Remote_control Stellarium V2

October 2, 2022

Código para importar "request" paquete utilizado para la comunicación con APIs

```
[1]: import requests
```

```
[2]: import json
     import numpy as np
     import sympy as sp
     import pandas as pd
     import math as mth
     import time
     from decimal import Decimal
     import matplotlib.pyplot as plt
     import datetime
     import jdcal as j
```

```
[3]: url_main = "http://localhost:8090/api/" #url para conexión con Stellarium, se␣
      ↪debe verificar en la conexión API
```

```
[4]: url_status = "main/status"
```

Se traen los datos de configuración actual, se extraen los requeridos como ubicación y tiempo

```
[5]: response_status = requests.get(url_main+url_status)
     print("Conexion con el servido principal: ", response_status.status_code)
     response_status.json()
```

```
Conexion con el servido principal:  200
```

```
[5]: {'location': {'altitude': 993,
       'landscapeKey': '',
       'latitude': 7.12539005279541,
       'longitude': -73.11979675292969,
       'name': 'Bucaramanga',
       'planet': 'Earth',
       'region': 'Southern America',
       'role': 'R',
       'state': 'Santander'},
      'selectioninfo': '',
 'time': {'deltaT': 0.0008002476446557006,
  'gmtShift': -0.20833333333333334,
  'isTimeNow': True,
  'jday': 2459834.736770116,
  'local': '2022-09-12T00:40:56.938',
  'timeZone': 'UTC-05:00',
  'timerate': 1.157407407407407e-05,
  'utc': '2022-09-12T05:40:56.938Z'},
 'view': {'fov': 42.991878634427316}}
```

[6]:

```python
json_status = response_status.json()
phi = json_status["location"]["latitude"]
time_var = json_status["time"]["jday"]
# print("Data setup: " , json_status)
```

Se Estipula el día y la fecha

[7]:

```python
#print("Ingrese la fecha en el formato AA-M-DTH:M:S")
#Date_time = str(input())
Date_time = "2021-12-26T18:30:00"
Date_time =time.strptime(Date_time,"%Y-%m-%dT%H:%M:%S")
Julian_date = sum(j.gcal2jd(Date_time.tm_year,Date_time.tm_mon,Date_time.
 ↪tm_mday))+0.5
Julian_date_dec = (((Date_time.tm_hour-7)*3600)+(Date_time.
 ↪tm_min*60)+(Date_time.tm_sec))*(1/86400)
Julian_date = Julian_date + Julian_date_dec
print(Julian_date)
url_time ="main/time"
params_time ={"time": Julian_date}
response_time = requests.post(url_main + url_time, data = params_time)
```

```
2459575.4791666665
```

Se realiza el tracking del objeto puesto en la variable target

[8]:

```python
url_obj_focus = "main/focus"
#En target poner la estrella o astro que se quiere seguir puede ser cualquiera␣
 ↪de la lista
#Yildun,Circitores, Alahakan, Alfirk, Alderamín, HIP 62046, Eltanin, Vega,␣
 ↪Sadir, Sulafat, Albireo, Sualocin
#Kornephoros, Rosalhague, Altair, Miaplacidus, Mimosa, Muhlifain, Menkent, Spica
target = "Sadalmelik"
params_focus={"target":target}
#Se pone en enfoque el objeto celeste deseado
response_focus = requests.post(url_main + url_obj_focus, data = params_focus)
response_focus.status_code
```

[8]:

```
200
```

[9]:

```
#El script a ejecutar es angulos_star.ssc
# Para poder correr el script, el script debe estar en la carpeta scripts de la␣
 ↪carpeta de instalación de stellarium
# esta se puede encontrar abriendo la ubicacion del archivo del ejecutable que␣
 ↪se instala

#El objetivo de este script es simplemente seguir el objetivo que se enfoco a␣
 ↪lo largo de la noche con un tiempo acelerado
url_script_info = "scripts/info"
params_info = {"id" : "angulos_star.ssc"}
script_info = requests.get(url_main +url_script_info,data = params_info)
json_scrp_info = script_info.json()
print("informacion del script: " ,json_scrp_info)
```

```
informacion del script:  {'author': 'David Becerra. 09/09/2021', 'description':
'', 'description_localized': '', 'id': 'angulos_star.ssc', 'license': 'Public
Domain', 'name': 'Motion of a specific Star', 'name_localized': 'Motion of a
specific Star', 'version': ''}
```

[10]:

```
#Se pone en ejecución el script
url_script = "scripts/run"
params_run ={"id" :"angulos_star.ssc"}
script_run = requests.post(url_main + url_script, data = params_run)

#se lee el status del script
url_scrun_info ="scripts/status"
script_runinfo = requests.get(url_main + url_scrun_info)
json_scrun_info = script_runinfo.json()
status_script = json_scrun_info["scriptIsRunning"]
azimut = []
altitud = []
tiempo = []
tiempo_jd =[]
tiempo_str =[]
```

[11]:

```
while status_script:
    ##Apunta al objeto
    response_focus_script = requests.post(url_main + url_obj_focus, data =␣
 ↪params_focus)
    ##Informacion del objeto
    url_obj_info = "objects/info"
    params_obj_info = {"name" : target, "format" : "json"}
    response_obj = requests.get(url_main + url_obj_info, data = params_obj_info)
    #print("Status informacion objeto: ", response_obj.status_code)
    json_objinfo = response_obj.json()
    azimuth = json_objinfo["azimuth"]
    altitude = json_objinfo["altitude"]
    #print("Azimuth: ", azimuth)
    #print("Altitud: ", altitude)
    azimut.append(azimuth)
    altitud.append(altitude)
    #print("status del script: ",status_script )
    ##Obtiene el tiempo en JS
    response_status = requests.get(url_main+url_status)
    json_status = response_status.json()
    time_var = json_status["time"]["jday"]
    time_var_utc = json_status["time"]["local"]
    time_pract = time_var_utc
    time_var_utc =time.strptime(time_var_utc,"%Y-%m-%dT%H:%M:%S.%f")

    #print("tiempo en JD: ", time_var, ",", type(time_var))
    #print("Time en UTC", time_var_utc, ",", type(time_var_utc))
    tiempo.append(time_var_utc)
    tiempo_jd.append(time_var)
    tiempo_str.append(time_pract)
    ##Revisa si el script se está ejecutando
    script_runinfo = requests.get(url_main + url_scrun_info)
    json_scrun_info = script_runinfo.json()
    status_script = json_scrun_info["scriptIsRunning"]
    time.sleep(0.0002)
    if (time_var_utc.tm_mday > 10 & time_var_utc.tm_hour > 18):
        status_script = False
```

Se calcula los vectores de velocidad

[12]:

```
#Se crean las nuevas listas que almacenarán los datos
delta_azimut = []
delta_tim_sec = []
rpm_azimut = []
vel_azimut =[]

delta_ele = []
vel_ele =[]
rpm_ele = []

date_calc = []
```

[13]:

```
#Funciones para saber si el azimut está en el primero o cuarto cuadrante
def flag_az_4(var_flag):
    flag_az = bool(270<=var_flag<=360)
    return flag_az
def flag_az_1(var_flag):
    flag_az = bool(0<=var_flag<=90)
    return flag_az
```

[14]:

```
#Calcula las velocidades de azimut, altitud y le da el tiempo
for i in range(len(azimut)-1):
    #Si el azimut pasa del primer al cuarto cuadrante
    if flag_az_4(azimut[i+1])&flag_az_1(azimut[i]):
        azimut_mod = azimut[i+1] - 360
        delta_azimut.append(azimut_mod -azimut[i])
        delta_azimut[i] = abs(delta_azimut[i])
        #print("este es el azimut inicial: ",azimut[i])
        #print("este es el azimut final: ", azimut[i+1])
        #print("1-4",delta_azimut[i])
    #Si el azimut pasa del cuarto al primer cuadrante
    elif flag_az_1(azimut[i+1]) & flag_az_4(azimut[i]):
        azimut_mod =azimut[i]-360
        delta_azimut.append(azimut[i+1]-azimut_mod)
        delta_azimut[i] = abs(delta_azimut[i])
        #print("este es el azimut inicial: ",azimut[i])
        #print("este es el azimut final: ", azimut[i+1])
        #print("4-1",delta_azimut[i])
    #En cualquier otro caso
    else:
        delta_azimut.append(azimut[i+1]-azimut[i])
        delta_azimut[i] = abs(delta_azimut[i])
        #print("este es el azimut inicial: ",azimut[i])
        #print("este es el azimut final: ", azimut[i+1])
        #print("1-1 o 4-4", delta_azimut[i])
    #print(delta_azimut[i])
    #Calcula la diferencia de tiempo en segundos y lo añade a delta_tim_sec
    delta_tim_sec.append((tiempo_jd[i+1] - tiempo_jd[i])*86400)
    #Calculo de las velocidades azimutales
    rpm_azimut.append((delta_azimut[i]/delta_tim_sec[i])*(1/360)*(60))
    vel_azimut.append(delta_azimut[i]/delta_tim_sec[i])
    #Calculo de la velocidades de elevacion
    delta_ele.append((altitud[i-1]-altitud[i]))
    delta_ele[i]=abs(delta_ele[i])
    vel_ele.append(delta_ele[i]/delta_tim_sec[i])
    rpm_ele.append(vel_ele[i]*(1/360)*60)
    #Se da el valor de fecha
    date_calc.append(tiempo[i])

    #print("Cambio en la altitud: ", delta_ele[i])
    #print("La velocidad azimutal es: ", vel_azimut)
    #print("La velocidadazimutal  en revoluciones por minuto es: ",␣
↪rpm_azimut[i])
```

[15]:
```
#Solo toma las horas de los datos
Hora_data = []
for i in range(len(tiempo)):
    Hora_data.append(str(tiempo[i].tm_hour) +":" + str(tiempo[i].tm_min) + ":"+␣
↪str(tiempo[i].tm_sec))
Hora_vel = []
for i in range(len(date_calc)):
    Hora_vel.append(str(date_calc[i].tm_hour) +":" + str(date_calc[i].tm_min) +␣
↪":"+ str(date_calc[i].tm_sec))
```

[16]:
```
#azimut_pd = pd.Series(vel_azimut, index = date_calc)
#altitud_pd = pd.Series(vel_ele, index = date_calc)
Astronomic_data_vel = pd.DataFrame({"velocidad azimut":vel_azimut, "velocidad␣
↪altitud": vel_ele,"Date":date_calc, "Hora":Hora_vel})

#Datos de azimiut y elevación
AstroData = {"azimut": azimut, "elevacion": altitud, "Date":tiempo, "Hora":␣
↪Hora_data}
Astro_data = pd.DataFrame(AstroData)
pd.options.display.max_rows=None

#Datos de posiciones
print("Posiciones: ")
print(Astro_data)

#Datos de velocidades
print("Velocidades:")
print(Astronomic_data_vel)
```

```
Posiciones:
         azimut  elevacion                                     Date      Hora
0     71.807812 -68.830187    (2021, 6, 9, 18, 7, 30, 2, 160, -1)   18:7:30
1     73.511884 -66.935221   (2021, 6, 9, 18, 15, 28, 2, 160, -1)  18:15:28
2     74.980095 -65.010287   (2021, 6, 9, 18, 23, 31, 2, 160, -1)  18:23:31
3     76.241946 -63.087457   (2021, 6, 9, 18, 31, 30, 2, 160, -1)  18:31:30
4     77.351302 -61.147590   (2021, 6, 9, 18, 39, 32, 2, 160, -1)  18:39:32
5     78.445954 -58.958538   (2021, 6, 9, 18, 48, 28, 2, 160, -1)  18:48:28
6     79.309751 -57.003451   (2021, 6, 9, 18, 56, 29, 2, 160, -1)  18:56:29
7     80.084837 -55.050419    (2021, 6, 9, 19, 4, 30, 2, 160, -1)   19:4:30
```

```
8     80.790686 -53.085609   (2021, 6, 9, 19, 12, 30, 2, 160, -1)  19:12:30
9     81.435243 -51.117039   (2021, 6, 9, 19, 20, 29, 2, 160, -1)  19:20:29
10    82.061454 -49.027016    (2021, 6, 9, 19, 29, 0, 2, 160, -1)   19:29:0
11    82.604226 -47.059742   (2021, 6, 9, 19, 36, 59, 2, 160, -1)  19:36:59
12    83.110534 -45.082723    (2021, 6, 9, 19, 45, 0, 2, 160, -1)   19:45:0
13    83.581590 -43.111077   (2021, 6, 9, 19, 52, 58, 2, 160, -1)  19:52:58
14    84.025177 -41.130240    (2021, 6, 9, 20, 0, 59, 2, 160, -1)   20:0:59
15    84.467382 -39.029003   (2021, 6, 9, 20, 10, 28, 2, 160, -1)  20:10:28
16    84.957128 -36.547147   (2021, 6, 9, 20, 19, 30, 2, 160, -1)  20:19:30
17    85.371315 -34.316380   (2021, 6, 9, 20, 28, 30, 2, 160, -1)  20:28:30
18    85.768268 -32.062012    (2021, 6, 9, 20, 39, 0, 2, 160, -1)   20:39:0
19    86.161133 -29.717167    (2021, 6, 9, 20, 47, 2, 2, 160, -1)   20:47:2
20    86.477958 -27.743697    (2021, 6, 9, 20, 55, 1, 2, 160, -1)   20:55:1
21    86.788328 -25.739763     (2021, 6, 9, 21, 3, 0, 2, 160, -1)    21:3:0
22    87.084294 -23.765040    (2021, 6, 9, 21, 11, 0, 2, 160, -1)   21:11:0
23    87.373826 -21.774896    (2021, 6, 9, 21, 19, 1, 2, 160, -1)   21:19:1
24    87.654668 -19.791753    (2021, 6, 9, 21, 27, 0, 2, 160, -1)   21:27:0
25    87.946192 -17.681437   (2021, 6, 9, 21, 35, 29, 2, 160, -1)  21:35:29
26    88.216291 -15.682623    (2021, 6, 9, 21, 44, 0, 2, 160, -1)   21:44:0
27    88.479461 -13.698437   (2021, 6, 9, 21, 51, 59, 2, 160, -1)  21:51:59
28    88.723982 -11.825921    (2021, 6, 9, 21, 59, 0, 2, 160, -1)   21:59:0
29    88.978802  -9.848770     (2021, 6, 9, 22, 7, 1, 2, 160, -1)    22:7:1
30    89.232743  -7.856540    (2021, 6, 9, 22, 15, 0, 2, 160, -1)   22:15:0
31    89.484524  -5.864191    (2021, 6, 9, 22, 23, 0, 2, 160, -1)   22:23:0
32    89.734778  -3.626372    (2021, 6, 9, 22, 31, 1, 2, 160, -1)   22:31:1
33    89.999047  -1.029784   (2021, 6, 9, 22, 40, 30, 2, 160, -1)  22:40:30
34    90.342348   1.344791    (2021, 6, 9, 22, 51, 0, 2, 160, -1)   22:51:0
35    90.607936   3.325347    (2021, 6, 9, 22, 59, 0, 2, 160, -1)   22:59:0
36    90.856977   5.239893     (2021, 6, 9, 23, 7, 1, 2, 160, -1)    23:7:1
37    91.077409   6.945618   (2021, 6, 9, 23, 13, 58, 2, 160, -1)  23:13:58
38    91.332259   8.913034    (2021, 6, 9, 23, 22, 1, 2, 160, -1)   23:22:1
39    91.590142  10.888009    (2021, 6, 9, 23, 30, 0, 2, 160, -1)   23:30:0
40    91.851742  12.867361    (2021, 6, 9, 23, 38, 0, 2, 160, -1)   23:38:0
41    92.115770  14.834524   (2021, 6, 9, 23, 45, 59, 2, 160, -1)  23:45:59
42    92.387998  16.825621    (2021, 6, 9, 23, 54, 1, 2, 160, -1)   23:54:1
43    92.664192  18.802868     (2021, 6, 10, 0, 2, 0, 3, 161, -1)     0:2:0
44    92.948316  20.788122    (2021, 6, 10, 0, 10, 1, 3, 161, -1)    0:10:1
45    93.240305  22.773643    (2021, 6, 10, 0, 18, 0, 3, 161, -1)    0:18:0
46    93.540063  24.751738    (2021, 6, 10, 0, 26, 0, 3, 161, -1)    0:26:0
47    93.851001  26.737082    (2021, 6, 10, 0, 34, 1, 3, 161, -1)    0:34:1
48    94.173291  28.722061    (2021, 6, 10, 0, 42, 1, 3, 161, -1)    0:42:1
49    94.508365  30.706511    (2021, 6, 10, 0, 50, 0, 3, 161, -1)    0:50:0
50    94.856492  32.682846    (2021, 6, 10, 0, 57, 1, 3, 161, -1)    0:57:1
51    95.198300  34.539539    (2021, 6, 10, 1, 5, 31, 3, 161, -1)    1:5:31
52    95.532588  36.276615   (2021, 6, 10, 1, 12, 28, 3, 161, -1)   1:12:28
53    95.931477  38.250156   (2021, 6, 10, 1, 20, 31, 3, 161, -1)   1:20:31
54    96.380752  40.348347    (2021, 6, 10, 1, 29, 0, 3, 161, -1)    1:29:0
55    96.772426  42.074505    (2021, 6, 10, 1, 36, 0, 3, 161, -1)    1:36:0
```

```
56   97.248676  44.050865   (2021, 6, 10, 1, 44, 0, 3, 161, -1)   1:44:0
57   97.724660  45.899506  (2021, 6, 10, 1, 51, 32, 3, 161, -1)  1:51:32
58   98.233766  47.746056   (2021, 6, 10, 1, 59, 2, 3, 161, -1)   1:59:2
59   98.817076  49.708138    (2021, 6, 10, 2, 7, 1, 3, 161, -1)    2:7:1
60   99.457336  51.689117   (2021, 6, 10, 2, 15, 1, 3, 161, -1)   2:15:1
61  100.056572  53.394592  (2021, 6, 10, 2, 22, 31, 3, 161, -1)  2:22:31
62  100.859407  55.477837  (2021, 6, 10, 2, 30, 32, 3, 161, -1)  2:30:32
63  101.692712  57.423752  (2021, 6, 10, 2, 38, 31, 3, 161, -1)  2:38:31
64  102.623645  59.370789  (2021, 6, 10, 2, 46, 30, 3, 161, -1)  2:46:30
65  103.668293  61.310360  (2021, 6, 10, 2, 54, 32, 3, 161, -1)  2:54:32
66  104.846251  63.233589   (2021, 6, 10, 3, 2, 31, 3, 161, -1)   3:2:31
67  106.203155  65.159973  (2021, 6, 10, 3, 10, 33, 3, 161, -1)  3:10:33
68  108.184450  67.526245   (2021, 6, 10, 3, 22, 3, 3, 161, -1)   3:22:3
69  111.040029  70.232407  (2021, 6, 10, 3, 32, 33, 3, 161, -1)  3:32:33
70  114.895504  72.967476   (2021, 6, 10, 3, 48, 1, 3, 161, -1)   3:48:1
71  120.713363  75.834709   (2021, 6, 10, 3, 58, 1, 3, 161, -1)   3:58:1
72  128.099481  78.202835   (2021, 6, 10, 4, 9, 30, 3, 161, -1)   4:9:30
73  138.250475  80.201546  (2021, 6, 10, 4, 19, 30, 3, 161, -1)  4:19:30
74  148.373448  81.396118  (2021, 6, 10, 4, 27, 32, 3, 161, -1)  4:27:32
75  161.985955  82.285332   (2021, 6, 10, 4, 36, 0, 3, 161, -1)   4:36:0
76  177.150688  82.650467   (2021, 6, 10, 4, 44, 2, 3, 161, -1)   4:44:2
77  192.653291  82.478722   (2021, 6, 10, 4, 52, 1, 3, 161, -1)   4:52:1
78  206.485901  81.809280    (2021, 6, 10, 5, 0, 1, 3, 161, -1)    5:0:1
79  217.748785  80.745293    (2021, 6, 10, 5, 8, 0, 3, 161, -1)    5:8:0
80  226.424827  79.411560   (2021, 6, 10, 5, 16, 1, 3, 161, -1)   5:16:1
81  233.456514  77.787048  (2021, 6, 10, 5, 24, 30, 3, 161, -1)  5:24:30
82  238.501853  76.140816  (2021, 6, 10, 5, 32, 31, 3, 161, -1)  5:32:31
83  242.484484  74.406349  (2021, 6, 10, 5, 40, 31, 3, 161, -1)  5:40:31
84  245.675464  72.614250  (2021, 6, 10, 5, 48, 32, 3, 161, -1)  5:48:32
85  248.280037  70.780655  (2021, 6, 10, 5, 56, 33, 3, 161, -1)  5:56:33
86  250.442564  68.916389   (2021, 6, 10, 6, 4, 31, 3, 161, -1)   6:4:31
87  252.259574  67.036049  (2021, 6, 10, 6, 12, 32, 3, 161, -1)  6:12:32
88  253.818984  65.130760  (2021, 6, 10, 6, 20, 31, 3, 161, -1)  6:20:31
89  255.163578  63.218609  (2021, 6, 10, 6, 28, 31, 3, 161, -1)  6:28:31
90  256.348652  61.280800  (2021, 6, 10, 6, 36, 30, 3, 161, -1)  6:36:30
91  257.380350  59.362946  (2021, 6, 10, 6, 44, 31, 3, 161, -1)  6:44:31
92  258.370006  57.284374  (2021, 6, 10, 6, 53, 31, 3, 161, -1)  6:53:31
93  259.247261  55.213493   (2021, 6, 10, 7, 1, 31, 3, 161, -1)   7:1:31
94  260.039764  53.129581   (2021, 6, 10, 7, 10, 4, 3, 161, -1)   7:10:4
95  260.755966  51.048244  (2021, 6, 10, 7, 18, 30, 3, 161, -1)  7:18:30
96  261.376775  49.073528   (2021, 6, 10, 7, 27, 1, 3, 161, -1)   7:27:1
97  262.045710  46.748619   (2021, 6, 10, 7, 36, 0, 3, 161, -1)   7:36:0
98  262.568588  44.775185  (2021, 6, 10, 7, 44, 31, 3, 161, -1)  7:44:31
99  263.084748  42.681099  (2021, 6, 10, 7, 52, 31, 3, 161, -1)  7:52:31
100 263.540406  40.703472   (2021, 6, 10, 8, 0, 32, 3, 161, -1)   8:0:32
101 263.994362  38.605564    (2021, 6, 10, 8, 9, 1, 3, 161, -1)    8:9:1
102 264.398595  36.624832   (2021, 6, 10, 8, 17, 2, 3, 161, -1)   8:17:2
103 264.783641  34.635487   (2021, 6, 10, 8, 25, 3, 3, 161, -1)   8:25:3
```

104	265.144975	32.674835	(2021, 6, 10, 8, 33, 1, 3, 161, -1)	8:33:1
105	265.495602	30.683640	(2021, 6, 10, 8, 41, 31, 3, 161, -1)	8:41:31
106	265.850169	28.580242	(2021, 6, 10, 8, 49, 31, 3, 161, -1)	8:49:31
107	266.191673	26.468832	(2021, 6, 10, 8, 58, 1, 3, 161, -1)	8:58:1
108	266.500039	24.490889	(2021, 6, 10, 9, 6, 1, 3, 161, -1)	9:6:1
109	266.818442	22.378925	(2021, 6, 10, 9, 14, 31, 3, 161, -1)	9:14:31
110	267.107714	20.400873	(2021, 6, 10, 9, 22, 31, 3, 161, -1)	9:22:31
111	267.390419	18.415701	(2021, 6, 10, 9, 30, 30, 3, 161, -1)	9:30:30
112	267.665377	16.438612	(2021, 6, 10, 9, 38, 33, 3, 161, -1)	9:38:33
113	267.936525	14.447783	(2021, 6, 10, 9, 46, 31, 3, 161, -1)	9:46:31
114	268.200622	12.473625	(2021, 6, 10, 9, 54, 30, 3, 161, -1)	9:54:30
115	268.461427	10.494975	(2021, 6, 10, 10, 2, 31, 3, 161, -1)	10:2:31
116	268.719612	8.513801	(2021, 6, 10, 10, 10, 31, 3, 161, -1)	10:10:31
117	268.972032	6.563221	(2021, 6, 10, 10, 18, 30, 3, 161, -1)	10:18:30
118	269.224116	4.614057	(2021, 6, 10, 10, 26, 31, 3, 161, -1)	10:26:31
119	269.474575	2.699834	(2021, 6, 10, 10, 34, 33, 3, 161, -1)	10:34:33
120	269.723093	0.878442	(2021, 6, 10, 10, 42, 32, 3, 161, -1)	10:42:32
121	269.972142	-0.816716	(2021, 6, 10, 10, 50, 31, 3, 161, -1)	10:50:31
122	270.221388	-3.197883	(2021, 6, 10, 10, 58, 31, 3, 161, -1)	10:58:31
123	270.470499	-5.506588	(2021, 6, 10, 11, 6, 32, 3, 161, -1)	11:6:32
124	270.721961	-7.498953	(2021, 6, 10, 11, 14, 33, 3, 161, -1)	11:14:33
125	270.974513	-9.483746	(2021, 6, 10, 11, 22, 33, 3, 161, -1)	11:22:33
126	271.230704	-11.475850	(2021, 6, 10, 11, 30, 32, 3, 161, -1)	11:30:32
127	271.491246	-13.475221	(2021, 6, 10, 11, 38, 33, 3, 161, -1)	11:38:33
128	271.754910	-15.466895	(2021, 6, 10, 11, 46, 31, 3, 161, -1)	11:46:31
129	272.022404	-17.450827	(2021, 6, 10, 11, 54, 32, 3, 161, -1)	11:54:32
130	272.295509	-19.434429	(2021, 6, 10, 12, 2, 34, 3, 161, -1)	12:2:34
131	272.576153	-21.425102	(2021, 6, 10, 12, 10, 33, 3, 161, -1)	12:10:33
132	272.864269	-23.415330	(2021, 6, 10, 12, 19, 1, 3, 161, -1)	12:19:1
133	273.177835	-25.516816	(2021, 6, 10, 12, 27, 1, 3, 161, -1)	12:27:1
134	273.485862	-27.513371	(2021, 6, 10, 12, 35, 2, 3, 161, -1)	12:35:2
135	273.802549	-29.494369	(2021, 6, 10, 12, 43, 3, 3, 161, -1)	12:43:3
136	274.131565	-31.474625	(2021, 6, 10, 12, 51, 1, 3, 161, -1)	12:51:1
137	274.475810	-33.461482	(2021, 6, 10, 12, 59, 2, 3, 161, -1)	12:59:2
138	274.835884	-35.447382	(2021, 6, 10, 13, 7, 31, 3, 161, -1)	13:7:31
139	275.238571	-37.558534	(2021, 6, 10, 13, 15, 32, 3, 161, -1)	13:15:32
140	275.636625	-39.534616	(2021, 6, 10, 13, 23, 31, 3, 161, -1)	13:23:31
141	276.057656	-41.509303	(2021, 6, 10, 13, 31, 31, 3, 161, -1)	13:31:31
142	276.536110	-43.615858	(2021, 6, 10, 13, 40, 1, 3, 161, -1)	13:40:1
143	277.017122	-45.594417	(2021, 6, 10, 13, 48, 1, 3, 161, -1)	13:48:1
144	277.530894	-47.563461	(2021, 6, 10, 13, 56, 2, 3, 161, -1)	13:56:2
145	278.086549	-49.537469	(2021, 6, 10, 14, 4, 3, 3, 161, -1)	14:4:3
146	278.686194	-51.501267	(2021, 6, 10, 14, 12, 1, 3, 161, -1)	14:12:1
147	279.341951	-53.469150	(2021, 6, 10, 14, 20, 2, 3, 161, -1)	14:20:2
148	280.060969	-55.433151	(2021, 6, 10, 14, 28, 3, 3, 161, -1)	14:28:3
149	280.854728	-57.392555	(2021, 6, 10, 14, 36, 3, 3, 161, -1)	14:36:3
150	281.737538	-59.346466	(2021, 6, 10, 14, 44, 2, 3, 161, -1)	14:44:2
151	282.790646	-61.410203	(2021, 6, 10, 14, 52, 31, 3, 161, -1)	14:52:31

```
152  283.919415 -63.348861   (2021, 6, 10, 15, 0, 34, 3, 161, -1)   15:0:34
153  285.210369 -65.277395   (2021, 6, 10, 15, 8, 33, 3, 161, -1)   15:8:33
154  286.697605 -67.186074  (2021, 6, 10, 15, 16, 31, 3, 161, -1)  15:16:31
155  288.445961 -69.085757  (2021, 6, 10, 15, 24, 34, 3, 161, -1)  15:24:34
156  290.515286 -70.957391  (2021, 6, 10, 15, 32, 33, 3, 161, -1)  15:32:33
157  293.023274 -72.808156  (2021, 6, 10, 15, 40, 31, 3, 161, -1)  15:40:31
158  296.108339 -74.620865  (2021, 6, 10, 15, 48, 34, 3, 161, -1)  15:48:34
159  299.980809 -76.380304  (2021, 6, 10, 15, 56, 33, 3, 161, -1)  15:56:33
160  304.926228 -78.056766   (2021, 6, 10, 16, 4, 33, 3, 161, -1)   16:4:33
161  311.900505 -79.720290   (2021, 6, 10, 16, 13, 1, 3, 161, -1)   16:13:1
162  320.645744 -81.096224   (2021, 6, 10, 16, 21, 3, 3, 161, -1)   16:21:3
163  332.274835 -82.206766   (2021, 6, 10, 16, 29, 4, 3, 161, -1)   16:29:4
164  346.736557 -82.904555   (2021, 6, 10, 16, 37, 4, 3, 161, -1)   16:37:4
165    3.189442 -83.081274   (2021, 6, 10, 16, 45, 3, 3, 161, -1)   16:45:3
166   19.208375 -82.689109   (2021, 6, 10, 16, 53, 6, 3, 161, -1)   16:53:6
167   32.565906 -81.819960    (2021, 6, 10, 17, 1, 3, 3, 161, -1)    17:1:3
168   43.043105 -80.589085    (2021, 6, 10, 17, 9, 1, 3, 161, -1)    17:9:1
169   50.874415 -79.135065   (2021, 6, 10, 17, 17, 2, 3, 161, -1)   17:17:2
170   57.125275 -77.424127  (2021, 6, 10, 17, 25, 31, 3, 161, -1)  17:25:31
171   61.610297 -75.714901  (2021, 6, 10, 17, 33, 32, 3, 161, -1)  17:33:32
172   65.151437 -73.932849  (2021, 6, 10, 17, 41, 31, 3, 161, -1)  17:41:31
173   67.995049 -72.104120  (2021, 6, 10, 17, 49, 31, 3, 161, -1)  17:49:31
174   70.315590 -70.248705  (2021, 6, 10, 17, 57, 34, 3, 161, -1)  17:57:34
Velocidades:
     velocidad azimut  velocidad altitud  \
0            0.003559           0.002963
1            0.003044           0.003928
2            0.002635           0.004020
3            0.002300           0.003986
4            0.002041           0.003616
5            0.001797           0.004555
6            0.001613           0.004068
7            0.001469           0.004064
8            0.001346           0.004104
9            0.001225           0.003851
10           0.001134           0.004365
11           0.001053           0.004093
12           0.000984           0.004129
13           0.000923           0.004102
14           0.000777           0.003482
15           0.000904           0.003878
16           0.000767           0.004596
17           0.000630           0.003541
18           0.000814           0.004673
19           0.000662           0.004897
20           0.000648           0.004122
21           0.000616           0.004170
22           0.000602           0.004109
```

23	0.000587	0.004157
24	0.000572	0.003893
25	0.000528	0.004128
26	0.000550	0.004175
27	0.000581	0.004711
28	0.000530	0.003896
29	0.000530	0.004129
30	0.000524	0.004145
31	0.000521	0.004146
32	0.000465	0.003934
33	0.000545	0.004122
34	0.000553	0.004941
35	0.000518	0.004121
36	0.000528	0.004585
37	0.000528	0.003536
38	0.000539	0.004109
39	0.000544	0.004109
40	0.000551	0.004134
41	0.000564	0.004078
42	0.000577	0.004159
43	0.000591	0.004114
44	0.000610	0.004146
45	0.000624	0.004131
46	0.000647	0.004116
47	0.000671	0.004131
48	0.000700	0.004146
49	0.000827	0.004711
50	0.000671	0.003880
51	0.000800	0.004446
52	0.000827	0.003601
53	0.000882	0.003874
54	0.000934	0.005003
55	0.000991	0.003592
56	0.001054	0.004374
57	0.001131	0.004108
58	0.001218	0.003857
59	0.001332	0.004083
60	0.001332	0.004402
61	0.001670	0.003549
62	0.001740	0.004351
63	0.001944	0.004064
64	0.002166	0.004036
65	0.002460	0.004051
66	0.002813	0.003987
67	0.002874	0.002794
68	0.004533	0.003756
69	0.004151	0.002914
70	0.009706	0.004563

71	0.010714	0.004159
72	0.016935	0.003951
73	0.020985	0.004143
74	0.026817	0.002353
75	0.031436	0.001843
76	0.032378	0.000763
77	0.028782	0.000357
78	0.023523	0.001398
79	0.018053	0.002214
80	0.013804	0.002618
81	0.010498	0.003380
82	0.008287	0.003425
83	0.006640	0.003609
84	0.005419	0.003729
85	0.004517	0.003830
86	0.003781	0.003879
87	0.003257	0.003927
88	0.002798	0.003964
89	0.002475	0.003994
90	0.002147	0.004032
91	0.001833	0.003552
92	0.001825	0.004325
93	0.001545	0.004037
94	0.001416	0.004120
95	0.001214	0.004071
96	0.001243	0.003669
97	0.001023	0.004548
98	0.001074	0.004106
99	0.000948	0.004357
100	0.000891	0.003882
101	0.000841	0.004365
102	0.000801	0.004121
103	0.000755	0.004155
104	0.000688	0.003849
105	0.000738	0.004143
106	0.000670	0.004129
107	0.000642	0.004393
108	0.000625	0.003883
109	0.000602	0.004394
110	0.000590	0.004131
111	0.000570	0.004115
112	0.000566	0.004129
113	0.000552	0.004158
114	0.000543	0.004108
115	0.000537	0.004117
116	0.000527	0.004138
117	0.000525	0.004059
118	0.000519	0.004041

119	0.000519	0.003998
120	0.000520	0.003804
121	0.000519	0.003527
122	0.000518	0.004955
123	0.000523	0.004804
124	0.000525	0.004146
125	0.000535	0.004145
126	0.000542	0.004145
127	0.000551	0.004176
128	0.000557	0.004144
129	0.000566	0.004113
130	0.000586	0.004143
131	0.000568	0.003922
132	0.000652	0.004141
133	0.000641	0.004373
134	0.000659	0.004154
135	0.000687	0.004137
136	0.000716	0.004120
137	0.000707	0.003900
138	0.000838	0.004132
139	0.000831	0.004409
140	0.000876	0.004112
141	0.000939	0.003876
142	0.001001	0.004383
143	0.001069	0.004117
144	0.001156	0.004097
145	0.001252	0.004123
146	0.001364	0.004086
147	0.001496	0.004095
148	0.001652	0.004087
149	0.001844	0.004092
150	0.002067	0.003836
151	0.002340	0.004278
152	0.002696	0.004049
153	0.003106	0.004028
154	0.003624	0.003957
155	0.004322	0.003968
156	0.005238	0.003909
157	0.006395	0.003837
158	0.008088	0.003786
159	0.010290	0.003661
160	0.013740	0.003303
161	0.018129	0.003448
162	0.024197	0.002863
163	0.030091	0.002311
164	0.034363	0.001457
165	0.033207	0.000366
166	0.028003	0.000822

```
167          0.021882           0.001815
168          0.016295           0.002561
169          0.012271           0.002854
170          0.009332           0.003560
171          0.007396           0.003570
172          0.005917           0.003708
173          0.004810           0.003791

                                      Date      Hora
0      (2021, 6, 9, 18, 7, 30, 2, 160, -1)   18:7:30
1     (2021, 6, 9, 18, 15, 28, 2, 160, -1)  18:15:28
2     (2021, 6, 9, 18, 23, 31, 2, 160, -1)  18:23:31
3     (2021, 6, 9, 18, 31, 30, 2, 160, -1)  18:31:30
4     (2021, 6, 9, 18, 39, 32, 2, 160, -1)  18:39:32
5     (2021, 6, 9, 18, 48, 28, 2, 160, -1)  18:48:28
6     (2021, 6, 9, 18, 56, 29, 2, 160, -1)  18:56:29
7      (2021, 6, 9, 19, 4, 30, 2, 160, -1)   19:4:30
8     (2021, 6, 9, 19, 12, 30, 2, 160, -1)  19:12:30
9     (2021, 6, 9, 19, 20, 29, 2, 160, -1)  19:20:29
10     (2021, 6, 9, 19, 29, 0, 2, 160, -1)   19:29:0
11    (2021, 6, 9, 19, 36, 59, 2, 160, -1)  19:36:59
12     (2021, 6, 9, 19, 45, 0, 2, 160, -1)   19:45:0
13    (2021, 6, 9, 19, 52, 58, 2, 160, -1)  19:52:58
14     (2021, 6, 9, 20, 0, 59, 2, 160, -1)   20:0:59
15    (2021, 6, 9, 20, 10, 28, 2, 160, -1)  20:10:28
16    (2021, 6, 9, 20, 19, 30, 2, 160, -1)  20:19:30
17    (2021, 6, 9, 20, 28, 30, 2, 160, -1)  20:28:30
18     (2021, 6, 9, 20, 39, 0, 2, 160, -1)   20:39:0
19     (2021, 6, 9, 20, 47, 2, 2, 160, -1)   20:47:2
20     (2021, 6, 9, 20, 55, 1, 2, 160, -1)   20:55:1
21      (2021, 6, 9, 21, 3, 0, 2, 160, -1)    21:3:0
22     (2021, 6, 9, 21, 11, 0, 2, 160, -1)   21:11:0
23     (2021, 6, 9, 21, 19, 1, 2, 160, -1)   21:19:1
24     (2021, 6, 9, 21, 27, 0, 2, 160, -1)   21:27:0
25    (2021, 6, 9, 21, 35, 29, 2, 160, -1)  21:35:29
26     (2021, 6, 9, 21, 44, 0, 2, 160, -1)   21:44:0
27    (2021, 6, 9, 21, 51, 59, 2, 160, -1)  21:51:59
28     (2021, 6, 9, 21, 59, 0, 2, 160, -1)   21:59:0
29      (2021, 6, 9, 22, 7, 1, 2, 160, -1)    22:7:1
30     (2021, 6, 9, 22, 15, 0, 2, 160, -1)   22:15:0
31     (2021, 6, 9, 22, 23, 0, 2, 160, -1)   22:23:0
32     (2021, 6, 9, 22, 31, 1, 2, 160, -1)   22:31:1
33    (2021, 6, 9, 22, 40, 30, 2, 160, -1)  22:40:30
34     (2021, 6, 9, 22, 51, 0, 2, 160, -1)   22:51:0
35     (2021, 6, 9, 22, 59, 0, 2, 160, -1)   22:59:0
36      (2021, 6, 9, 23, 7, 1, 2, 160, -1)    23:7:1
37    (2021, 6, 9, 23, 13, 58, 2, 160, -1)  23:13:58
38     (2021, 6, 9, 23, 22, 1, 2, 160, -1)   23:22:1
```

```
39     (2021, 6, 9, 23, 30, 0, 2, 160, -1)   23:30:0
40     (2021, 6, 9, 23, 38, 0, 2, 160, -1)   23:38:0
41    (2021, 6, 9, 23, 45, 59, 2, 160, -1)  23:45:59
42     (2021, 6, 9, 23, 54, 1, 2, 160, -1)   23:54:1
43      (2021, 6, 10, 0, 2, 0, 3, 161, -1)     0:2:0
44     (2021, 6, 10, 0, 10, 1, 3, 161, -1)    0:10:1
45     (2021, 6, 10, 0, 18, 0, 3, 161, -1)    0:18:0
46     (2021, 6, 10, 0, 26, 0, 3, 161, -1)    0:26:0
47     (2021, 6, 10, 0, 34, 1, 3, 161, -1)    0:34:1
48     (2021, 6, 10, 0, 42, 1, 3, 161, -1)    0:42:1
49     (2021, 6, 10, 0, 50, 0, 3, 161, -1)    0:50:0
50     (2021, 6, 10, 0, 57, 1, 3, 161, -1)    0:57:1
51     (2021, 6, 10, 1, 5, 31, 3, 161, -1)    1:5:31
52    (2021, 6, 10, 1, 12, 28, 3, 161, -1)   1:12:28
53    (2021, 6, 10, 1, 20, 31, 3, 161, -1)   1:20:31
54     (2021, 6, 10, 1, 29, 0, 3, 161, -1)    1:29:0
55     (2021, 6, 10, 1, 36, 0, 3, 161, -1)    1:36:0
56     (2021, 6, 10, 1, 44, 0, 3, 161, -1)    1:44:0
57    (2021, 6, 10, 1, 51, 32, 3, 161, -1)   1:51:32
58     (2021, 6, 10, 1, 59, 2, 3, 161, -1)    1:59:2
59      (2021, 6, 10, 2, 7, 1, 3, 161, -1)     2:7:1
60     (2021, 6, 10, 2, 15, 1, 3, 161, -1)    2:15:1
61    (2021, 6, 10, 2, 22, 31, 3, 161, -1)   2:22:31
62    (2021, 6, 10, 2, 30, 32, 3, 161, -1)   2:30:32
63    (2021, 6, 10, 2, 38, 31, 3, 161, -1)   2:38:31
64    (2021, 6, 10, 2, 46, 30, 3, 161, -1)   2:46:30
65    (2021, 6, 10, 2, 54, 32, 3, 161, -1)   2:54:32
66     (2021, 6, 10, 3, 2, 31, 3, 161, -1)    3:2:31
67    (2021, 6, 10, 3, 10, 33, 3, 161, -1)   3:10:33
68     (2021, 6, 10, 3, 22, 3, 3, 161, -1)    3:22:3
69    (2021, 6, 10, 3, 32, 33, 3, 161, -1)   3:32:33
70     (2021, 6, 10, 3, 48, 1, 3, 161, -1)    3:48:1
71     (2021, 6, 10, 3, 58, 1, 3, 161, -1)    3:58:1
72     (2021, 6, 10, 4, 9, 30, 3, 161, -1)    4:9:30
73    (2021, 6, 10, 4, 19, 30, 3, 161, -1)   4:19:30
74    (2021, 6, 10, 4, 27, 32, 3, 161, -1)   4:27:32
75     (2021, 6, 10, 4, 36, 0, 3, 161, -1)    4:36:0
76     (2021, 6, 10, 4, 44, 2, 3, 161, -1)    4:44:2
77     (2021, 6, 10, 4, 52, 1, 3, 161, -1)    4:52:1
78      (2021, 6, 10, 5, 0, 1, 3, 161, -1)     5:0:1
79      (2021, 6, 10, 5, 8, 0, 3, 161, -1)     5:8:0
80     (2021, 6, 10, 5, 16, 1, 3, 161, -1)    5:16:1
81    (2021, 6, 10, 5, 24, 30, 3, 161, -1)   5:24:30
82    (2021, 6, 10, 5, 32, 31, 3, 161, -1)   5:32:31
83    (2021, 6, 10, 5, 40, 31, 3, 161, -1)   5:40:31
84    (2021, 6, 10, 5, 48, 32, 3, 161, -1)   5:48:32
85    (2021, 6, 10, 5, 56, 33, 3, 161, -1)   5:56:33
86     (2021, 6, 10, 6, 4, 31, 3, 161, -1)    6:4:31
```

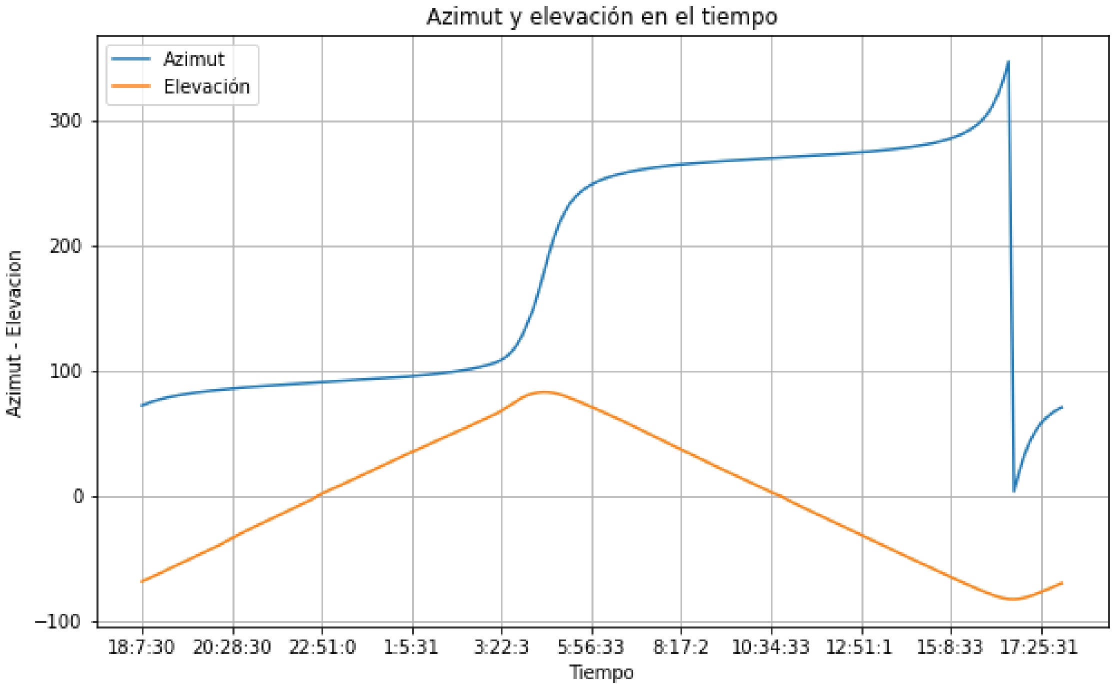
```
87    (2021, 6, 10, 6, 12, 32, 3, 161, -1)   6:12:32
88    (2021, 6, 10, 6, 20, 31, 3, 161, -1)   6:20:31
89    (2021, 6, 10, 6, 28, 31, 3, 161, -1)   6:28:31
90    (2021, 6, 10, 6, 36, 30, 3, 161, -1)   6:36:30
91    (2021, 6, 10, 6, 44, 31, 3, 161, -1)   6:44:31
92    (2021, 6, 10, 6, 53, 31, 3, 161, -1)   6:53:31
93     (2021, 6, 10, 7, 1, 31, 3, 161, -1)    7:1:31
94     (2021, 6, 10, 7, 10, 4, 3, 161, -1)    7:10:4
95    (2021, 6, 10, 7, 18, 30, 3, 161, -1)   7:18:30
96     (2021, 6, 10, 7, 27, 1, 3, 161, -1)    7:27:1
97     (2021, 6, 10, 7, 36, 0, 3, 161, -1)    7:36:0
98    (2021, 6, 10, 7, 44, 31, 3, 161, -1)   7:44:31
99    (2021, 6, 10, 7, 52, 31, 3, 161, -1)   7:52:31
100    (2021, 6, 10, 8, 0, 32, 3, 161, -1)    8:0:32
101     (2021, 6, 10, 8, 9, 1, 3, 161, -1)     8:9:1
102    (2021, 6, 10, 8, 17, 2, 3, 161, -1)    8:17:2
103    (2021, 6, 10, 8, 25, 3, 3, 161, -1)    8:25:3
104    (2021, 6, 10, 8, 33, 1, 3, 161, -1)    8:33:1
105   (2021, 6, 10, 8, 41, 31, 3, 161, -1)   8:41:31
106   (2021, 6, 10, 8, 49, 31, 3, 161, -1)   8:49:31
107    (2021, 6, 10, 8, 58, 1, 3, 161, -1)    8:58:1
108     (2021, 6, 10, 9, 6, 1, 3, 161, -1)     9:6:1
109   (2021, 6, 10, 9, 14, 31, 3, 161, -1)   9:14:31
110   (2021, 6, 10, 9, 22, 31, 3, 161, -1)   9:22:31
111   (2021, 6, 10, 9, 30, 30, 3, 161, -1)   9:30:30
112   (2021, 6, 10, 9, 38, 33, 3, 161, -1)   9:38:33
113   (2021, 6, 10, 9, 46, 31, 3, 161, -1)   9:46:31
114   (2021, 6, 10, 9, 54, 30, 3, 161, -1)   9:54:30
115   (2021, 6, 10, 10, 2, 31, 3, 161, -1)   10:2:31
116  (2021, 6, 10, 10, 10, 31, 3, 161, -1)  10:10:31
117  (2021, 6, 10, 10, 18, 30, 3, 161, -1)  10:18:30
118  (2021, 6, 10, 10, 26, 31, 3, 161, -1)  10:26:31
119  (2021, 6, 10, 10, 34, 33, 3, 161, -1)  10:34:33
120  (2021, 6, 10, 10, 42, 32, 3, 161, -1)  10:42:32
121  (2021, 6, 10, 10, 50, 31, 3, 161, -1)  10:50:31
122  (2021, 6, 10, 10, 58, 31, 3, 161, -1)  10:58:31
123   (2021, 6, 10, 11, 6, 32, 3, 161, -1)   11:6:32
124  (2021, 6, 10, 11, 14, 33, 3, 161, -1)  11:14:33
125  (2021, 6, 10, 11, 22, 33, 3, 161, -1)  11:22:33
126  (2021, 6, 10, 11, 30, 32, 3, 161, -1)  11:30:32
127  (2021, 6, 10, 11, 38, 33, 3, 161, -1)  11:38:33
128  (2021, 6, 10, 11, 46, 31, 3, 161, -1)  11:46:31
129  (2021, 6, 10, 11, 54, 32, 3, 161, -1)  11:54:32
130   (2021, 6, 10, 12, 2, 34, 3, 161, -1)   12:2:34
131  (2021, 6, 10, 12, 10, 33, 3, 161, -1)  12:10:33
132   (2021, 6, 10, 12, 19, 1, 3, 161, -1)   12:19:1
133   (2021, 6, 10, 12, 27, 1, 3, 161, -1)   12:27:1
134   (2021, 6, 10, 12, 35, 2, 3, 161, -1)   12:35:2
```

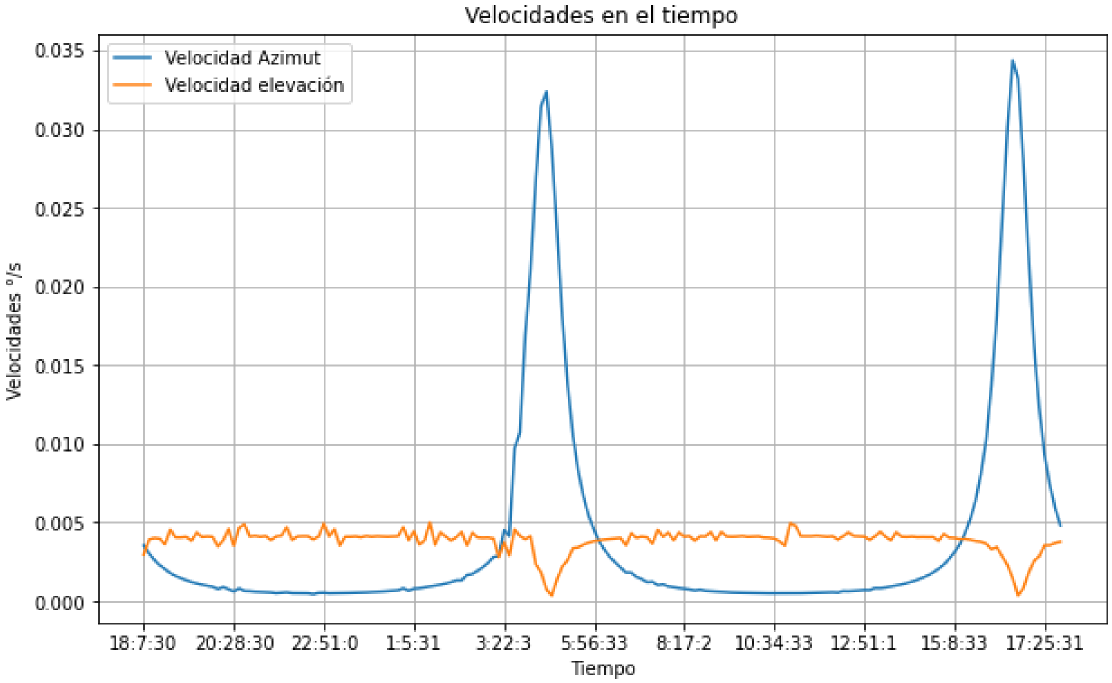
```
135   (2021, 6, 10, 12, 43, 3, 3, 161, -1)   12:43:3
136   (2021, 6, 10, 12, 51, 1, 3, 161, -1)   12:51:1
137   (2021, 6, 10, 12, 59, 2, 3, 161, -1)   12:59:2
138   (2021, 6, 10, 13, 7, 31, 3, 161, -1)   13:7:31
139  (2021, 6, 10, 13, 15, 32, 3, 161, -1)  13:15:32
140  (2021, 6, 10, 13, 23, 31, 3, 161, -1)  13:23:31
141  (2021, 6, 10, 13, 31, 31, 3, 161, -1)  13:31:31
142   (2021, 6, 10, 13, 40, 1, 3, 161, -1)   13:40:1
143   (2021, 6, 10, 13, 48, 1, 3, 161, -1)   13:48:1
144   (2021, 6, 10, 13, 56, 2, 3, 161, -1)   13:56:2
145    (2021, 6, 10, 14, 4, 3, 3, 161, -1)    14:4:3
146   (2021, 6, 10, 14, 12, 1, 3, 161, -1)   14:12:1
147   (2021, 6, 10, 14, 20, 2, 3, 161, -1)   14:20:2
148   (2021, 6, 10, 14, 28, 3, 3, 161, -1)   14:28:3
149   (2021, 6, 10, 14, 36, 3, 3, 161, -1)   14:36:3
150   (2021, 6, 10, 14, 44, 2, 3, 161, -1)   14:44:2
151  (2021, 6, 10, 14, 52, 31, 3, 161, -1)  14:52:31
152   (2021, 6, 10, 15, 0, 34, 3, 161, -1)   15:0:34
153   (2021, 6, 10, 15, 8, 33, 3, 161, -1)   15:8:33
154  (2021, 6, 10, 15, 16, 31, 3, 161, -1)  15:16:31
155  (2021, 6, 10, 15, 24, 34, 3, 161, -1)  15:24:34
156  (2021, 6, 10, 15, 32, 33, 3, 161, -1)  15:32:33
157  (2021, 6, 10, 15, 40, 31, 3, 161, -1)  15:40:31
158  (2021, 6, 10, 15, 48, 34, 3, 161, -1)  15:48:34
159  (2021, 6, 10, 15, 56, 33, 3, 161, -1)  15:56:33
160   (2021, 6, 10, 16, 4, 33, 3, 161, -1)   16:4:33
161   (2021, 6, 10, 16, 13, 1, 3, 161, -1)   16:13:1
162   (2021, 6, 10, 16, 21, 3, 3, 161, -1)   16:21:3
163   (2021, 6, 10, 16, 29, 4, 3, 161, -1)   16:29:4
164   (2021, 6, 10, 16, 37, 4, 3, 161, -1)   16:37:4
165   (2021, 6, 10, 16, 45, 3, 3, 161, -1)   16:45:3
166   (2021, 6, 10, 16, 53, 6, 3, 161, -1)   16:53:6
167    (2021, 6, 10, 17, 1, 3, 3, 161, -1)    17:1:3
168    (2021, 6, 10, 17, 9, 1, 3, 161, -1)    17:9:1
169   (2021, 6, 10, 17, 17, 2, 3, 161, -1)   17:17:2
170  (2021, 6, 10, 17, 25, 31, 3, 161, -1)  17:25:31
171  (2021, 6, 10, 17, 33, 32, 3, 161, -1)  17:33:32
172  (2021, 6, 10, 17, 41, 31, 3, 161, -1)  17:41:31
173  (2021, 6, 10, 17, 49, 31, 3, 161, -1)  17:49:31
```

[17]:

```
#Grafica de posicion
xpos_stick = len(Astro_data["Hora"])
fig, ax = plt.subplots(figsize=(10,6))
azimut_plot = ax.plot(Astro_data["Hora"], Astro_data["azimut"],label="Azimut")
elevacion_plot = ax.plot(Astro_data["Hora"],␣
 ↪Astro_data["elevacion"],label="Elevación")
ax.set_ylabel("Azimut - Elevacion")
ax.set_xlabel("Tiempo")
ax.set_title("Azimut y elevación en el tiempo")
xpos_values = ax.get_xticks()
newx_pos_values = [xpos_values[i] for i in range(0,xpos_stick,xpos_stick//10)]
ax.set_xticks(newx_pos_values)
ax.legend()
ax.grid(b=True)

#Grafica de velocidad
xvel_stick = len(Astronomic_data_vel["Hora"])
fig_vel, ax_vel = plt.subplots(figsize=(10,6))
velazim_plot = ax_vel.
 ↪plot(Astronomic_data_vel["Hora"],Astronomic_data_vel["velocidad␣
 ↪azimut"],label="Velocidad Azimut")
velalt_plot = ax_vel.
 ↪plot(Astronomic_data_vel["Hora"],Astronomic_data_vel["velocidad␣
 ↪altitud"],label ="Velocidad elevación")
ax_vel.set_ylabel("Velocidades °/s")
ax_vel.set_xlabel("Tiempo")
ax_vel.set_title("Velocidades en el tiempo")
xvel_values = ax_vel.get_xticks()
newx_vel_values = [xvel_values[i] for i in range(0,(xvel_stick//2)*2,xvel_stick/
 ↪/10)]
ax_vel.set_xticks(newx_vel_values)
ax_vel.legend()
ax_vel.grid(b=True)
#Astronomic_data_vel.plot()
```





[18]:

```
#Valores promedio, máx y min de velocidad en el eje del azimut
Velocidades_azimut = [
    Astronomic_data_vel["velocidad azimut"].mean(),
    Astronomic_data_vel["velocidad azimut"].max(),
     Astronomic_data_vel["velocidad azimut"].min()
]
Velocidades_elevacion = [
    Astronomic_data_vel["velocidad altitud"].mean(),
    Astronomic_data_vel["velocidad altitud"].max(),
    Astronomic_data_vel["velocidad altitud"].min(),
]
print("Valores promedio, max y min de velocidad en el eje de azimut")
print(Velocidades_azimut)
print("Valores promedio, max y min de velocidad en el eje de elevacion")
print(Velocidades_elevacion)
```

```
Valores promedio, max y min de velocidad en el eje de azimut
[0.0041553673861102315, 0.03436274944721639, 0.00046460801127904143]
Valores promedio, max y min de velocidad en el eje de elevacion
[0.0038514867498842175, 0.005002837900087811, 0.00035735600770381195]
```

[19]:

```
#Se guardan los datos del seguimiento en un archivo .csv

#Datos de posición como vectores numpy
data_azim_csv = Astro_data["azimut"].to_numpy(copy = True)
data_elev_csv = Astro_data["elevacion"].to_numpy(copy= True)
data_date_csv = Astro_data["Date"].to_numpy(copy= True)
data_hour_csv = Astro_data["Hora"].to_numpy(copy= True)

#Datos de velocidad como vectores numpy
data_velazim_csv = Astronomic_data_vel["velocidad azimut"].to_numpy(copy = True)
data_velelev_csv = Astronomic_data_vel["velocidad altitud"].to_numpy(copy =␣
 ↪True)

data_csv = pd.DataFrame(list(
zip(
    data_date_csv,
    data_hour_csv,
    data_azim_csv,
    data_elev_csv,
    data_velazim_csv,
    data_velelev_csv,
)),
columns=[
    "Fecha",
    "Hora",
    "Azimut",
    "Elevación",
    "Velocidad Azimut",
    "Velocidad Elevación"
])
#Cambiar el data por el nombre de la estrella que se siguió

data_csv.to_csv('16_Sadalmelik.csv',sep = ',',index = False)
```