando informacion del
FIR');
    }
}
else if (/ (N|S) OF (N|S) (\d{2,4}) AND (N|S) OF
(N|S) (\d{2,4}) /.test(txt)) {
    var res = / (N|S) OF (N|S) (\d{2,4}) AND (N|S) OF
(N|S) (\d{2,4}) /.exec(txt);

```

```

        if(firData != null) {
            var lat1 = str2coord(res[2]+res[3]);
            var lat2 = str2coord(res[5]+res[6]);
            var corteslat1 = getCortes(firData,
res[2]+res[3]);
            var corteslat2 = getCortes(firData,
res[5]+res[6]);

            cortes = corteslat1;
            lat = lat1;
            var polyCoords;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i]));
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i+1]));
                createPolyline(polyCoords);
                console.log(lat, cortes[i]);
                console.log(lat, cortes[i+1]);
            }

            cortes = corteslat2;
            lat = lat2;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i]));
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i+1]));
                createPolyline(polyCoords);
                console.log(lat, cortes[i]);
                console.log(lat, cortes[i+1]);
            }

            sigmet.position = 'A1
'+cdirection(res[1])+' de '+res[2]+res[3]+' y '+cdirection(res[4])+'
de '+res[5]+res[6];

            createMarker(new
google.maps.LatLng((lat1/2+lat2/2), firData.properties.centlong));
        }
        else {
            alert('Error recuperando informacion del
FIR');
        }
    }
    else if(/(E|W) OF (E|W) (\d{3,5}) AND (E|W) OF
(E|W) (\d{3,5})/.test(txt)) {

```

```

        var res = /(E|W) OF (E|W) (\d{3,5}) AND (E|W) OF
(E|W) (\d{3,5})/.exec(txt);

        if(firData != null) {
            var long1 = str2coord(res[2]+res[3]);
            var long2 = str2coord(res[5]+res[6]);
            var corteslong1 = getCortes(firData,
res[2]+res[3]);
            var corteslong2 = getCortes(firData,
res[5]+res[6]);

            cortes = corteslong1;
            long = long1;
            var polyCoords;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(cortes[i], long));
                polyCoords.push(new
google.maps.LatLng(cortes[i+1], long));
                createPolyline(polyCoords);
            }

            cortes = corteslong2;
            long = long2;
            var polyCoords;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(cortes[i], long));
                polyCoords.push(new
google.maps.LatLng(cortes[i+1], long));
                createPolyline(polyCoords);
            }

            sigmet.position = 'Al
'+cdirection(res[1])+' de '+res[2]+res[3]+' y '+cdirection(res[4])+'
de '+res[5]+res[6];

            createMarker(new
google.maps.LatLng(firData.properties.centlat, long1/2+long2/2));
        }
        else {
            alert('Error recuperando informacion del
FIR');
        }
    }
    else if(/(N|S) OF (N|S) (\d{2,4})/.test(txt)) {
        var res = /(N|S) OF (N|S) (\d{2,4})/.exec(txt);
    }

```

```

        if(firData != null) {
            var lat = str2coord(res[2]+res[3]);

            cortes = getCortes(firData,
res[2]+res[3]);

            var polyCoords;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i]));
                polyCoords.push(new
google.maps.LatLng(lat, cortes[i+1]));
                createPolyline(polyCoords);
//         console.log(lat, cortes[i]);
//         console.log(lat, cortes[i+1]);
            }

            sigmet.position = 'A1
'+cdirection(res[1])+' de la latitud '+res[2]+res[3];

            createMarker(new
google.maps.LatLng(str2coord(res[2]+res[3]),
(cortes.max()+cortes.min())/2));
        }
        else {
            alert('Error recuperando informacion del
FIR');
        }
    }
    else if(/(W|E) OF (W|E) (\d{3,5})/.test(txt)) {
        var res = /(W|E) OF (W|E) (\d{3,5})/.exec(txt);

        if(firData != null) {
            var long = str2coord(res[2]+res[3]);

            cortes = getCortes(firData,
res[2]+res[3]);

            var polyCoords;
            for(i = 0; i<cortes.length; i=i+2) {
                polyCoords = new Array();
                polyCoords.push(new
google.maps.LatLng(cortes[i], long));
                polyCoords.push(new
google.maps.LatLng(cortes[i+1], long));
                createPolyline(polyCoords);
            }
        }
    }
}

```

```

        sigmet.position = 'Al
'+cdirection(res[1])+' de la longitud '+res[2]+res[3];
        createMarker(new
google.maps.LatLng((cortes.max()+cortes.min())/2,
str2coord(res[2]+res[3])));
    }
    else {
        alert('Error recuperando informacion del
FIR');
    }
}
else if (/ (N|S) (\d{2,4}) (W|E) (\d{3,5}) /.test(txt)) {
    var res = / (N|S) (\d{2,4})
(W|E) (\d{3,5}) /.exec(txt);
    sigmet.position = 'Punto '+res[0];
    var location = new
google.maps.LatLng(str2coord(res[1]+res[2]),str2coord(res[3]+res[4]));
    createMarker(location);
}
else {
    if(firData != null) {
        sigmet.position = 'Localización no
reconocida';
        createMarker(new
google.maps.LatLng(firData.properties.centlat,
firData.properties.centlong));
    }
    else {
        alert('Error recuperando informacion del
FIR');
    }
}
}
}

```

Nótese en el código arriba expuesto que se comprueba que la variable *firData* tenga valor, puesto que para la mayoría de posibilidades de localización se necesitan las coordenadas del FIR/UIR/CTA.

#### 4.1.5 Nivel

Indica la altitud a la cual se encuentra presente el fenómeno. Según terminología OACI:

*[SFC]/FLnnn o [SFC]/nnnnM (o) [SFC]/nnnnFT) o FLnnn/nnn o TOP FLnnn o [TOP] ABV FLnnn*

*o*

*CB TOP [ABV] FLnnn WI nnnKM OF CENTRE*

*(o CB TOP [ABV] FLnnn WI nnnNM OF CENTRE) o*



*CB TOP [BLW] FLnnn WI nnnKM OF CENTRE*

*(o CB TOP [BLW] FLnnn WI nnnNM OF CENTRE) o*

*FLnnn/nnn [APRX nnnKM BY nnnKM] [nnKM WID LINE BTN (nnNM WID LINE BTN)] [Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] [– Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]] [– Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]]] (o FLnnn/nnn [APRX nnnNM BY nnnNM] [Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] – Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn] [– Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]] [– Nnn[nn] o Snn[nn] Wnnn[nn] o Ennn[nn]]])*

Para mayor claridad, contéplense los siguientes ejemplos:

FL180 FL050/080 TOP FL390 SFC/ FL070 TOP ABV FL100 FL310/450

CB TOP FL500 WI 270KM OF CENTRE (CB TOP FL500 WI 150NM OF CENTRE)

FL310/350 APRX 220KM BY 35KM

FL390

#### 4.1.6 Movimiento o movimiento previsto

Indica el movimiento (observado o previsto) con dirección y velocidad en algunos casos con referencia a uno de los dieciséis puntos de la brújula o bien estacionario. Según terminología OACI:

*MOV N [nnKMH] o MOV NNE [nnKMH] o MOV NE [nnKMH] o MOV ENE [nnKMH] o MOV E [nnKMH] o MOV ESE [nnKMH] o MOV SE [nnKMH] o MOV SSE [nnKMH] o MOV S [nnKMH] o MOV SSW [nnKMH] o MOV SW [nnKMH] o MOV WSW [nnKMH] o MOV W [nnKMH] o MOV WNW [nnKMH] o MOV NW [nnKMH] o MOV NNW [nnKMH] (o MOV N [nnKT] o MOV NNE [nnKT] o MOV NE [nnKT] o MOV ENE [nnKT] o MOV E [nnKT] o MOV ESE [nnKT] o MOV SE [nnKT] o MOV SSE [nnKT] o MOV S [nnKT] o MOV SSW [nnKT] o MOV SW [nnKT] o MOV WSW [nnKT] o MOV W [nnKT] o MOV WNW [nnKT] o MOV NW [nnKT] o MOV NNW [nnKT])*

*o STNR*

Para mayor claridad, contéplense los siguientes ejemplos:

MOV E 40KMH (MOV E 20KT)

MOV SE

STNR

#### 4.1.7 Cambios de intensidad

Indica los cambios de intensidad previstos para el fenómeno. Según terminología OACI:

*INTSF o WKN o NC*

El significado de cada uno es intensificándose, debilitándose o sin cambios, respectivamente.

#### 4.1.8 Fenómeno

Indica el fenómeno que lleva a expedir el SIGMET o AIRMET. Han de distinguirse los fenómenos correspondientes a mensajes tipo SIGMET y los correspondientes a mensajes tipo AIRMET.

- **Mensajes tipo SIGMET:**

#### **A niveles de crucero (independientemente de la altitud):**

##### **Tormentas**

- oscurecidas
- inmersas
- frecuentes
- línea de turbonada
- oscurecidas por granizo
- inmersas con granizo
- frecuentes con granizo
- línea de turbonada con granizo

**OBSC<sup>(2)</sup> TS**  
**EMBD<sup>(3)</sup> TS**  
**FRQ<sup>(4)</sup> TS**  
**SQL<sup>(5)</sup> TS**  
**OBSC TSGR<sup>(6)</sup>**  
**EMBD TSGR**  
**FRQ TSGR**  
**SQL TSGR**

##### **Ciclón tropical**

- ciclón tropical con vientos en la superficie de velocidad media, en 10 minutos, de 17 m/s (34kt) o más.

**TC (+ nombre del ciclón)**

##### **Turbulencia**

- turbulencia fuerte

**SEV TURB<sup>(7)</sup>**

##### **Engelamiento**

- engelamiento fuerte
- engelamiento fuerte debido a lluvia engelante

**SEV ICE<sup>(8)</sup>**  
**SEV ICE (FZRA)<sup>(9)</sup>**

##### **Ondas orográficas**

- ondas orográficas fuertes

**SEV MTW<sup>(10)</sup>**

##### **Tempestad de polvo**

- tempestad fuerte de polvo

**HVY DS**

##### **Tempestad de arena**

- tempestad fuerte de arena

**HVY SS**

##### **Cenizas volcánicas**

- cenizas volcánicas
- nube radiactiva

**VA (+nombre del volcán si se conoce)**  
**RDOACT CLD**

Figura 5. Lista de fenómenos en mensajes tipo SIGMET, extraído de la Guía SIGMET de la AEMet

• **Mensajes tipo AIRMET:**

**Velocidad del viento en superficie**

- Zonas extensas donde la velocidad media generalizada del viento en superficie es superior a 30 kt (15 m/s) **SFC WSPD (+ velocidad del viento + unidades)**

**Visibilidad en la superficie**

- Zonas extensas donde la visibilidad haya quedado reducida a menos de 5000 m, incluido el fenómeno meteorológico que produce la reducción de la visibilidad. El fenómeno meteorológico se seleccionará de la lista que aparece en la **nota 3**. **SFC VIS (+visibilidad+fenómeno meteorológico que causa la reducción de visibilidad)**

**Tormentas**

- Aisladas sin granizo **ISOL TS**
- Ocasionales sin granizo **OCNL TS**
- Aisladas con granizo **ISOL TSGR**
- Ocasionales con granizo **OCNL TSGR**

**Oscurecimiento de montañas**

- Montañas oscurecidas **MT OBSC**

**Nubes**

- Zonas extensas de cielo nuboso (BKN) o cubierto (OVC) con altura de la base de las nubes a menos de 1000 ft (300 m) del suelo
  - Cielo nuboso **BKN CLD (+ altura base/cima + unidades)**
  - Cielo cubierto **OVC CLD (+ altura base/cima + unidades)**
- Cumulonimbus
  - Aislados **ISOL CB**
  - Ocasionales **OCNL CB**
  - Frecuentes **FRQ CB**
- Cúmulos en forma de torre
  - Aislados **ISOL TCU**
  - Ocasionales **OCNL TCU**
  - Frecuentes **FRQ TCU**

**Engelamiento**

- Engelamiento moderado (excepto engelamiento en nubes convectivas) **MOD ICE**

**Turbulencia**

- Turbulencia moderada (excepto turbulencia en nubes convectivas) **MOD TURB**

**Onda orográfica**

- Onda orográfica moderada **MOD MTW**

**NOTAS:**

1. Se pondrá un solo fenómeno significativo, de la lista anterior, en cada AIRMET
2. **Fenómenos meteorológicos que causan reducción de la visibilidad:** DZ (llovizna), RA (lluvia), SN (nieve), SG (cinarra), PL (hielo granulado), GR (granizo,  $d \geq 5$  mm), GS (granizo pequeño o nieve granulada,  $d < 5$  mm), FG (niebla, si visibilidad  $< 1000$  m), BR (neblina, si  $1000 \leq$  visibilidad  $< 5000$  m), SA (arena), DU (polvo), HZ (calima), FU (humo), VA (ceniza volcánica), PO (remolinos de polvo o arena), SQ (turbonada), FC (nubes de embudo: tornado o tromba marina), DS (tempestad de polvo) o SS (tempestad de arena).

Figura 6. Lista de fenómenos en mensajes tipo AIRMET, extraído de la Guía AIRMET de la AEMet

#### 4.1.9 Consideraciones adicionales

Ténganse en consideración las siguientes definiciones:

**Información SIGMET:** «Información expedida por una Oficina de Vigilancia Meteorológica, relativa a la existencia real o prevista de fenómenos meteorológicos en ruta especificados, que puedan afectar a la seguridad de las operaciones de aeronaves.»<sup>9</sup>

**Información AIRMET:** «Información que expide una Oficina de Vigilancia Meteorológica respecto a la presencia real o prevista de determinados fenómenos meteorológicos en ruta que pueden afectar a la seguridad de los vuelos de baja altura (por debajo del FL150), y que no estaba incluida en el pronóstico expedido para los vuelos a baja altura en la región de información de vuelo que se trate o en una subzona de la misma.»

Las **Oficinas de Vigilancia Meteorológica** serán las encargadas de preparar y difundir la información SIGMET y AIRMET, en territorio nacional a saber: OVM de Valencia (LEVA, encargada del FIR/UIR de Madrid y Barcelona) y OVM de Las Palmas de Gran Canaria (GCGC, encargada del FIR/UIR de Canarias).

Los mensajes tipo SIGMET no tienen límites verticales al **espacio de responsabilidad**, mientras que los mensajes tipo AIRMET tienen espacio de responsabilidad desde el nivel del mar hasta el nivel de vuelo 150.

En España, las regiones de información de vuelo (FIR) abarcan desde tierra hasta el nivel de vuelo 245, y las regiones superiores de información de vuelo (UIR) abarcan desde el nivel de vuelo 245 hacia arriba.

---

<sup>9</sup> Capítulo 1. Anexo 3 de la OACI.

## 4.2 IMPLEMENTACIÓN DEL SISTEMA AUTOMÁTICO DE DESCODIFICACIÓN

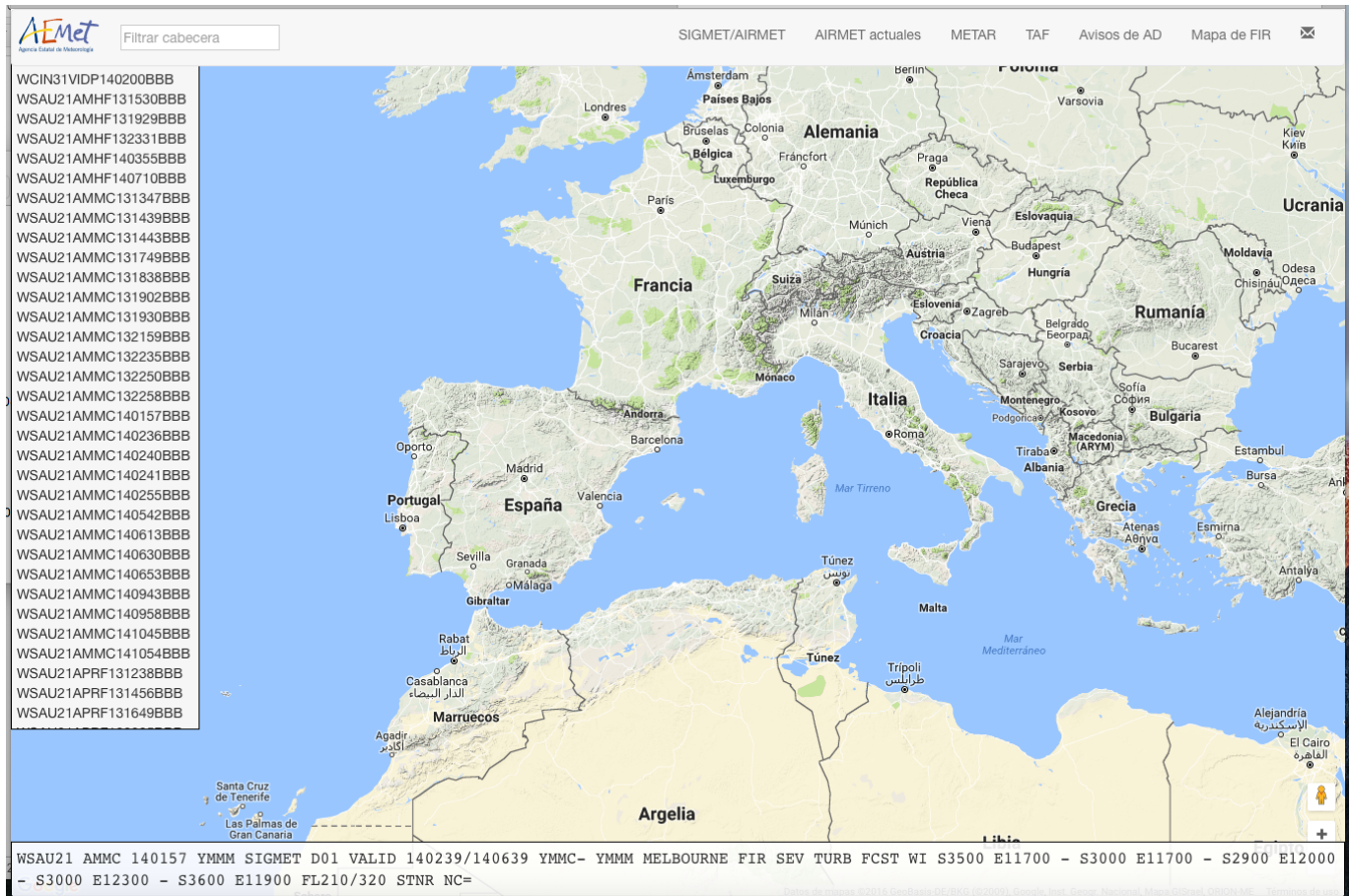


Figura 7. Página web correspondiente a la decodificación automática de mensajes tipo SIGMET

Tras conseguirse el objetivo de la decodificación de mensajes tipo SIGMET y AIRMET, se planteó la decodificación automática con los SIGMET y AIRMET vigentes (contenidos en una carpeta propia de la página *abweb*<sup>10</sup> de la *intranet* de la propia AEMet. Posteriormente, aprovechando el trabajo realizado anteriormente por la Agencia en materia de decodificación de METAR, TAF y Avisos de Aeródromo, también se planteó la decodificación automática con los mensajes anteriormente citados del territorio nacional representados en un mapa.

De esta manera, se ideó un sistema para que en esa carpeta aparecieran los SIGMET, AIRMET, METAR, TAF y Avisos de Aeródromo cada uno en una carpeta, a saber: *Sigmat*, *Airmet*, *MetarTaf* y *AvisosAero* en la cual se contienen todos los mensajes de cada tipo en ficheros de texto cuyo nombre contiene la cabecera (mensajes tipo SIGMET y AIRMET) o el identificador OACI del aeródromo (mensajes tipo METAR, TAF y avisos de aeródromo) y la extensión .txt excepto para los mensajes tipo METAR y TAF, cuya extensión sería .M para METAR y .T para TAF (e.g. GCFV.M y GCFV.T).

<sup>10</sup> <http://abweb.aemet.es> únicamente accesible desde la *intranet* de AEMet.

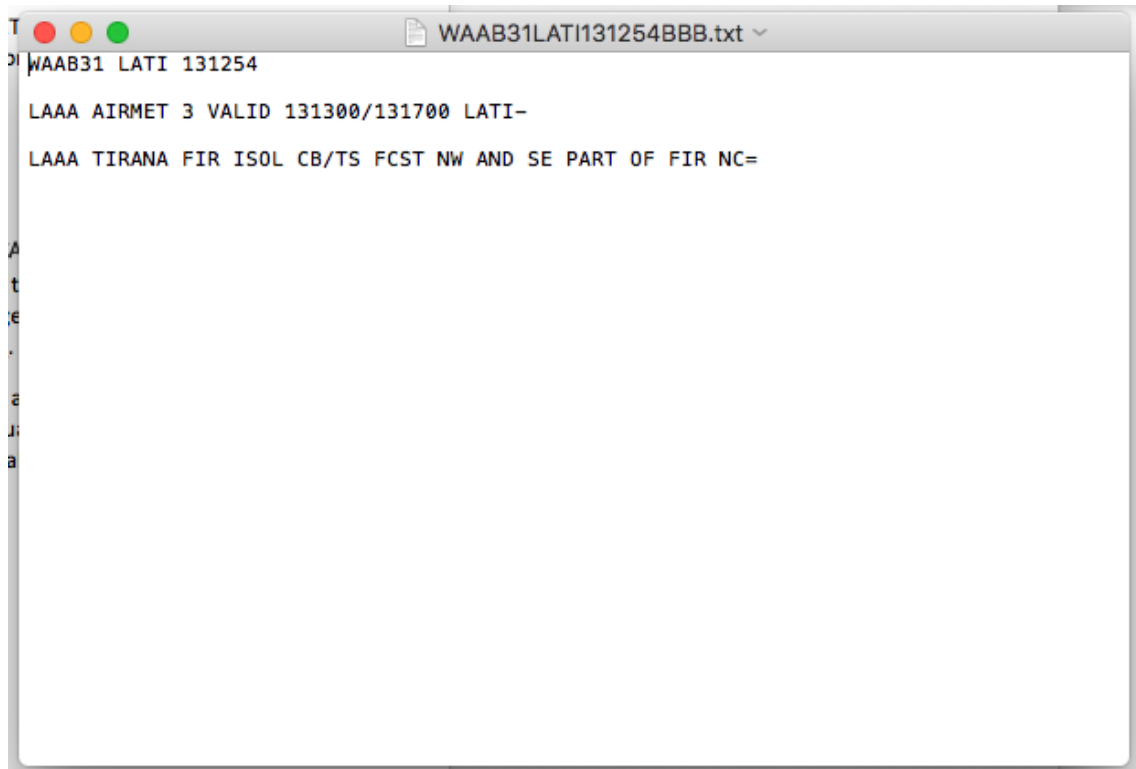


Figura 8. Ejemplo de fichero cuyo contenido es el propio mensaje, en este caso se trata de un mensaje tipo AIRMET

A través de Javascript no es posible acceder a un fichero de este tipo directamente, por lo que se hubo de recurrir a otra alternativa para tener acceso desde la propia página web al contenido de los ficheros correspondientes.

#### 4.2.1 Implementación de mensajes tipo SIGMET

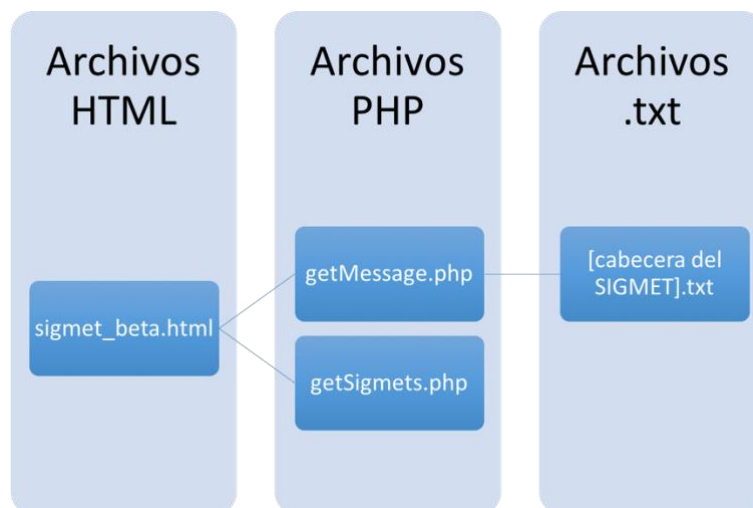


Figura 9. Esquema correspondiente a la obtención del listado de SIGMET y el contenido de un determinado mensaje

Se planteó como opción interesante interesante mostrarse una lista con las cabeceras de los mensajes y la posibilidad de filtrar el texto de la cabecera para el caso de SIGMET y AIRMET. Para ello se debía obtener una lista en tiempo real de los ficheros nombrados con las cabeceras de los mensajes correspondientes.

Por no ser posible realizarlo directamente con Javascript, se recurrió al lenguaje de programación en servidor PHP, creando un fichero cuyo único cometido era devolver una lista cuyo contenido eran los nombres de fichero con unas características determinadas, a saber en este caso: poseer extensión .txt y pertenecer a la carpeta *Sigmat*.

Además, se aprovecharía el mismo fichero PHP para filtrar con la cabecera suministrada si la hubiere a través de una petición del tipo GET mediante HTTP. Se optó por una petición tipo GET por su simplicidad de uso y por no transferir información comprometida en la propia petición.

Véase el proceso en el código del fichero *getSigmets.php*:

```
<?php
    $cabecera = $_GET['cabecera'];
    $mensajes = array();
    foreach (glob("aero/Sigmat/*.txt") as $file) {
        if(preg_match('/'.$cabecera.'/', $file)) {
            // Valid match
            if(filesize($file)!=0) {
                array_push($mensajes, str_replace("aero/Sigmat/", "",
$file));
            }
        }
    }
    echo json_encode($mensajes);
?>
```

Nótese que lo que se devuelve es un objeto de tipo JSON que posteriormente podrá ser utilizado por el código Javascript de la página principal cuando obtenga la calificación de *success* en el proceso.

Por darse frecuentemente casos de ficheros incorrectamente alojados en el servidor de la AEMet, se comprueba que el tamaño del fichero sea diferente a cero y evitar así listar dichos ficheros en aras de no confundir al usuario.

Una vez obtenido el objeto JSON, se muestra la lista de forma automática con todos los mensajes tipo SIGMET que tiene el servidor actualmente.

Al poner el usuario el ratón encima de cualquier elemento de lista (cabecera del mensaje), se realiza otra petición tipo GET en la cual se especifica la cabecera seleccionada al fichero *getMessage.php* que, previa comprobación de que el fichero en efecto exista, devuelve como cadena de texto el contenido del fichero.

El fichero *getMessage.php* se diseñó como un proceso universal para todos los tipos de mensajes con el fin de economizar código y tamaño de la aplicación. Véase su contenido en las líneas subsiguientes:

```
<?php
    $tipo = $_GET['tipo'];
    $icao = $_GET['icao'];
    switch ($tipo) {
```

```

    case 'metar':
        $path = './aero/MetarTaf/'.$icao.'.M';
        break;
    case 'taf':
        $path = './aero/MetarTaf/'.$icao.'.T';
        break;
    case 'aviso':
        $path =
'./aero/AvisosAero/WWSP60'.$icao.'.txt';
        break;
    case 'sigmet':
        $reference = $_GET['reference'];
        $path = './aero/Sigmet/'.$reference.'.txt';
        break;
    case 'airmet':
        $reference = $_GET['reference'];
        $path = './aero/Airmet/'.$reference.'.txt';
        break;
    default:
        # code...
        break;
}
if (file_exists($path)) {
    echo file_get_contents($path);
} else {
    echo false;
}
?>

```

Nótese en el código que se obtienen dos parámetros con el método GET, a saber: tipo de mensaje e identificador de aeródromo o cabecera del mensaje dependiendo del tipo de mensaje. En el caso de los mensajes tipo SIGMET, el parámetro *reference* contiene la cabecera del mensaje que coincide con el nombre del fichero cuyo contenido se requiere.

Tras obtener el contenido del fichero satisfactoriamente, el propio código de la página web muestra al usuario un elemento visual en la parte inferior con el contenido del mensaje sobre el cual tiene el ratón encima.

Haciendo *click* en el mensaje el contenido se decodifica y se muestra el fenómeno visualmente como se realizaba en la decodificación *per se*.



#### 4.2.2 Implementación de mensajes tipo AIRMET

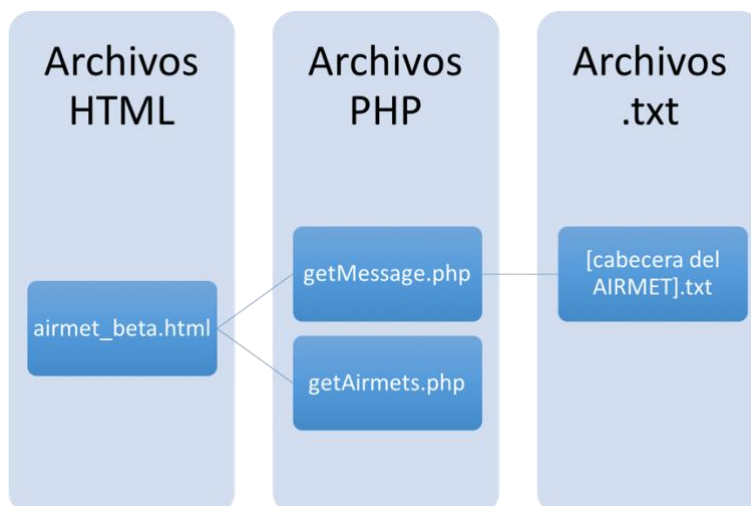


Figura 10. Esquema correspondiente a la obtención del listado de AIRMET y el contenido de un determinado mensaje

Por tratarse de un caso similar al de los mensajes tipo SIGMET, se usó la misma estructura de proceso con la salvedad del cambio del nombre de carpeta, a saber: *Airmet*.

Para mayor claridad, véase el fichero siguiente, *getAirmets.php*.

```

<?php
    $cabecera = $_GET['cabecera'];
    $mensajes = array();
    foreach (glob("aero/Airmet/*.txt") as $file) {
        if(preg_match('/'.$cabecera.'/', $file)) {
            // Valid match
            if(filesize($file)!=0) {
                array_push($mensajes, str_replace("aero/Airmet/", "",
    $file));
            }
        }
    }
    echo json_encode($mensajes);
?>
  
```

#### 4.2.3 Implementación de mensajes tipo METAR

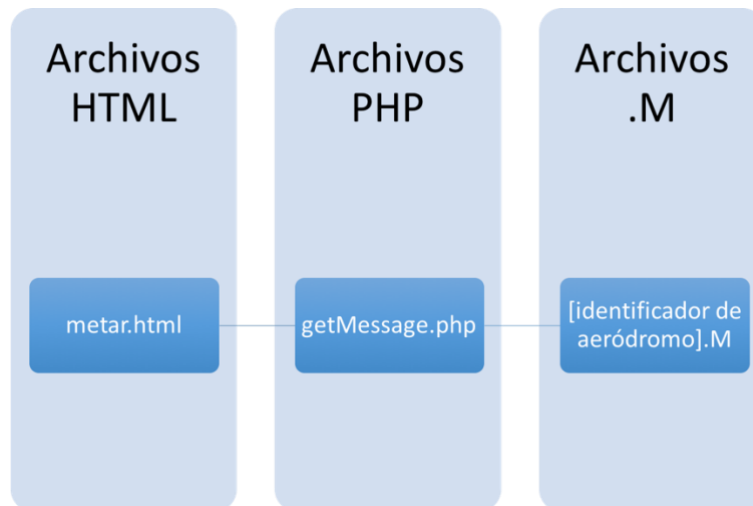


Figura 11. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo METAR

Por contemplarse en el alcance del proyecto la visualización de METAR, TAF y Avisos de Aeródromo únicamente pertenecientes al territorio nacional se utilizó una base de datos con los aeródromos españoles más importantes con las siguientes entradas: nombre, identificador OACI y latitud y longitud en la cual se encuentra situado el aeródromo.

Tratándose de un proceso más simple que el de implementación SIGMET y AIRMET por siempre disponer del mismo número de ficheros con los mismos nombres, únicamente era necesario un proceso en el servidor cuyo cometido era devolver el contenido de los ficheros .M almacenados en la carpeta *MetarTaf* realizado por el fichero *getMessage* explicado anteriormente. Los ficheros se reemplazan por parte de la propia Agencia pertinentemente, por lo que cada vez que el usuario accede a la página web tiene a su disposición la información más reciente.

#### 4.2.4 Implementación de mensajes tipo TAF

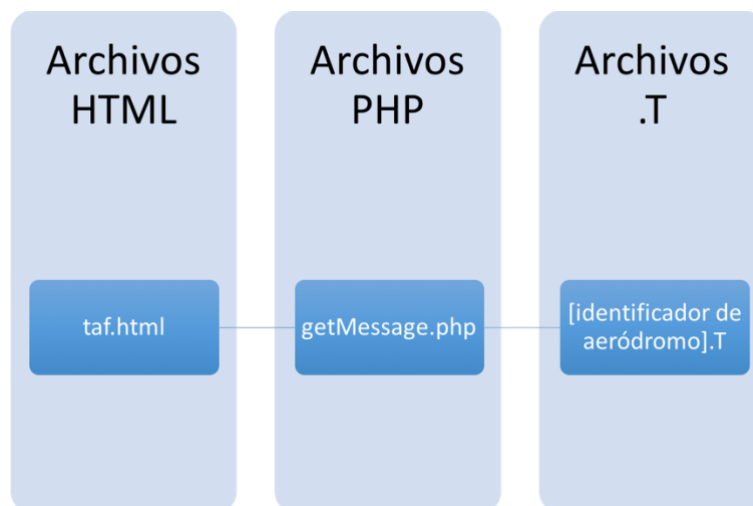


Figura 12. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo TAF

Se trata de un proceso similar al de obtención de mensajes al de los mensajes tipo METAR, con la salvedad de que la extensión del archivo es .T.

#### 4.2.5 Implementación de mensajes tipo Avisos de Aeródromo

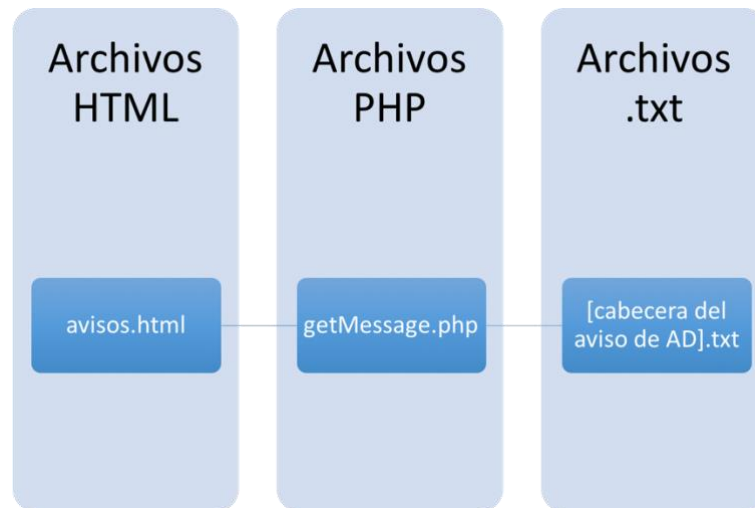


Figura 13. Esquema correspondiente a la obtención del contenido de un determinado mensaje tipo Aviso de Aeródromo

Se trata de un proceso similar al de obtención de mensajes al de los mensajes tipo METAR y TAF, con la salvedad de que la extensión del archivo es *.txt*.

Nótese que los avisos de aeródromo requieren un tratamiento especial por ser mensajes no emitidos periódicamente como los mensajes tipo METAR y TAF. No obstante, no es necesario un proceso secundario como en el caso de SIGMET y AIRMET para devolver una lista de ficheros presentes en un directorio, dado que al realizar el proceso *getMessage* el servidor devuelve *false* si no se encuentra un fichero con el identificador OACI suministrado.

Así pues, únicamente se mostrarán los Avisos de Aeródromo vigentes (vigencia comprobada previamente por los procesos establecidos en la propia Agencia) de los aeródromos en el territorio nacional.

## 5 RESULTADOS

Tras la descripción detallada del proceso y la metodología usada para el desarrollo, se realiza una labor de testeo del software desarrollado a través de pruebas con muestras de cincuenta a cien mensajes por cada tipo de los mismos llegando a resultados satisfactorios en todos los ámbitos: tiempos de carga, latencia y precisión en la decodificación.

Recursos	Todo	Dominio	Tipo	Método	Esquema	Estado	Guard...	Tamaño	Transferido	Inicio	Latencia	Duración
getMessage.php		localhost	XHR	GET	HTTP	200	No	213 B	435 B	3.99 s	2.685 ms	0.091 ms
getMessage.php		localhost	XHR	GET	HTTP	200	No	213 B	435 B	4.54 s	2.846 ms	0.100 ms
coordenadas_oficiales.json		localhost	XHR	GET	HTTP	200	No	10.53 MB	10.53 MB	4.57 s	44.64ms	0.080 ms
ViewportInfoService.GetViewportInfo		maps.googleapi...	Script	GET	HTTPS	200	No	33.71 KB	6.13 KB	4.79 s	127.9ms	0.110 ms
gen_204		maps.googleapi...	Imagen	GET	HTTPS	204	No	—	371 B	4.79 s	127.2ms	0.062 ms
tormentas.png		localhost	Imagen	GET	HTTP	200	No	19.48 KB	19.77 KB	5.08 s	47.79ms	0.295 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	178 B	611 B	5.10 s	26.29ms	0.148 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	1.29 KB	1.71 KB	5.10 s	54.14ms	0.196 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	2.47 KB	2.89 KB	5.10 s	24.70ms	0.237 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	341 B	775 B	5.10 s	23.91ms	0.331 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	9.56 KB	9.97 KB	5.10 s	63.09ms	1.016 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	755 B	1.16 KB	5.10 s	22.17ms	0.220 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	1.51 KB	1.94 KB	5.10 s	21.54ms	0.219 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	13.61 KB	14.02 KB	5.10 s	64.92ms	1.403 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	2.57 KB	3.00 KB	5.10 s	20.28ms	0.138 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	178 B	613 B	5.11 s	19.56ms	0.193 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	11.40 KB	11.82 KB	5.11 s	85.02ms	0.238 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	181 B	615 B	5.11 s	18.00ms	0.270 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	260 B	682 B	5.11 s	49.22ms	0.360 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	15.29 KB	15.71 KB	5.11 s	82.96ms	0.249 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	178 B	612 B	5.11 s	15.63ms	0.134 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	518 B	940 B	5.11 s	47.39ms	0.460 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	17.90 KB	18.32 KB	5.11 s	82.39ms	0.225 ms
vt		maps.googleapi...	Imagen	GET	HTTPS	200	No	3.42 KB	3.85 KB	5.11 s	12.13ms	18.31ms
vt		maps.googleapi...	Script	GET	HTTPS	200	No	988 B	668 B	5.13 s	64.93ms	0.113 ms

Figura 14. Consola web en la que se miden los diferentes parámetros en la ejecución de la herramienta

En las páginas subsecuentes se muestran ejemplos de decodificación para los diferentes tipos de mensajes:

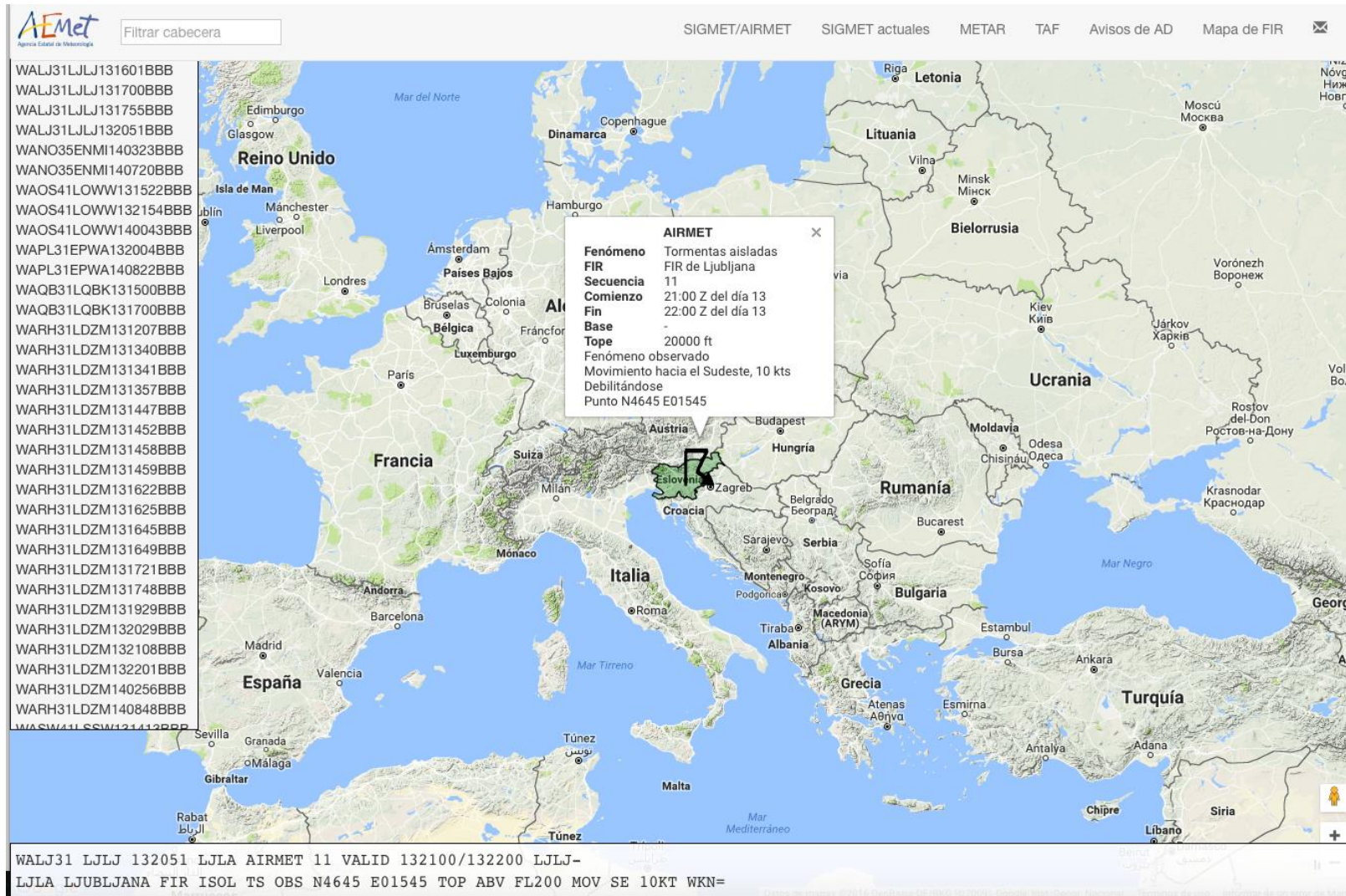


Figura 15. Ejemplo de AIRMET con localización dada a través de un punto

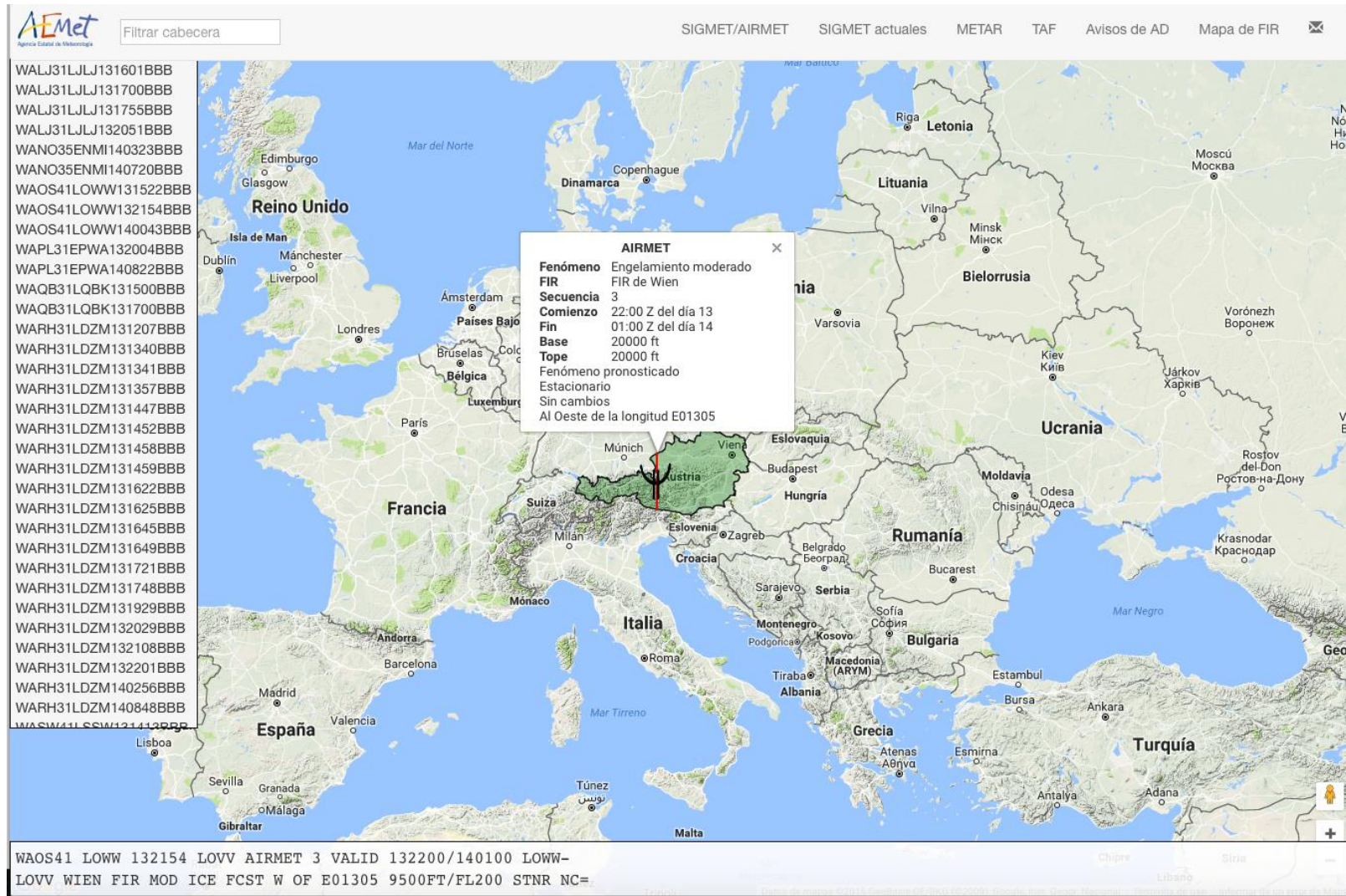


Figura 16. Ejemplo de AIRMET con localización dada a través de una línea de longitud y una dirección

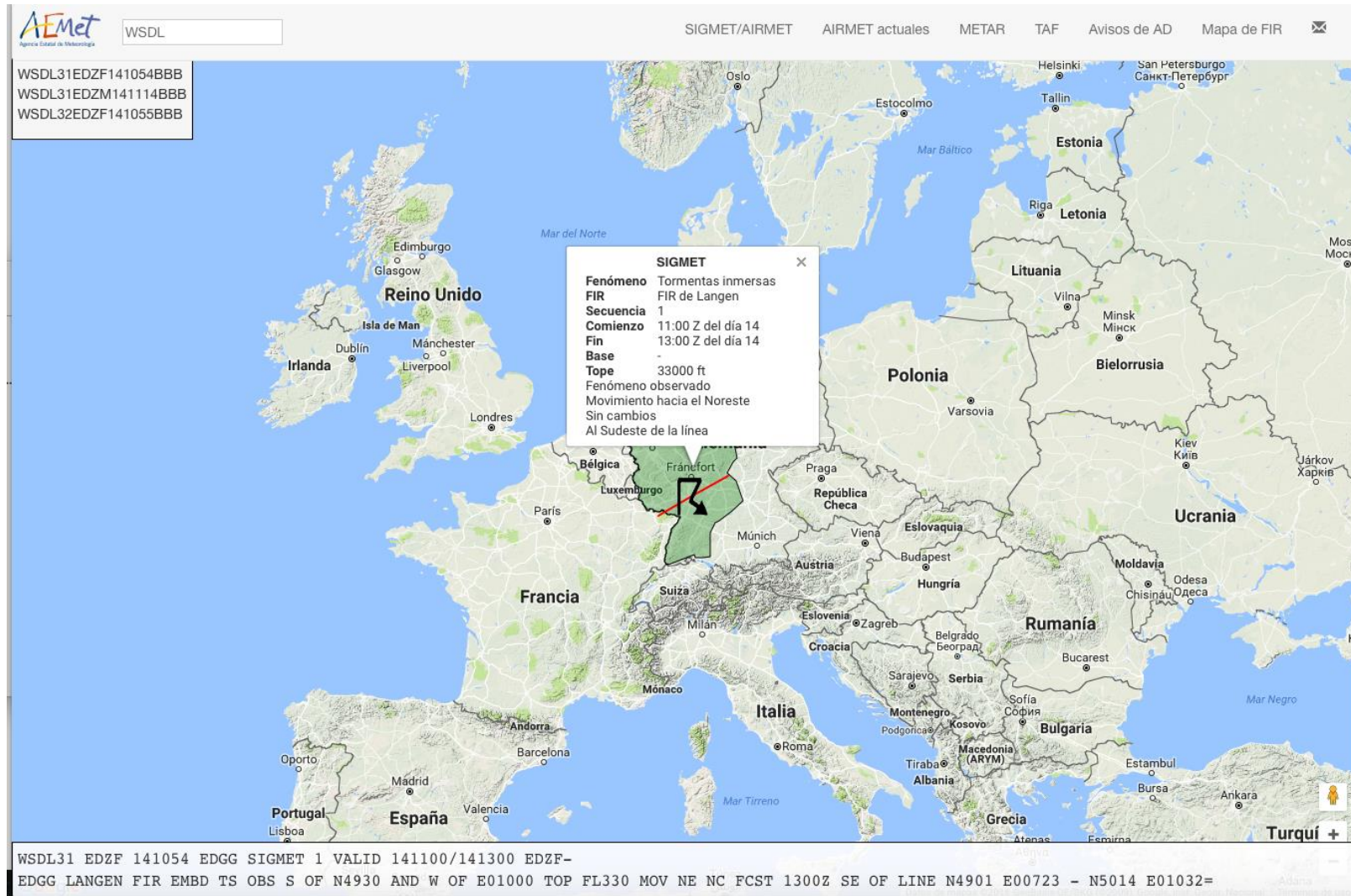


Figura 17. Ejemplo de SIGMET con localización dada a través de una línea por dos puntos y una dirección

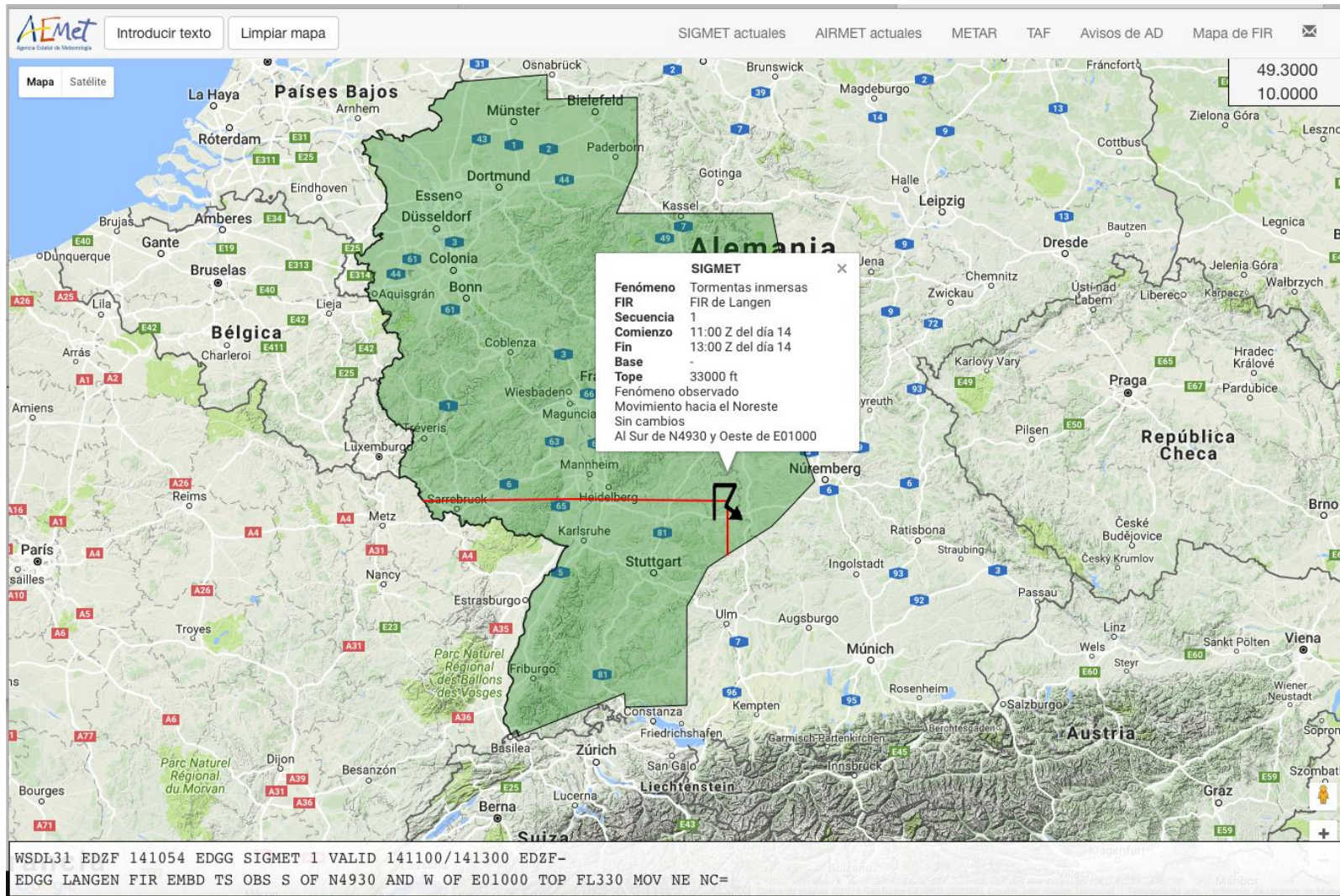


Figura 18. Ejemplo de SIGMET con localización dada a través de direcciones respecto a dos líneas de latitud y longitud



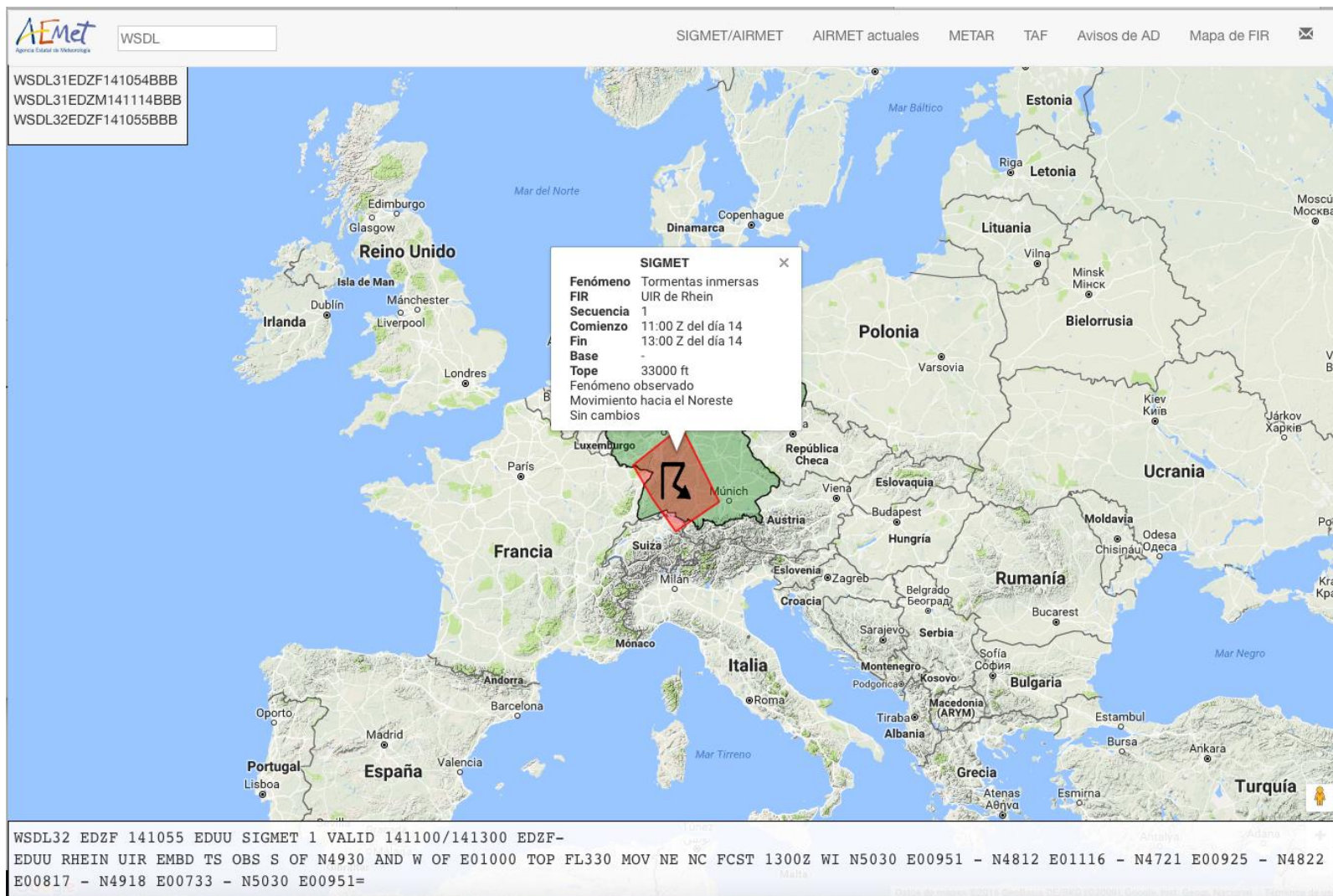


Figura 19. Ejemplo de SIGMET con localización dada a través de un polígono

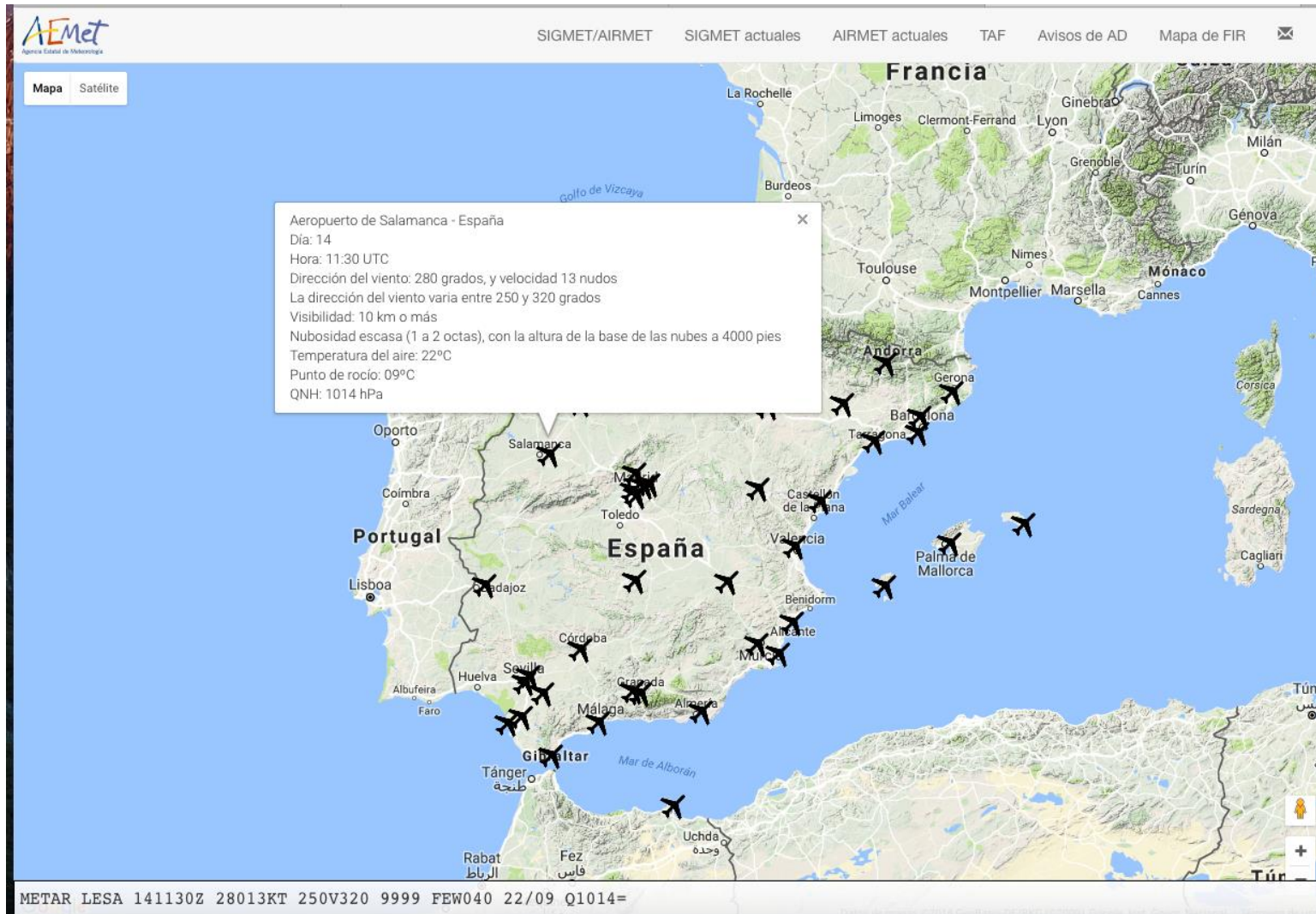


Figura 20. Ejemplo de METAR

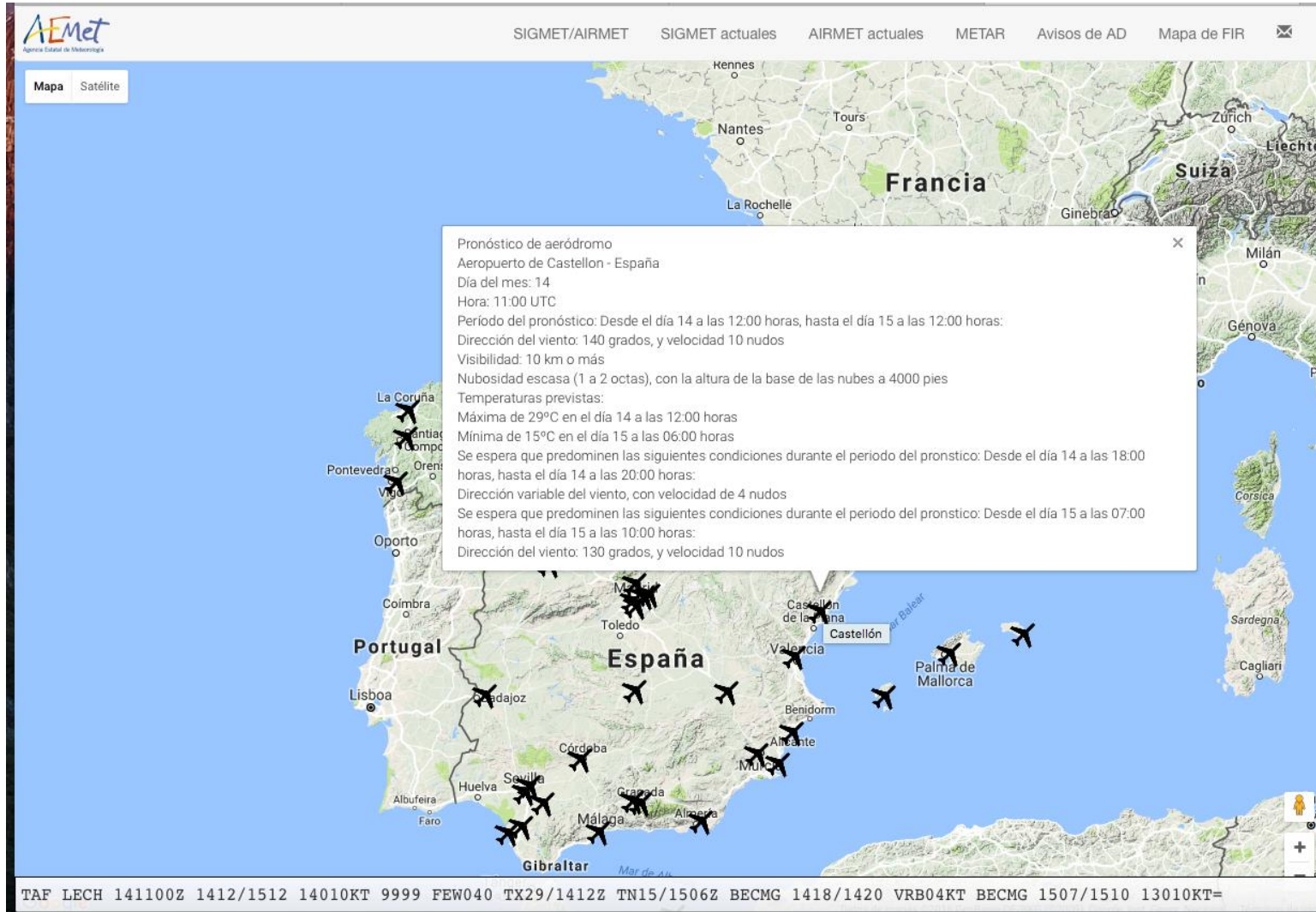


Figura 21. Ejemplo de TAF

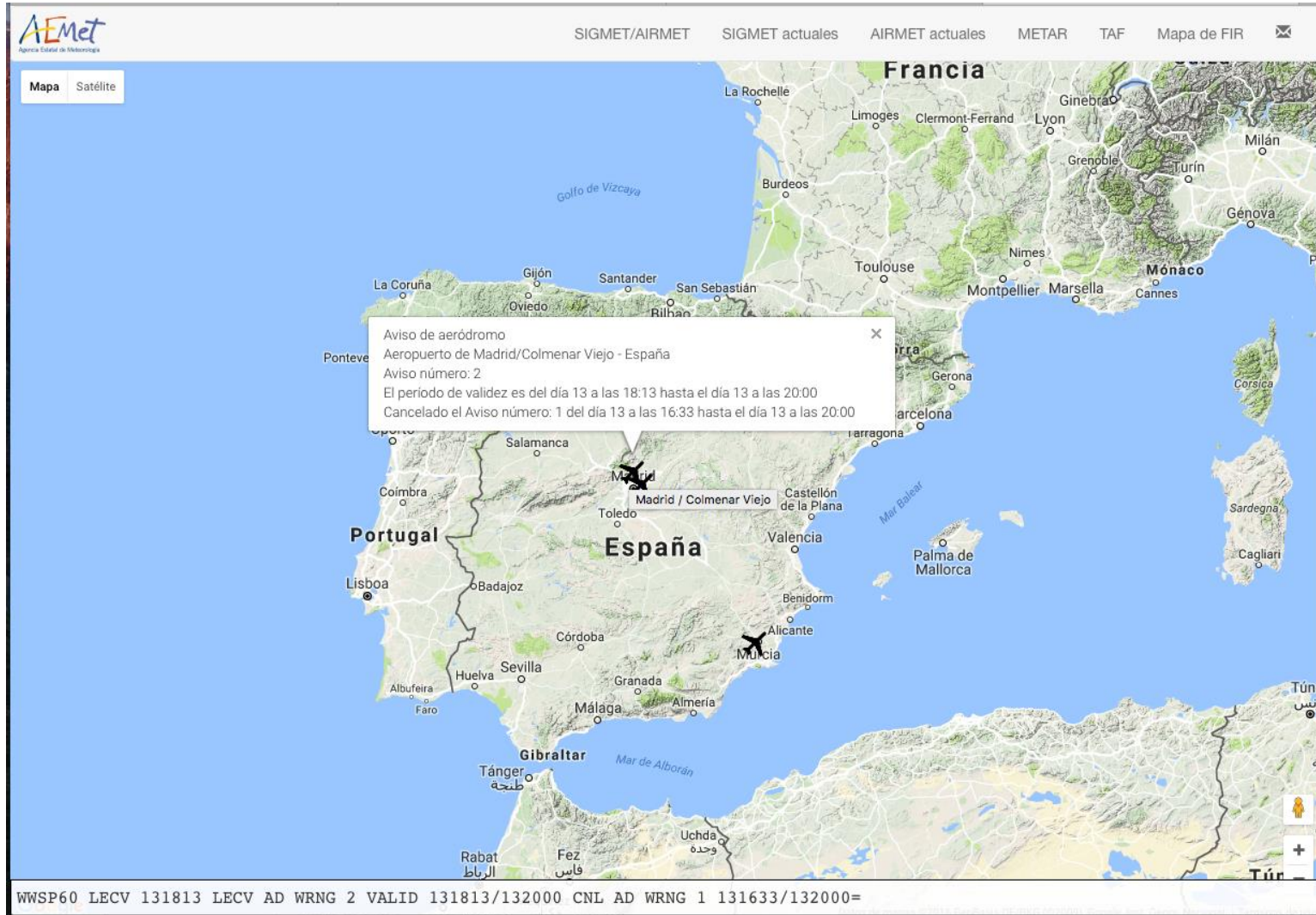


Figura 22. Ejemplo de Aviso de Aeródromo

Es de especial atención la utilidad de la presentación gráfica de los mensajes (tal como se muestra en las figuras anteriores) por suministrarse a través de coordenadas mediante latitud y longitud.

Nótese la complejidad de imaginar espacialmente un polígono dado por n puntos, a modo de ejemplo, y la inmediatez de observarlo representado gráficamente en un mapa interactivo que permite incluso moverlo y hacer zoom (siguiendo las características propias de Google Maps).

Se permite de igual manera la visualización inmediata de los fenómenos a causa de los cuales los mensajes han sido emitidos a través de la simbología definida en la Figura 1 en el caso de SIGMET y AIRMET.

Con respecto a la utilización de la herramienta, se señalan también las cancelaciones de mensajes tipo SIGMET y AIRMET mediante una ventana de información tipo *alert*.

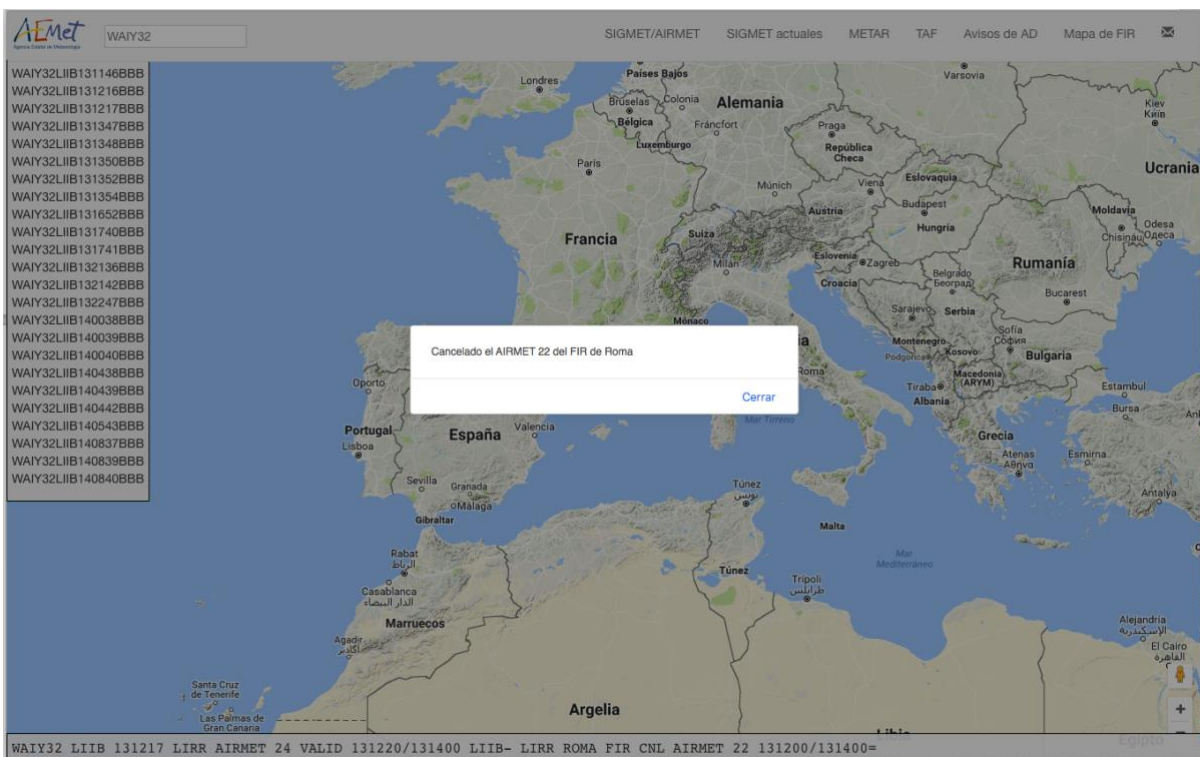


Figura 23. Cancelación de un AIRMET

## 6 CONCLUSIONES

---

Tras el desarrollo de la herramienta se ha conseguido tanto mostrar de una forma visual, clara y concisa la información relativa a las diferentes tipologías de mensajes como permitir la lectura de los mensajes a usuarios profesionales (en este caso el público objetivo de forma mayoritaria, i.e. los pilotos privados y de líneas aéreas) como a usuarios legos que deseen conocer el estado de la meteorología para diferentes regiones o aeródromos.

Además, se ha conseguido implementar un sistema para que la información esté actualizada en cooperación con la Agencia, permitiendo al usuario visualizar los últimos mensajes sean del tipo que fueren dentro de los contemplados en el alcance del presente documento.

Esta herramienta se considera útil por reducir el tiempo de lectura de los mensajes y de búsqueda de los mismos a través de cabecera en el caso de SIGMET y AIRMET o directamente de forma gráfica para los mensajes tipo METAR, TAF y Avisos de Aeródromo. Se encuentra útil de igual manera por suministrar el fenómeno directamente de forma gráfica para mensajes SIGMET y AIRMET de acuerdo a la simbología propuesta en la Figura 1.

Para mensajes tipo SIGMET y AIRMET (sobre los cuales se centró originalmente la atención en el desarrollo de la presente herramienta) se opta por seguir siempre un formato del tipo *Característica: Información*, de modo que el usuario habitual tenga siempre la información con el mismo orden e igualmente resaltada. Al final de la ventana de información, en algunos casos, se añade información adicional relevante por si fuera necesaria (como por ejemplo, en el caso de darse la localización a través de un punto, se incluye el punto suministrado).

Finalmente, se ha conseguido implementar esta herramienta dentro del sistema de AEMet para que sus usuarios disfruten de sus posibilidades.

### 6.1 LIMITACIONES

Se encuentran dos tipos de limitaciones, las determinadas por factores propios y las determinadas por factores ajenos:

Entre las primeras, no se dispuso del tiempo necesario para desarrollar los mensajes tipo SIGMET correspondientes a cenizas volcánicas y ciclones tropicales.

Entre las segundas, se encuentran la imposibilidad de la obtención de una base de datos que incluya las coordenadas para la representación de regiones de tráfico controladas (CTA) así como la inutilidad de la herramienta contra mensajes provenientes de ciertas regiones que no siguen con la normativa OACI.

### 6.2 RECOMENDACIONES

Pese haberse testado el código contra numerosos mensajes de todas las tipologías, se recomienda al usuario realizar una rápida comprobación con el propio código del mensaje puesto que existe un sinfín de mensajes emitidos que no se corresponden con la normativa OACI establecida con exactitud pese a encontrarse en países que oficialmente sí la siguen.

### 6.3 TRABAJO FUTURO

Se marcan las siguientes pautas para posibles desarrollos futuros sobre la presente herramienta:

Se podría desarrollar el código (más complejo) de decodificación de mensajes tipo SIGMET correspondientes a cenizas volcánicas y/o ciclones tropicales siguiendo la línea marcada por la decodificación a través de expresiones regulares más elaboradas.

Se podría desarrollar una interfaz para dispositivos móviles y *tablets* para dotar a la herramienta de una mayor versatilidad y poder realizar consultas desde cualquier lugar.

Podría encontrarse una fuente de información diferente a la de la propia Agencia Estatal de Meteorología para los mensajes, tal como *Aviation Weather* de la NOAA que permita obtener los mensajes desde fuera de la *intranet* de la Agencia.

Podría completarse el área válida para mensajes del tipo SIGMET y AIRMET realizando una decodificación alternativa para países que no siguen las directrices de la OACI sino una propia, tales como los Estados Unidos.

## 7 BIBLIOGRAFÍA Y REFERENCIAS

---

Las referencias se han incorporado como notas al pie en cada una de las páginas.

Respecto a la bibliografía utilizada, se incluyen los siguientes documentos:

- Guía MET. Ministerio de Agricultura, Alimentación y Medioambiente, Agencia Estatal de Meteorología y Servicio de Aplicaciones Aeronáuticas. Publicación de 2015.
- Guía de elaboración del SIGMET. Miguel Ángel Pelacho Aja. Publicación del 4 de septiembre de 2014. Agencia Estatal de Meteorología.
- Guía de elaboración del AIRMET. Miguel Ángel Pelacho Aja. Publicación del 4 de septiembre de 2014. Agencia Estatal de Meteorología.
- Anexo 3 al Convenio sobre Aviación Civil Internacional. Servicio meteorológico para la navegación aérea internacional. Partes I y II. Decimoctava edición publicada en julio de 2013. Organización de Aviación Civil Internacional.



## 8 SIGLAS Y ACRÓNIMOS

---

SIGMET	Significant Meteorological Information
AIRMET	Airmen's Meteorological Information
METAR	Meteorological Aerodrome Report
TAF	Terminal Aerodrome Forecast
AEMet	Agencia Estatal de Meteorología
NOAA	<i>National Oceanic and Atmospheric Administration</i>
OACI	Organización de Aviación Civil Internacional (también ICAO por sus siglas en inglés)
SAA	Servicio de Aplicaciones Aeronáuticas (pertenece a AEMet)
OVM	Oficina de Vigilancia Meteorológica (también MWO por sus siglas en inglés)
FIR	<i>Flight Information Region</i>
UIR	<i>Upper Information Region</i>
CTA	<i>Controlled Traffic Area</i>
HTTP	<i>Hypertext Transfer Protocol</i>