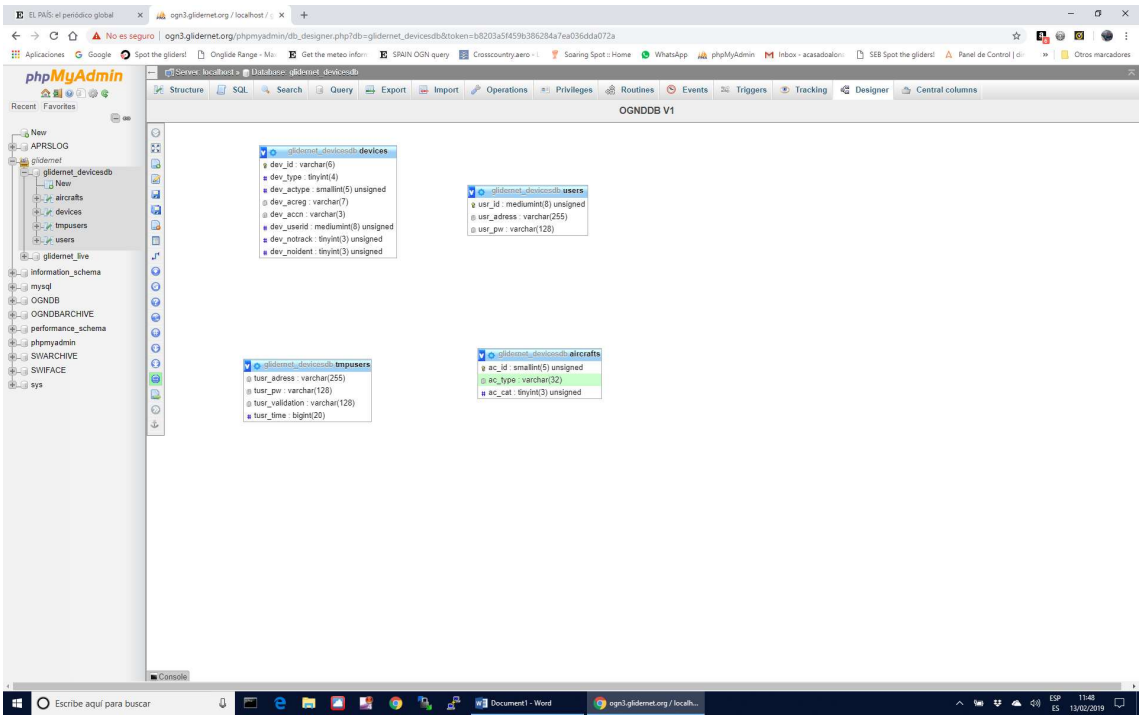
OGN Device Data Base (DDB) V2 proposal

Background:

The current OGN DDB contains the registration data of all the devices that can be handled by the OGN.

As version 1 the DB it contains 4 tables:

- *Devices*: that contains the registration information
- *Aircrafts*: is a table with all the possible aircraft types
- *Users*: with information about the registered users
- *TmpUsers*: with information about users that still did not confirm the email.



Rational for the change:

When the OGN DDB was designed, it was with the intention of register mostly Flarms and OGN trackers, under the assumption that only one device was in one plane. It was an alternate of FlarmNET.

Nowadays, we have a plethora of devices that pilots carry on their aircraft or paraglider, for example: Flarms, OGN trackers, Spider, Spot, InReach, Captur, Naviter Oudie, Flymaster, FANET devices, tablets with XCsoar, smart phones with apps like Naviter Navigator or XC Guide, etc., …

The OGN is just not longer only for gliders and glider pilots, it is been used now more and more by paragliders, helicopters, tow planes, drones, etc.

Also there are many requests to be used by a plethora of virtual radars in conjunction with the popularity of drones.

And in many cases, the pilots can carry more than one device on board, but in that case, we do not want to show on the web apps as two or three aircrafts on the web map.

Proposal (V2.01):

What is proposed in this new version of the OGNDDB, it is to decouple the information about the device information itself of the about the aircraft or flying object (glider, paraglider, helicopter, etc., …).

So on the new design of the database, we will have 6 tables:

- DeviceTypes: a small table making the correspondence for the device type names: Flarms, OGNtrackers, SPOT, Naviter, InReach, Spider, Flymaster, etc. …, and their assigned unique device type. It also contains information about how long is the ID for this kind of device, the one char code and the 3 char code used on APRS.

- Devices: that contains the device registration information, but only info about the device, but with a link where this device is being carry on. We are adding the option of this device to be

active or not. When the aircraft/tracked object is deleted, all the devices associated to that aircraft are deleted. We expanded the size of this field to be 36 chars, so we can accomplish other kink of device type like SPOT, Spider, … also now each device type has its own address space.

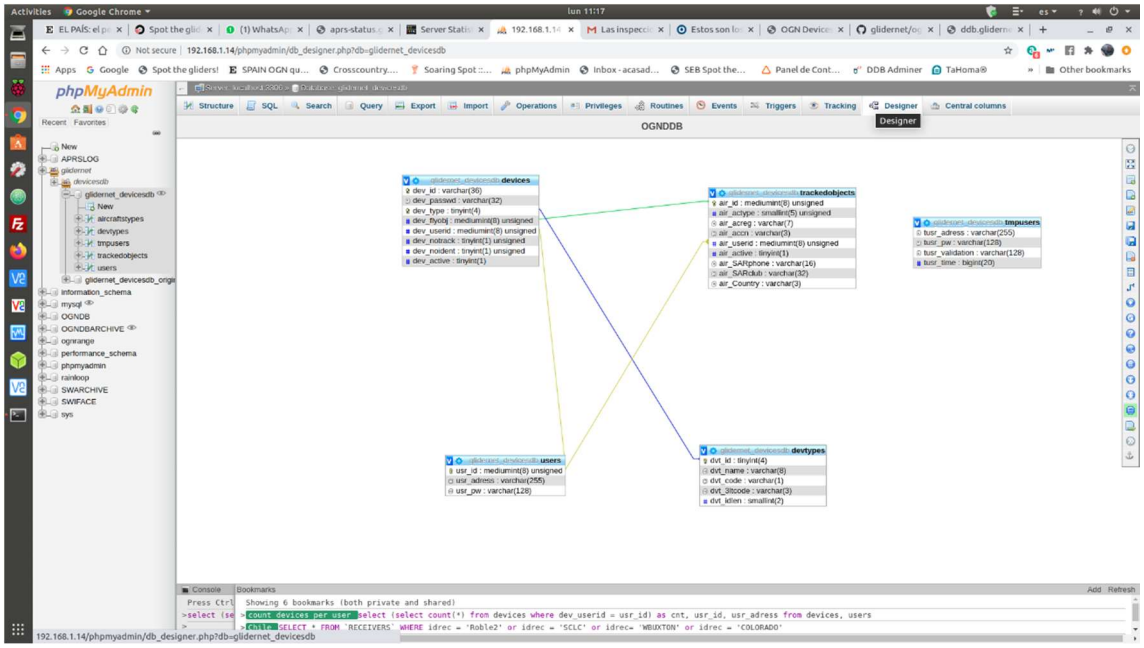
- AircraftsTypes: is a table with all the possible aircraft types. (same except of change the name to be: AircraftTypes).

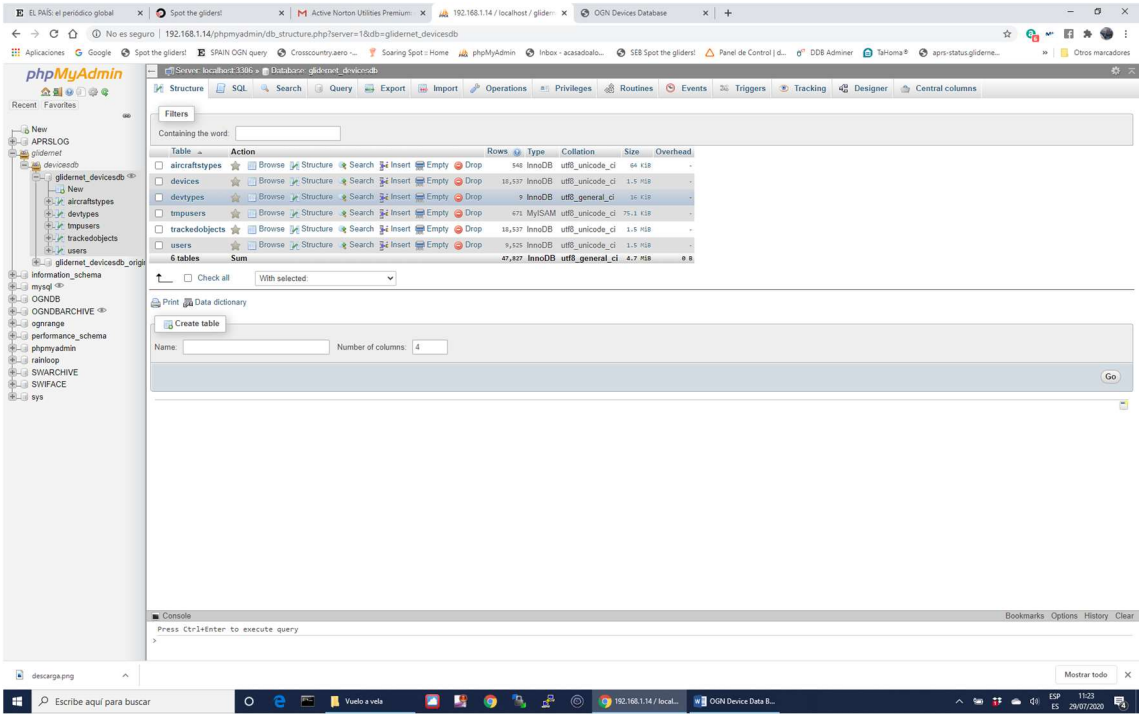
- Users: with information about the registered users (same).

- TmpUsers: with information about users that still did not confirm the email (same). We need to have a script to delete the entries that are one month old.

- TrackedObjects: with information about the aircraft like registration, competition ID, aircraft type, etc. … We are adding information that can be helpful in case of SAR like: phone, club and country. This information is optional. We are adding the option of this TrackedObject to be active or not.

See below the new design and the SQL data.





Migration and API compatible:

Once that the new software has been tested, we can define a cut day and migrate with a specific utility the current data to the new format.

We have a utility written in Python that does the migration of the structure and data from the old design DB to the new design DB.

In terms of the current API, we will maintain the compatibility, however we will extend it in order to gather the new data, perhaps restricting the new data to the JSON format for easier handling.

We have written a utility in order to analyze the consistency of the database.

We are entertaining thoughts of having a new API in order to allow to authorized developers to add dynamically devices and aircrafts/tracked objects from their own programs.

API-Endpoints

/download

```
#DEVICE_TYPE,DEVICE_ID,AIRCRAFT_MODEL,REGISTRATION,CN,TRACKED,IDENTIFIED
'F','0123BC','LS-4','X-0123','23','Y','Y'
'F','DEADBE','DR-400','X-EABC','','N','N'
```

URL parameters

parameter	values	default	effect
t	0/1	0	show field `aircraft type` if set to 1
j	0/1	0	forces JSON output when set to 1 (regardless of accept h
device_id	csv	n/a	select a comma separated list of device ID's
from_id	id	n/a	select list of device ID's starting from this provided ID
till_id	id	n/a	select list of device ID's until this provided ID
registration	csv	n/a	select a comma separated list of registrations
cn	csv	n/a	select a comma separated list of call signs (comp ID)

/download/?j=1

This returns all devices of the **DDB in JSON**. The output validates against the ogn-ddb-schema-1.0.0. With **J=2** also returns the DeviceTypes table

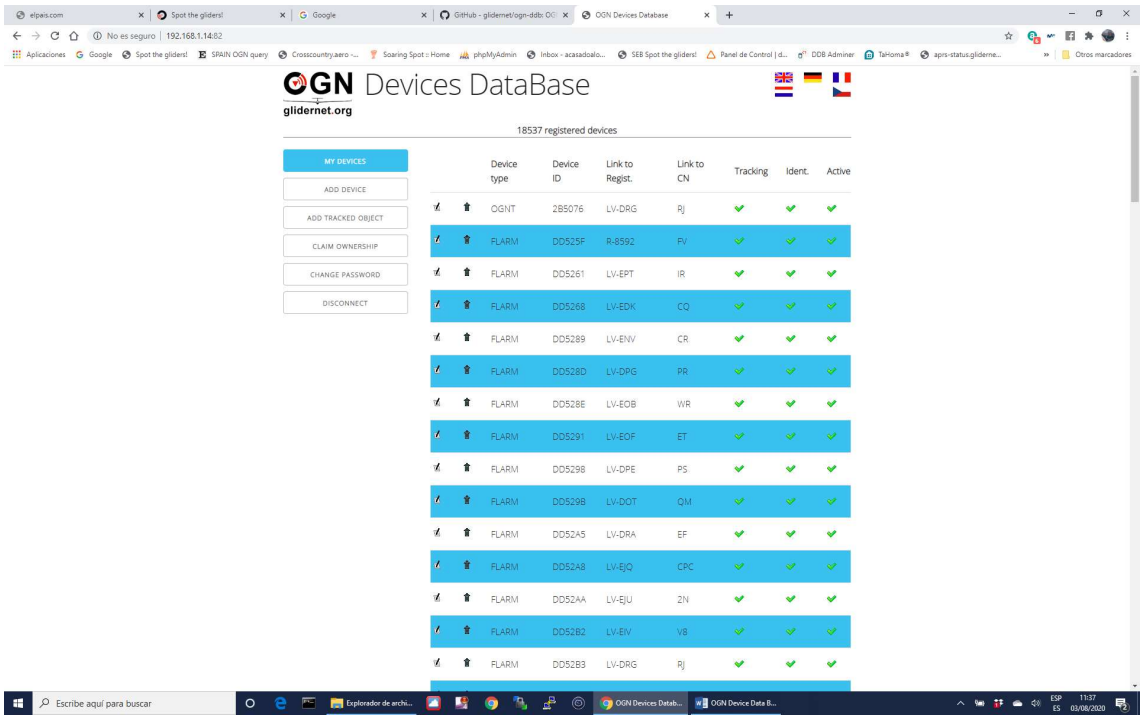
/download/download-fln.php

This returns the device database in a flarmnet-compatible format.

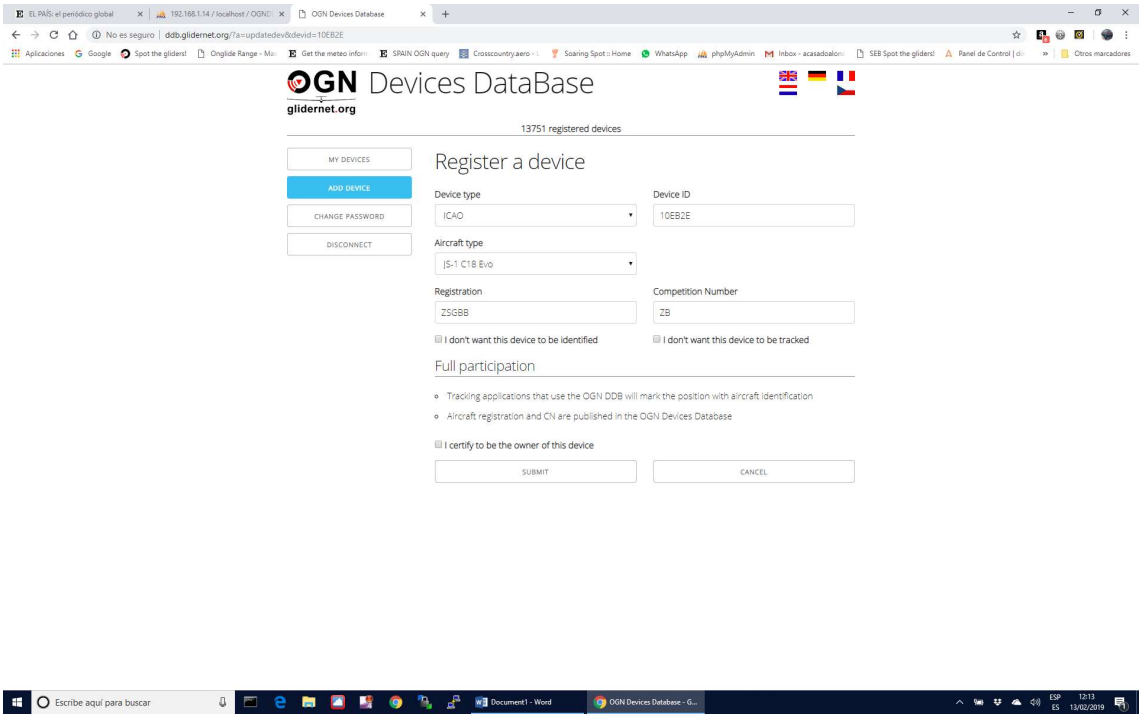
GitHub repository

The GitHub repo is at: https://github.com/glidernet/ogn-ddb

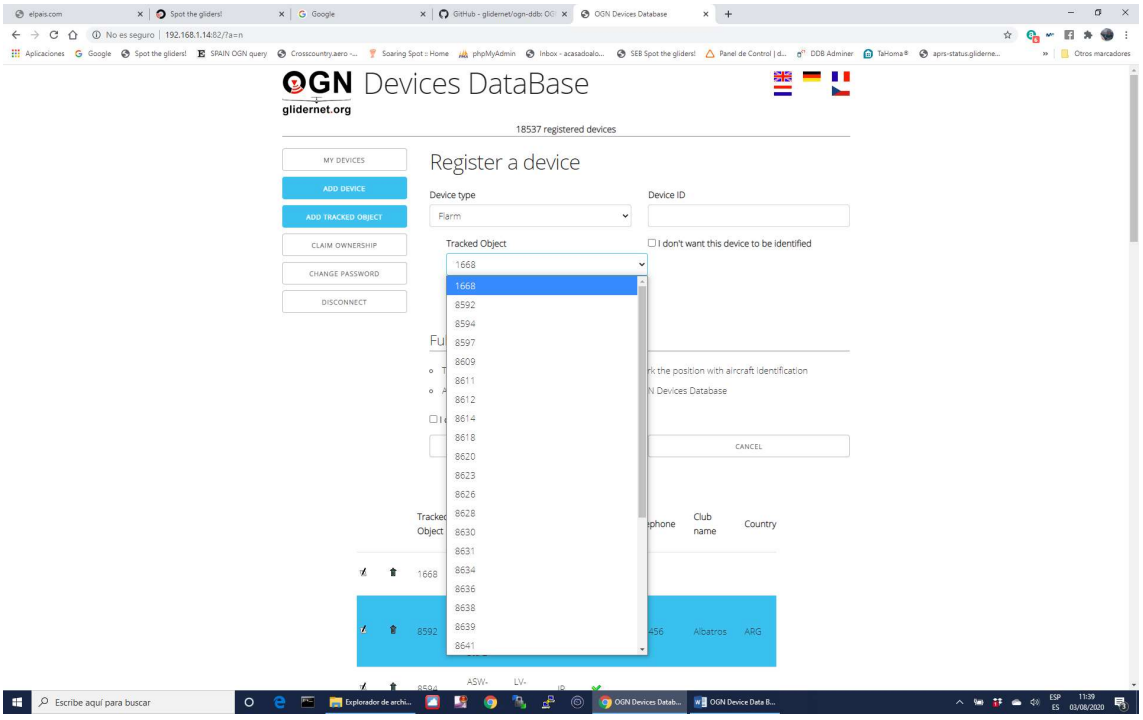
WEB presentation.



On the current presentation, we need to delete the aircraft data from the device information page and move it to a new push down page (**ADD AIRCRAFT**) similar to the current **ADD DEVICE**



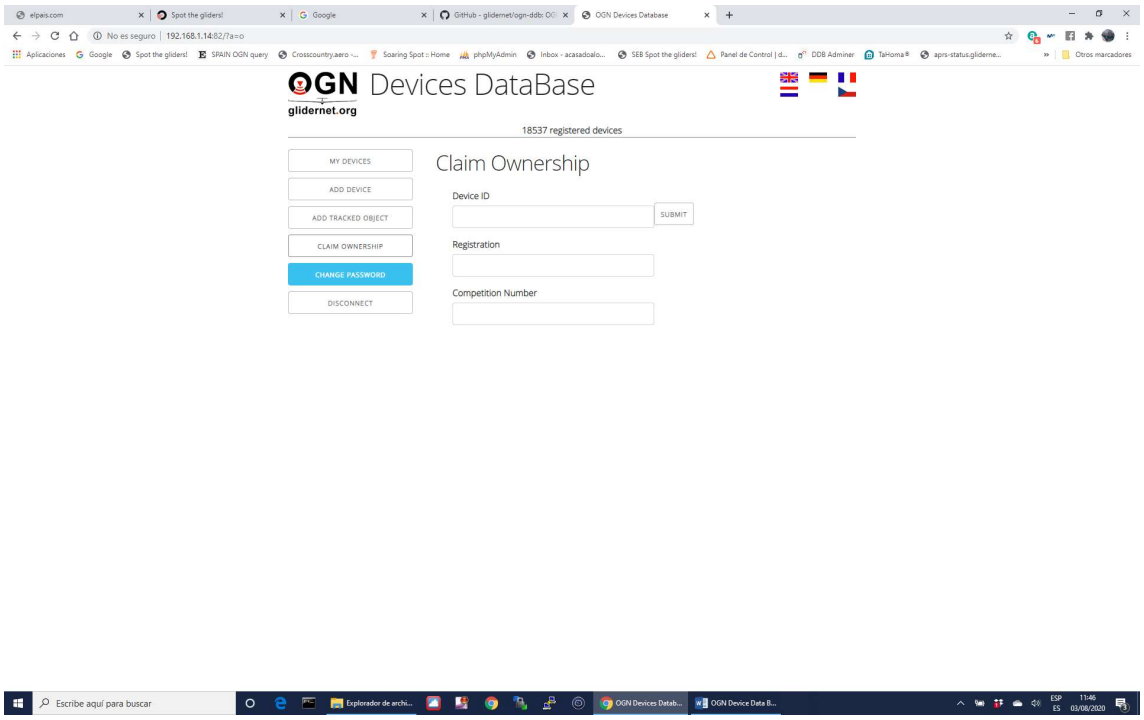
On the new screen we link the device with the tracked object in which list is showed on the low part of the screen.



OGNDDB New feature:

We are adding a new feature to claim the ownership of a device, that is the situation where a person that registered a device, sell the device/aircraft and forgets to de register the device on the OGNDDB, so the new feature CLAIM OWNERSHIP allows to send an email (hidden to the person claiming the ownership) to the person that registered that device asking to de register.

The person claiming the ownership should know: the device id, the registration and the competition ID



OGNDDB Tables schema:

OGNDDB.pdf

OGNDDB SQL

database_schema_v2.0.sql