

GPS NAVIGATION MESSAGE FILE - HEADER SECTION DESCRIPTION		
HEADER LABEL (Columns 61-80)	DESCRIPTION	FORMAT
RINEX VERSION / TYPE	- Format version (2.10) - File type ('N' for Navigation data)	F9.2,11X, A1,19X
PGM / RUN BY / DATE	- Name of program creating current file - Name of agency creating current file - Date of file creation	A20, A20, A20
* COMMENT	Comment line(s)	A60
* ION ALPHA	Ionosphere parameters A0-A3 of almanac (page 18 of subframe 4)	2X,4D12.4
* ION BETA	Ionosphere parameters B0-B3 of almanac	2X,4D12.4
* DELTA-UTC: A0,A1,T,W	Almanac parameters to compute time in UTC (page 18 of subframe 4) A0,A1: terms of polynomial T : reference time for UTC data W : UTC reference week number. Continuous number, not mod(1024)!	3X,2D19.12, 2I9 *)
* LEAP SECONDS	Delta time due to leap seconds	I6
END OF HEADER	Last record in the header section.	60X

Records marked with * are optional

GPS NAVIGATION MESSAGE FILE - DATA RECORD DESCRIPTION		
OBS. RECORD	DESCRIPTION	FORMAT
PRN / EPOCH / SV CLK	- Satellite PRN number - Epoch: Toc - Time of Clock year (2 digits, padded with 0 if necessary) month day hour minute second - SV clock bias (seconds) - SV clock drift (sec/sec) - SV clock drift rate (sec/sec2)	I2, 1X,I2.2, 1X,I2, 1X,I2, 1X,I2, 1X,I2, F5.1, 3D19.12 *)
BROADCAST ORBIT - 1	- IODE Issue of Data, Ephemeris - Crs (meters) - Delta n (radians/sec) - M0 (radians)	3X,4D19.12
BROADCAST ORBIT - 2	- Cuc (radians) - e Eccentricity - Cus (radians) - sqrt(A) (sqrt(m))	3X,4D19.12
BROADCAST ORBIT - 3	- Toe Time of Ephemeris (sec of GPS week) - Cic (radians)	3X,4D19.12

	- OMEGA (radians)	
	- CIS (radians)	
BROADCAST ORBIT - 4	- i0 (radians) - Crc (meters) - omega (radians) - OMEGA DOT (radians/sec)	3X, 4D19.12
BROADCAST ORBIT - 5	- IDOT (radians/sec) - Codes on L2 channel - GPS Week # (to go with TOE) Continuous number, not mod(1024)! - L2 P data flag	3X, 4D19.12
BROADCAST ORBIT - 6	- SV accuracy (meters) - SV health (bits 17-22 w 3 sf 1) - TGD (seconds) - IODC Issue of Data, Clock	3X, 4D19.12
BROADCAST ORBIT - 7	- Transmission time of message **) (sec of GPS week, derived e.g. from Z-count in Hand Over Word (HOW) - Fit interval (hours) (see ICD-GPS-200, 20.3.4.4) Zero if not known - spare - spare	3X, 4D19.12

**) Adjust the Transmission time of message by -604800 to refer to the reported week, if necessary.

*) In order to account for the various compilers, E,e,D, and d are allowed letters between the fraction and exponent of all floating point numbers in the navigation message files.
Zero-padded two-digit exponents are required, however.

GPS OBSERVATION DATA FILE - HEADER SECTION DESCRIPTION			
HEADER LABEL (Columns 61-80)	DESCRIPTION	FORMAT	
RINEX VERSION / TYPE	- Format version (2.10) - File type ('O' for Observation Data) - Satellite System: blank or 'G': GPS 'R': GLONASS 'S': Geostationary signal payload 'T': NNSS Transit 'M': Mixed	F9.2,11X, A1,19X, A1,19X	
PGM / RUN BY / DATE	- Name of program creating current file - Name of agency creating current file - Date of file creation	A20, A20, A20	
* COMMENT	Comment line(s)	A60	*
MARKER NAME	Name of antenna marker	A60	
* MARKER NUMBER	Number of antenna marker	A20	*
OBSERVER / AGENCY	Name of observer / agency	A20,A40	
REC # / TYPE / VERS	Receiver number, type, and version (Version: e.g. Internal Software Version)	3A20	
ANT # / TYPE	Antenna number and type	2A20	
APPROX POSITION XYZ	Approximate marker position (WGS84)	3F14.4	
ANTENNA: DELTA H/E/N	- Antenna height: Height of bottom surface of antenna above marker - Eccentricities of antenna center relative to marker to the east and north (all units in meters)	3F14.4	
WAVELENGTH FACT L1/2	- Default wavelength factors for L1 and L2 1: Full cycle ambiguities 2: Half cycle ambiguities (squaring) 0 (in L2): Single frequency instrument - zero or blank The default wavelength factor line is required and must precede satellite-specific lines.	2I6, I6	
* WAVELENGTH FACT L1/2	- Wavelength factors for L1 and L2 1: Full cycle ambiguities 2: Half cycle ambiguities (squaring) 0 (in L2): Single frequency instrument - Number of satellites to follow in list for which these factors are valid. - List of PRNs (satellite numbers with system identifier) These optional satellite specific lines may follow, if they identify a state different from the default values. Repeat record if necessary.	2I6, I6, 7(3X,A1,I2)	*
# / TYPES OF OBSERV	- Number of different observation types	I6,	

	stored in the file - Observation types If more than 9 observation types: Use continuation line(s) The following observation types are defined in RINEX Version 2.10: L1, L2: Phase measurements on L1 and L2 C1 : Pseudorange using C/A-Code on L1 P1, P2: Pseudorange using P-Code on L1,L2 D1, D2: Doppler frequency on L1 and L2 T1, T2: Transit Integrated Doppler on 150 (T1) and 400 MHz (T2) S1, S2: Raw signal strengths or SNR values as given by the receiver for the L1,L2 phase observations Observations collected under Antispoofing are converted to "L2" or "P2" and flagged with bit 2 of loss of lock indicator (see Table A2). Units : Phase : full cycles Pseudorange : meters Doppler : Hz Transit : cycles SNR etc : receiver-dependent The sequence of the types in this record has to correspond to the sequence of the observations in the observation records	9(4X,A2) 6X,9(4X,A2)	
* INTERVAL	Observation interval in seconds	F10.3	*
TIME OF FIRST OBS	- Time of first observation record (4-digit-year, month,day,hour,min,sec) - Time system: GPS (=GPS time system) GLO (=UTC time system) Compulsory in mixed GPS/GLONASS files Defaults: GPS for pure GPS files GLO for pure GLONASS files	5I6,F13.7, 5X,A3	
* TIME OF LAST OBS	- Time of last observation record (4-digit-year, month,day,hour,min,sec) - Time system: Same value as in TIME OF FIRST OBS record	5I6,F13.7, 5X,A3	*
* RCV CLOCK OFFS APPL	Epoch, code, and phase are corrected by applying the realtime-derived receiver clock offset: 1=yes, 0=no; default: 0=no Record required if clock offsets are reported in the EPOCH/SAT records	I6	*
* LEAP SECONDS	Number of leap seconds since 6-Jan-1980 Recommended for mixed GPS/GLONASS files	I6	*
* # OF SATELLITES	Number of satellites, for which observations are stored in the file	I6	*
* PRN / # OF OBS	PRN (sat.number), number of observations for each observation type indicated in the "# / TYPES OF OBSERV" - record. If more than 9 observation types: Use continuation line(s) This record is (these records are)	3X,A1,I2,9I6 6X,9I6	*

	repeated for each satellite present in the data file	
END OF HEADER	Last record in the header section.	60X

Records marked with * are optional

GPS OBSERVATION DATA FILE - DATA RECORD DESCRIPTION		
OBS. RECORD	DESCRIPTION	FORMAT
EPOCH/SAT or EVENT FLAG	- Epoch : - year (2 digits, padded with 0 if necessary) - month, day, hour, min, - sec - Epoch flag 0: OK 1: power failure between previous and current epoch >1: Event flag - Number of satellites in current epoch - List of PRNs (sat.numbers with system identifier, see 5.1) in current epoch - receiver clock offset (seconds, optional) If more than 12 satellites: Use continuation line(s) If epoch flag 2-5: - Event flag: 2: start moving antenna 3: new site occupation (end of kinem. data) (at least MARKER NAME record follows) 4: header information follows 5: external event (epoch is significant, same time frame as observation time tags) - "Number of satellites" contains number of special records to follow. Maximum number of records: 999 - For events without significant epoch the epoch fields can be left blank If epoch flag = 6: 6: cycle slip records follow to optionally report detected and repaired cycle slips (same format as OBSERVATIONS records; slip instead of observation; LLI and signal strength blank or zero)	1X, I2.2, 4(1X, I2), F11.7, 2X, I1, I3, 12(A1, I2), F12.9 32X, 12(A1, I2) [2X, I1,] [I3]
OBSERVATIONS	- Observation - LLI - Signal strength If more than 5 observation types (=80 char): continue observations in next record. This record is (these records are) repeated for each satellite given in EPOCH/SAT - record. Observations: Phase : Units in whole cycles of carrier Code : Units in meters	rep. within record for each obs.type (same seq as given in header) m(F14.3, I1, I1)

Missing observations are written as 0.0 or blanks.

Phase values overflowing the fixed format F14.3 have to be clipped into the valid interval (e.g. add or subtract 10^9), set LLI indicator.

Loss of lock indicator (LLI). Range: 0-7

0 or blank: OK or not known

Bit 0 set : Lost lock between previous and current observation: cycle slip possible

Bit 1 set : Opposite wavelength factor to the one defined for the satellite by a previous WAVELENGTH FACT L1/2 line. Valid for the current epoch only.

Bit 2 set : Observation under Antispoofing (may suffer from increased noise)

Bits 0 and 1 for phase only.

Signal strength projected into interval 1-9:

1: minimum possible signal strength

5: threshold for good S/N ratio

9: maximum possible signal strength

0 or blank: not known, don't care

```
function mainWEB

clear all; close all;

datos= load ( '/home/web/www/ana00/objetos/datos.m' );
lat= datos(1);
lng= datos(2);
alt= datos(3);
tip=datos(4);

sett.navFile = '/home/web/www/ana00/objetos/file.08n';
sett.obsfile='/home/web/www/ana00/objetos/file.08o';
flag.loadXMLionpar = 0;

a_info.refLocation.llh(1) = lat/180*pi;
a_info.refLocation.llh(2) = lng/180*pi;
a_info.refLocation.llh(3) = alt;

flag.plot = 0;

a_info.refLocation.xyz = llh2xyz(a_info.refLocation.llh);

null_position = [0 0 0];
null_TOW = 0;

if tip== 2
    [acq, sett.numSnapShots]=read_uab_arc(sett.obsfile);

else
    [acq, sett.numSnapShots]=read_rin_ob(sett.obsfile);
end

a_info.TOW = acq(1).SV(1).TOW;
resllh = zeros(sett.numSnapShots,2);

fid= fopen('/home/web/www/ana00/objetos/data.xml', 'w');
fprintf (fid, '<markers>\n');
for nSnapShot=1:sett.numSnapShots,
    acq(nSnapShot).TOW = a_info.TOW;
    if( acq(nSnapShot).Nsv > 4 )
        [acq(nSnapShot).xyzPos,acq(nSnapShot).PosError,acq(nSnapShot).estTOW,
            TruePosition,acq(nSnapShot).SVpos,a_info] =
            aPosPet(a_info,acq(nSnapShot), flag, acq(nSnapShot).TOW,
                a_info.refLocation.xyz, sett.navFile);
```

```
PositionLLH      = xyz2llh(acq(nSnapShot).xyzPos);
TruePositionLLH = xyz2llh(TruePosition);
resllh(nSnapShot,:) = [PositionLLH(1)/pi*180 PositionLLH(2)/pi*180];

else
    acq(nSnapShot).xyzPos = null_position;
    acq(nSnapShot).estTOW = null_TOW;
end;
fprintf(fid, '<marker lat="%f" lng="%f"/>\n', resllh(nSnapShot,1),
resllh(nSnapShot,2));
end;
fprintf(fid, '</markers>');
fclose(fid);

exit;
```

```
function [acq,numSnapShots] = read_rin_ob(obsfile)

tmp=[];

fide=fopen(obsfile, 'r');

while 1
    line = fgetl(fide);
    answer = findstr(line,'END OF HEADER');
    if ~isempty(answer), break; end;
end;

fide2=fopen(obsfile);
b=0;
line2 = fgetl(fide2);
while line2 ~= -1
    answer = findstr(line2,'G');
    if ~isempty(answer), b=b+1;
    end;
    line2 = fgetl(fide2);
end

sett.numSnapShots      =b;
sett.Nsats              =15;
SV(1:sett.Nsats)       =
struct('CN0',0,'Fe',0,'pseudoRange',0,'PRN',0);
acq(1:sett.numSnapShots) =
struct('SV',SV,'Nsv',0,'estTOW',0,'PosError',0,'TOW',0,'xyzPos',[0 0 0]);

linein=0;
linein=fgetl(fide);
%empiezo a leer fichero
%fijo tow inicial
    anio=str2num(linein(1:3));
    mes=str2num(linein(4:6));
    dia=str2num(linein(7:9));
    hora=str2num(linein(10:12));
    min=str2num(linein(13:15));
    seg=str2num(linein(16:26));
    tmp = ymdhms2gps(anio+2000,mes,dia,hora,min,seg);
    TOWinic=tmp(2);
k=0;
while linein ~= -1
    k=k+1;
    nsat=str2num(linein(30:32));

    satelites=[];
    satelites(1)=str2num(linein(34:35));
    t=37;
    for(n=1:nsat-1)
```

```
        satellites(n+1)=str2num(linein(t:t+1));
        t=t+3;
    end;

    for(i=1:nsat)
        acq(k).SV(i).TOW= TOWinic;
        acq(k).SV(i).PRN= satellites(i);
        acq(k).SV(i).CN0=1;
        acq(k).SV(i).OK=1;
        acq(k).Nsv=nsat;

        linein = fgetl(fide);
        acq(k).SV(i).Fe= str2num(linein(36:46));
        acq(k).SV(i).pseudoRange= str2num(linein(1:14));

    end
    linein=fgetl(fide);

end
fide = fopen(obsfile);
numSnapShots      =b;
end
```

```
function [acq,numSnapShots] = read_uab_arc(obsfile)
DATA= load(obsfile);
t= DATA(end,1);
s=size(DATA);
a1=[1:t];
a2= ones(s(1),1);
A= kron(a1,a2);

b1=DATA(:,1);
b2= ones(1,t);
B= kron(b1,b2);

S= sum(A==B);
sett.numSnapShots =t;
sett.Nsats= max(S);

SV(1:sett.Nsats) =
struct('CN0',0,'Fe',0,'pseudoRange',0,'PRN',0);
acq(1:sett.numSnapShots) =
struct('SV',SV,'Nsv',0,'estTOW',0,'PosError',0,'TOW',0,'xyzPos',[0 0 0]);

n=1;
for(k=1:t)
    for(i=1:sett.Nsats)
        acq(k).SV(i).TOW= DATA(n,2);
        acq(k).SV(i).PRN= DATA(n,4);
        acq(k).SV(i).Fe= DATA(n,6);
        acq(k).SV(i).pseudoRange= DATA(n,5);
        acq(k).SV(i).CN0=1;
        acq(k).SV(i).estTOW= DATA(n,10);
        acq(k).SV(i).OK=1;
        acq(k).Nsv=sett.Nsats;
        n=n+1;
    end
    numSnapShots=t;
end
```

```
1:  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2:
3:  <html>
4:  <head>
5:  <title>Formulario</title>
6:  </head>
7:
8:
9:  <BODY background="objetos/brujula.png">
10:  <form method="POST" enctype="multipart/form-data" action="recoger.php">
11:  <table width="396" align="left">
12:  <tr>
13:
14:  <td colspan="2"><div align="left"><span class="titulo"><font size="2"
15:  face="Arial Black"> REFERENCE POSITION:</font></span></div></td>
16:  </tr>
17:  <tr></tr>
18:  <tr></tr>
19:  <tr></tr>
20:  <tr></tr>
21:  <tr></tr>
22:  <tr>
23:  <td width="121">
24:  <div align="left"><font size="2" face="Arial Black" color= "blue">
25:  Latitude:</font></div>
26:  </div></td>
27:  <td width="263"><div align="left">
28:  <input name="latitude" type="text" id="latitude" />
29:  </div></td>
30:  </tr>
31:  <tr>
32:  <td>
33:  <div align="left"><font size="2" face="Arial Black" color= "blue">
34:  Longitude:</font></div>
35:  </div></td>
36:  <td><div align="left">
37:  <input name="longitude" type="text" id="longitude" />
38:  </div></td>
39:  </tr>
40:  <tr>
41:  <td>
42:  <div align="left"><font size="2" face="Arial Black" color= "blue">
43:  Height:</font></div>
44:  </div></td>
45:  <td><div align="left">
46:  <input name="height" type="text" id="height" />
47:  </div></td>
48:  </tr>
49:  <tr></tr>
50:  <tr></tr>
51:  <tr></tr>
52:  <tr></tr>
53:  <tr></tr>
54:  <tr></tr>
55:  <tr></tr>
56:  <tr></tr>
57:  <tr></tr>
58:  <tr></tr>
59:  <tr></tr>
60:  <tr>
61:
62:  <td colspan="2"><div align="left"><font size="2" face="Arial Black">
63:  FILES:</font></span></div></td>
64:  </tr>
65:  <tr>
66:  <td><div align="left"><font size="2" face="Arial Black" color= "blue">
```

```
67: DATA FILE:</font></div></td>
68: </tr>
69: <tr>
70:
71: <td><div align="left"><font size="2" face="Arial Black" color= "blue">
72: Seleccione el tipo de archivo a subir</font></div></td>
73: </tr>
74: <tr>
75:
76: <td>
77: <div align="left">
78: <INPUT type="radio" name="tipo" value="2"> <font size="2" face="Arial Black"
79: color= "blue"> SwRx</font><BR>
80: <INPUT type="radio" name="tipo" value="1"><font size="2" face="Arial Black"
81: color= "blue"> RINEXobs</font><BR>
82: </div>
83:
84: </tr>
85:
86: <tr>
87: <td>
88: <div align="left"> <input type="file" name="archivo" size="30"></div>
89:
90: </td>
91: </tr>
92: <tr></tr>
93: <tr></tr>
94: <tr></tr>
95: <tr></tr>
96: <tr>
97: <td><div align="left"><font size="2" face="Arial Black" color= "blue">
98: EPHEMERIDS FILE: </font></div></td>
99: </tr>
100: <tr>
101: <td>
102: <div align="left"> <input type="file" name="archivo1" size="30"></div>
103:
104: <div align="center"><input type="submit" value="Enviar" name="send"></div>
105: </td>
106: </tr>
107: <tr>
108: <td><div align="center"></div>
109: <div align="center"><input name="Borrar" type="reset" id="Borrar"
110: value="Borrar" /></div>
111: </td>
112: </tr>
113:
114:
115: </form>
116: </body>
117: </html>
118:
119:
120:
121:
122:
123:
```

```
1:  <?php
2:  $archivo= "/home/web/www/ana00/objetos/data.xml";
3:  if(is_file($archivo))
4:  {
5:  unlink($archivo);
6:  include("formulario.html");
7:
8:  }
9:  else
10:  {
11:    include("formulario.html");
12:  }
13:
14:
15:  ?>
```

```
1:  <html>
2:  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3:  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4:  <html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
5:  <head>
6:  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
7:  <title>Mapa de Google</title>
8:  <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA1Mva
9:  -SoetL-ErkTVRb01zRSctkfHrS9mIs8KjEnA60IulV_HzRRCgAJR0eow3Fsv2FVWimOHhWztsw"
10:  type="text/javascript"></script>
11:  <script type="text/javascript">
12:  //
13:
14:  function load() {
15:    if (GBrowserIsCompatible()) {
16:      var map = new GMap2(document.getElementById("map"));
17:      map.addControl(new GLargeMapControl());
18:      map.addControl(new GMapTypeControl());
19:      map.setMapType(G_NORMAL_MAP);
20:
21:      var iconoMarca = new GIcon(G_DEFAULT_ICON);
22:      iconoMarca.image = "http://maps.google.com/mapfiles/kml/pal4/icon49.png";
23:      var tamanoIcono = new GSize (20,20);
24:      iconoMarca.iconSize = tamanoIcono;
25:      iconoMarca.shadow = "http://maps.google.com/mapfiles/kml/pal4/icon49s.png";
26:      var tamanoSombra = new GSize(40,20);
27:      iconoMarca.shadowSize = tamanoSombra;
28:      iconoMarca.iconAnchor = new GPoint(10,20);
29:
30:
31:
32:      GDownloadUrl("objetos/data.xml", function(data) {
33:      var xml = GXml.parse(data);
34:      var markers = xml.documentElement.getElementsByTagName("marker");
35:      for (var i = 0; i &lt; markers.length; i++) {
36:        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
37:                                parseFloat(markers[i].getAttribute("lng")));
38:        var marker = new GMarker(point, iconoMarca);
39:        map.addOverlay(marker);
40:      }
41:      var pcenter = new GLatLng(parseFloat(markers[0].getAttribute("lat")),
42:                                parseFloat(markers[0].getAttribute("lng")));
43:      map.setCenter(pcenter,15);
44:    });
45:  });
46:
47:
48:
49:
50:
51:
52:
53:
54:
55:
56:      }
57:  }
58:
59:
60:  window.onload=load
61:
62:  //]]&gt;
63:  &lt;/script&gt;
64:  &lt;body&gt;
65:  &lt;div id= "global"&gt;
66:  &lt;div id="map" style="width: 100%; height: 85%"&gt;&lt;/div&gt;</pre></div><div data-bbox="471 960 523 975" data-label="Page-Footer">- 1 -</div>
```

```
67: <BODY background="objetos/brujula.png">
68: <a href="lineas.html">MOSTRAR RUTA</a>
69: <p>
70: <a href="formulario.php">Rellenar otro formulario</a>
71: <p>
72:
73: </ul></div>
74:
75: </div>
76:
77: <br />
78: <br />
79: </body>
80: </html>
```

```
1:
2:
3:
4:  <html>
5:  <head>
6:  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
7:  <title>GOOGLE-GPS</title>
8:  </head>
9:  <BODY background="objetos/brujula.png">
10:
11:  <div style="text-align: center">
12:
13:    <font size="10" face="Arial Black">Bienvenidos a google-GPS </font>
14:    <p>
15:      <a href="formulario.php">
16:        </a>
17:
18:
19:
20:    </div>
21:
22:  </body>
23: </html>
```

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2: "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3: <html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
4: <head>
5: <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
6: <title>Mapa de Google</title>
7: <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA1Mva-
8: SoetL-ErkTVRb01zRSctkfHrS9mIs8KjEnA60IulV_HzRRCgAJR0eow3Fsv2FVWimOHhWztsw"
9: type="text/javascript"></script>
10: <script type="text/javascript">
11: //
12:
13: function load() {
14:     if (GBrowserIsCompatible()) {
15:         var map = new GMap2(document.getElementById("map"));
16:         var pcenter=new GLatLng(41.50035, 2.104697);
17:         map.addControl(new GLargeMapControl());
18:         map.setCenter(pcenter,8);
19:         map.addControl(new GMapTypeControl());
20:         map.setMapType(G_NORMAL_MAP);
21:
22:         var iconoMarca = new GIcon(G_DEFAULT_ICON);
23:         iconoMarca.image = "http://maps.google.com/mapfiles/kml/pal4/icon49.png";
24:         var tamanoIcono = new GSize (20,20);
25:         iconoMarca.iconSize = tamanoIcono;
26:         iconoMarca.shadow = "http://maps.google.com/mapfiles/kml/pal4/icon49s.png";
27:         var tamanoSombra = new GSize(40,20);
28:         iconoMarca.shadowSize = tamanoSombra;
29:         iconoMarca.iconAnchor = new GPoint(10,20);
30:
31:
32:
33:         GDownloadUrl("objetos/data.xml", function(data) {
34: var xml = GXml.parse(data);
35: var markers = xml.documentElement.getElementsByTagName("marker");
36: for (var i = 0; i &lt; markers.length-1; i++) {
37:     var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
38:                             parseFloat(markers[i].getAttribute("lng")));
39:     var point2 = new GLatLng(parseFloat(markers[i+1].getAttribute("lat")),
40:                             parseFloat(markers[i+1].getAttribute("lng")));
41:     var marker = new GMarker(point, iconoMarca);
42:     var marker2 = new GMarker(point2, iconoMarca);
43:     map.addOverlay(marker2);
44:     map.addOverlay(marker);
45:     var opciones_poligono = {geodesic:true};
46:     var poligono = new GPolyline([point, point2], "#cccc00", 6, 0.6,
opciones_poligono);
47:     map.addOverlay(poligono);
48:
49: }
50: var pcenter = new GLatLng(parseFloat(markers[0].getAttribute("lat")),
51:                             parseFloat(markers[0].getAttribute("lng")));
52: map.setCenter(pcenter,12);
53: });
54:
55:
56:
57:
58:
59:
60:
61:
62:
63:
64:
65: }</pre></div><div data-bbox="471 961 523 975" data-label="Page-Footer">- 1 -</div>
```

```
66:  }
67:
68:
69:  window.onload=load
70:
71:  //]]>
72:  </script>
73:
74:  <BODY background="objetos/brujula.png">
75:
76:  </head>
77:  <body>
78:
79:  <div id= "global">
80:    <div id="map" style="width: 100%; height: 580px"></div>
81:    <div id= "cabecera"><font size="2" face="Arial Black"> bienvenidos a la ruta de
google-GPS <ul>
82:      <a href="googlemaps.html">VOLVER A MAPA</a>
83:      <p>
84:      <a href="formulario.php">Rellenar otro formulario</a>
85:      <p>
86:
87:    </ul></div>
88:
89:  </div>
90:
91:  <br />
92:  <br />
93:  </body>
94:  </html>
```

```
1:  <?php
2:
3:
4:
5:  if(empty($_POST["latitude"])){
6:  echo "<b><center>Ingrese un valor de latitud inicial</b><BR><BR>";
7:  }
8:  if(empty($_POST["longitude"])){
9:  echo "<b><center>- Ingrese un valor de longitud inicial.</b><BR><BR>";
10: }
11: if(empty($_POST["height"])){
12: echo "<b><center>- Ingrese un valor de altura.</b><BR><BR>";
13: }
14: if(empty($_POST["tipo"])){
15: echo "<b><center>- seleccione el tipo de archivo de observacion que
    subirá</b><BR><BR>";
16: }
17: else {
18:
19: $latitude=$_POST["latitude"];
20: $longitude=$_POST["longitude"];
21: $height=$_POST["height"];
22: $tipo=$_POST["tipo"];
23:
24: $string="$latitude\n$longitude\n$height\n$tipo";
25: $archivod='/home/web/www/ana00/objetos/datos.m';
26: $a1 = fopen($archivod, "w");
27: fwrite($a1, $string);
28: fclose($a1);
29: $path="/home/web/www/ana00/objetos";
30: $nombre1= "file.08n";
31: $nombre="file.08o";
32:
33: $ruta_archivo1= "$path/$nombre1";
34: $ruta_archivo= "$path/$nombre";
35:
36: if (is_uploaded_file($_HTTP_POST_FILES['archivo']['tmp_name']))
37: {
38:     copy($_HTTP_POST_FILES['archivo']['tmp_name'], $ruta_archivo);
39:     if (is_uploaded_file($_HTTP_POST_FILES['archivo1']['tmp_name']))
40:     {
41:         copy($_HTTP_POST_FILES['archivo1']['tmp_name'], $ruta_archivo1);
42:         exec("/home/web/www/ana00/run_mainWEB.sh /usr/local/matlab/MCR/v76");
43:     }
44: }
45:
46:
47: else { echo "error al subir el archivo de navegación";}
48:
49:
50:
51:
52: }
53:
54:
55: else { echo "error al subir el archivo de observación";}
56: }
57:
58:
59:
60:
61:
62:
63:
64:
65:
```

```
66:  if(is_file("/home/web/www/ana00/objetos/data.xml"))
67:  {
68:      chmod("/home/web/www/ana00/objetos/data.xml", 0777);
69:      include ("googlemaps.html");
70:  }
71:  else
72:  {
73:      sleep(5);
74:      include ("googlemaps.html");
75:  }
76:
77:
78:
79:  ?>
80:
81:
82:
83:
84:
```