

## DRAGON BALL. EPISODIO 1

Célula, desde un universo paralelo, ha vuelto al futuro de nuestros héroes en búsqueda de su cuerpo perfecto, para ello necesita una información que sólo una persona es capaz de proporcionarle. En la búsqueda de la información, y tras un enfrentamientos con uno de nuestros héroes, Célula consigue escapar con vida jurando que volvería con el cuerpo perfecto y eliminaría la Tierra por completo.

Debes ayudar a los héroes de la Tierra con el fin de evitar que Célula consiga su objetivo. Llegan a la conclusión de que han de encontrar a la persona buscada por Célula antes que éste. Para ello, y con la ayuda del radar de Bulma, deciden ir en busca de las bolas de dragón para conocer quién es el objetivo de Célula a través de Shenron. ¿Serás capaz de conseguir el nombre?

**\*\* Es necesario que deis acceso a vuestra ubicación para que funcione correctamente \*\***

Servicio contra el que comprobar el nombre:

34.253.120.147:9999

Radar de Bulma:

<https://34.253.120.147/dragonball/episodio1/>

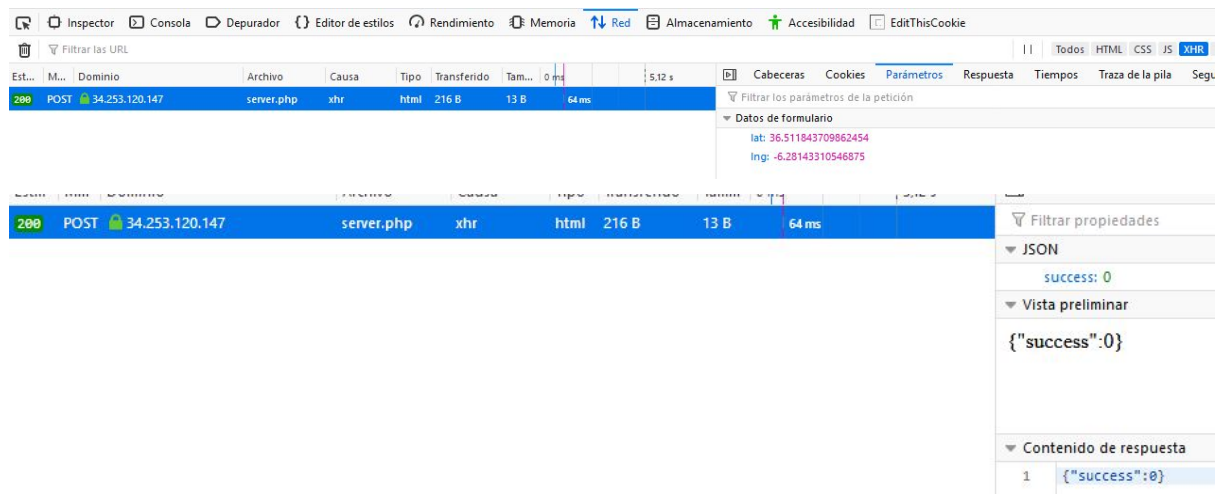
Info: La flag tiene el formato UAM{md5}

### Resolución

Accedemos a la página del radar, nos solicita acceso a nuestra ubicación, utilizamos el complemento para Firefox, "*Location Guard*", que nos permite fijar nuestra ubicación a la que deseemos. Tras probar, obtenemos

**Te encuentras en 36.511843709862454 / -6.28143310546875. No estás cerca de ninguna bola de dragón.**

Desde las herramientas de desarrollador, observamos las diferentes peticiones que se realizan, entre ellas, destaca la llamada a **server.php**, en la que se le pasan como parámetros la latitud y longitud de nuestra posición, y como resultado, un JSON indicando **{"success":0}**.



Ya tenemos el servicio web que nos permite comprobar si la ubicación es correcta. Como punto de partida, buscamos un listado de ciudades del mundo, para realizar las peticiones y ver si obtenemos alguna “bola de dragón”

Descargamos desde <https://simplemaps.com/data/world-cities> un listado csv, con unas 12.00 ciudades. (worldcities.csv)

Creamos un pequeño script en python que recorra el listado de ciudades del csv y que nos muestre el resultado en caso de ser diferente a `{\"sucess\":0 }`.

Formato worldcities.csv

`city,\"city_ascii\",\"lat\",\"lng\",\"country\",\"iso2\",\"iso3\",\"admin_name\",\"capital\",\"population\",\"id\"`

`MalishevÃ«,\"Malisheve\",\"42.4822\",\"20.7458\",\"Kosovo\",\"XK\",\"XKS\",\"MalishevÃ«\",\"admin\", \"\", \"1901597212\"`

Antes de procesar eliminamos "

`city,city_ascii,lat,lng,country,iso2,iso3,admin_name,capital,population,id`

`MalishevÃ«,Malisheve,42.4822,20.7458,Kosovo,XK,XKS,MalishevÃ«,admin,,1901597212`

```

import urllib3
import requests
import csv

urllib3.disable_warnings()
URL="https://34.253.120.147/dragonball/episodio1/server.php"
err={'"success":0}'

f = open ("ok.txt", "a")
f1 = open ("proc.txt", "a")
with open('worldcities.csv','rb') as csvfile:
    fr = csv.reader(csvfile, delimiter=',', quotechar='"')
    for row in fr:
        DATA = {'lat':row[2],'lng':row[3]}
        r = requests.post(url=URL,data=DATA,verify=False)
        f1.write(row[0] + ';' + row[2] + ';' + row[3] + "\n")
        the_page=r.text
        if the_page <> err:
            print(the_page,row[0],DATA)
            f.write(the_page + ';' + row[0] + ';' + row[2] + ';' + row[3] + "\n")
            f.flush()
f.close()
f1.close()

```

Tras unos minutos, empiezan a aparecer resultados:

```

u{"stars":1,"city":"Damasco","lat":33.513645,"lng":36.276762,"locInRadar":"<circle
cx=\\\"150\\\" cy=\\\"150\\\" r=\\\"10\\\"><\\Vcircle>"}', {'lat': '33.8720', 'lng': '35.5097'})

```

```

u{"stars":7,"city":"Odessa","lat":46.482921,"lng":30.722892,"locInRadar":"<circle cx=\\\"30\\\"
cy=\\\"280\\\" r=\\\"10\\\"><\\Vcircle>"}', {'lat': '46.5129', 'lng': '29.6619'})

```

```

(u"", ""Ulaanbaatar", u{"stars":4,"city":"UlanBator","lat":47.906641,"lng":106.895085,"locInRad
ar":"<circle cx=\\\"50\\\" cy=\\\"240\\\" r=\\\"10\\\"><\\Vcircle>"}', {'lat': '47.9167', 'lng': '106.9166'})

```

```

(u"", ""Stockholm", u{"stars":5,"city":"Estocolmo","lat":59.328694,"lng":18.068505,"locInRadar
":"<circle cx=\\\"320\\\" cy=\\\"270\\\" r=\\\"10\\\"><\\Vcircle>"}', {'lat': '59.3508', 'lng': '18.0973'})

```

Con estos datos ya podemos empezar a buscar personajes de Dragon Ball (7 letras)

[link](#)

Tomando la primera letra de cada ciudad, ordenando por número de bola:

```

D X X U E X O
1 2 3 4 5 6 7

```

Tenemos a DR.GERO como candidato, no existen ciudades que empiezan por ., por lo que podemos deducir que se ha quitado y añadido una "U"

DRGUERO

Probamos:

```
echo "DRGUERO" | nc 34.253.120.147 9999
```

**UAM{2f3c45a7fdd272de9f43836e5ca2f39c}**

Tras unas horas, obtenemos el resto de bolas, confirmando nuestra suposición.

```
(u"", "Marbella', u'{"stars":2, "city": "Ronda", "lat":36.745473, "lng":-5.161438, "locInRadar": "<circle cx=\\\"250\\\" cy=\\\"125\\\" r=\\\"10\\\"><\\\"/circle>"}', {"lat": '36.5166', 'lng': '-4.8833'})
```

```
(u"", "Reykjav\x3\xadk', u'{"stars":6, "city": "Reikiavik", "lat":64.145144, "lng":-21.942496, "locInRadar": "<circle cx=\\\"80\\\" cy=\\\"80\\\" r=\\\"10\\\"><\\\"/circle>"}', {"lat": '64.1500', 'lng': '-21.9500'})
```

```
(u"", "Hag\x3\xa5t\x3\xb1a', u'{"stars":3, "city": "Guam", "lat":13.440439, "lng":144.779184, "locInRadar": "<circle cx=\\\"125\\\" cy=\\\"270\\\" r=\\\"10\\\"><\\\"/circle>"}', {"lat": '13.4745', 'lng': '144.7504'})
```

@bicacaro