Deliverable No.	7.5		
Deliverable Title	QGIS plug-in for execution and visualization of marine pollutant dispersion model: LI4MOHID		
Grant agreement number	MYCOAST EAPA_285/2016		
Status	Draft		
Dissemination level	RE-Restricted		
Due date deliverable	5-2020	**Submission date**	15th/04/2020

INDEX

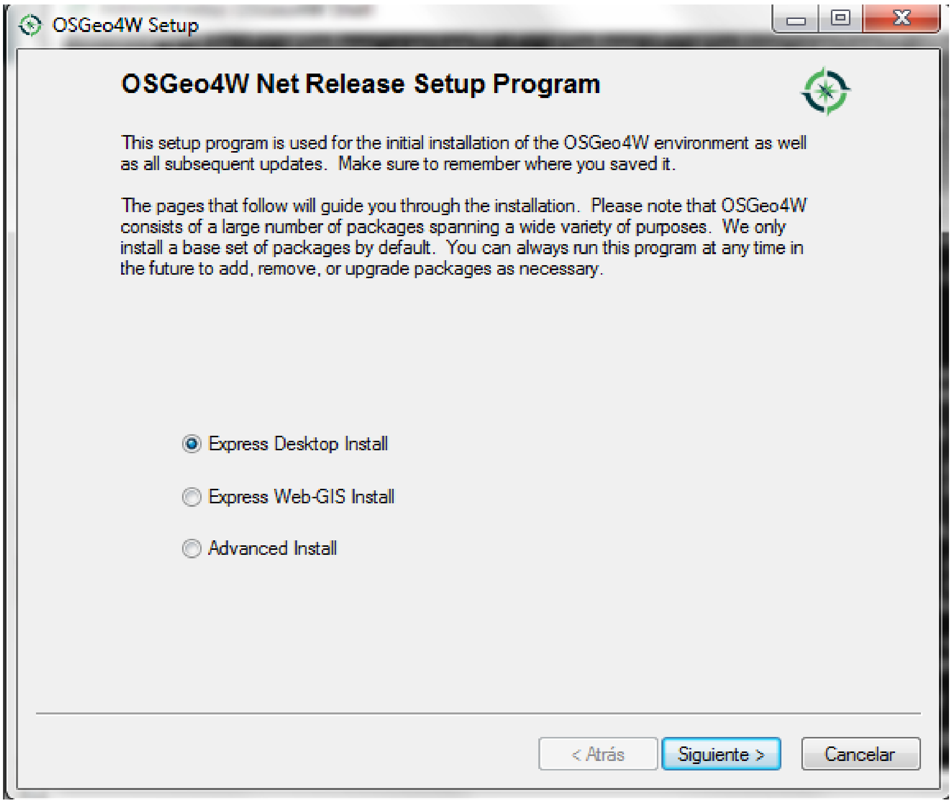
1. SETUP

A plug-in, LI4MOHID, for the open-source geographic information system, QGIS (qgis.org) has been developed in order to run and visualize the results of a marine pollutant dispersion model. The Lagrangian module of MOHID hydrodynamic model will be used, and more specifically the version developed by Canelas and Garaboa[1], which among many other things, optimizes the CPU time of the program.

In order to install the plug-in, we recommend having a version of QGIS 3.10 LRT or higher installed. In Windows operating system, it is necessary to install it from OSGeo4W.

To install the program, visit https://www.qgis.org/es/site/forusers/download.html and choose the option to download it from OSGeo4W.

When executing the "setup" of the OSGeo4W, follow these steps:



[1] http://www.mohid.com/PublicData/Products/ConferencePapers/MOHIDING_2019/12_RCanelas_DGaraboa_BENTLEY_USC_MOHID_Lagrancian_Introduction_to_version_0p3.pdf

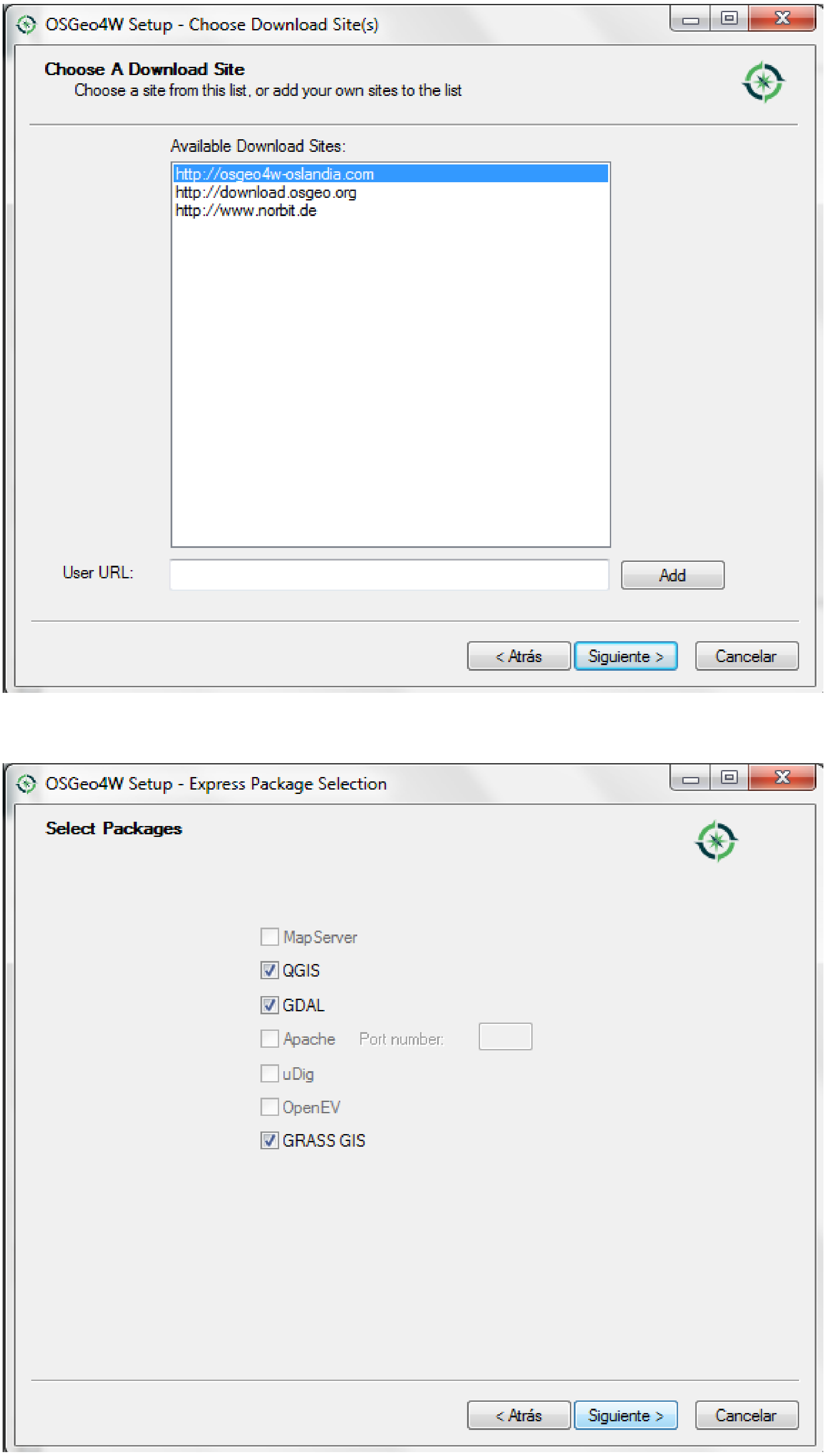


Figure 1.1.- QGIS setup from OSGeo4W.

Once the packages to be installed have been selected, they will be downloaded and installed automatically. This task will take several minutes (depending on the network download rate).
Once finished, and without obtaining any warning message, the QGIS will be installed on our system.

The next step is installation of the plug-in in the QGIS:
Firstly, we need some libraries as NetCDF4 and vtk (both are needed for reading of input data and for results visualization). To install them we recommend the following steps:

1) Open OSGeo4W Shell
2) Path update, using "py3_env".
3) Install "pip" command:
    1) Download "get_pip.py" from https://bootstrap.pypa.io
    2) Put the file in OSGeo4W folder and execute from Shell (python get_pip.py). This program will download and install pip, setup tools and wheel.
    3) Update the paths, again, with "py3_env".
4) In https://www.lfd.uci.edu/~gohlke/pythonlibs/
   download:
    1) cftime-1.0.4-cp37-cp37m-win_amd64.whl
    2) netCDF4-1.5.3-cp37-cp37m-win_amd64.whl
    3) VTK-8.2.0-cp37-cp37m-win_amd64.whl
5) Install these three packages with pip install package.whl

Once NetCDF and VTK libraries are installed, we can install the plug-in.

Unzip "test.tar.gz" file, sent with this document, in C:\OSGeo4W64\apps\qgis\python\plugins folder. In QGIS program, in tab "Complements/Manage and install plugins" looking for "test" and activate. Close the tab; look for an "oil-spill" icon in the toolbar. If you have it, plug-in is installed.

2. USER'S MANUAL

To use the plug-in, click on "spill oil" image located in toolbar.
Define the work folder. In this folder all the results will be saved. At the right-side will appear all the options to execute the Lagrangian model:

- First one, select the grid Lagrangian model (MOHID) and, if we need it, the wind model (WRF). Click on "Apply" and two layers will appear. A point layer and another one with the "bounding box" of Lagrangian model (Fig 2.1)

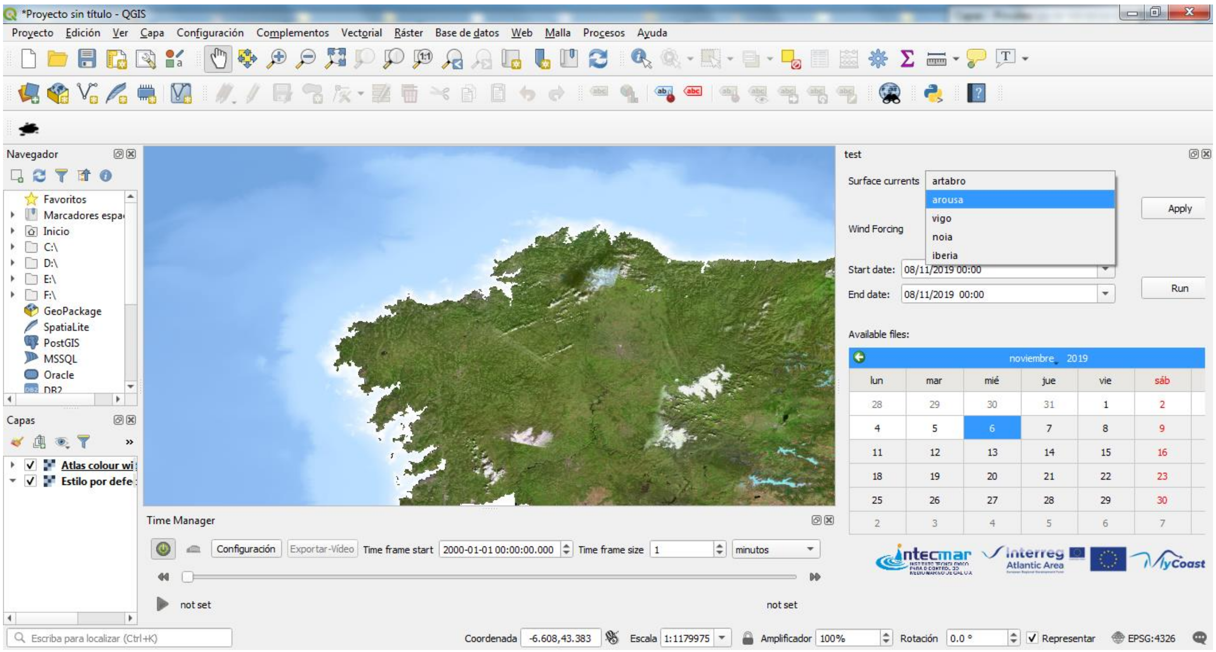


Figure 2.1.- Lagrangian model grid selection

- Select spill points.

  First, select the initial and final date of our simulation, in order to select the currents and wind data and downloaded it.

  To fix the spill point (or points), we must select the "Input points" layer, click on enable editing and "Add a point". Appear a peephole cursor to select the location. After this click, a form appears to fill in the data of the spill point (Figure 2.2)

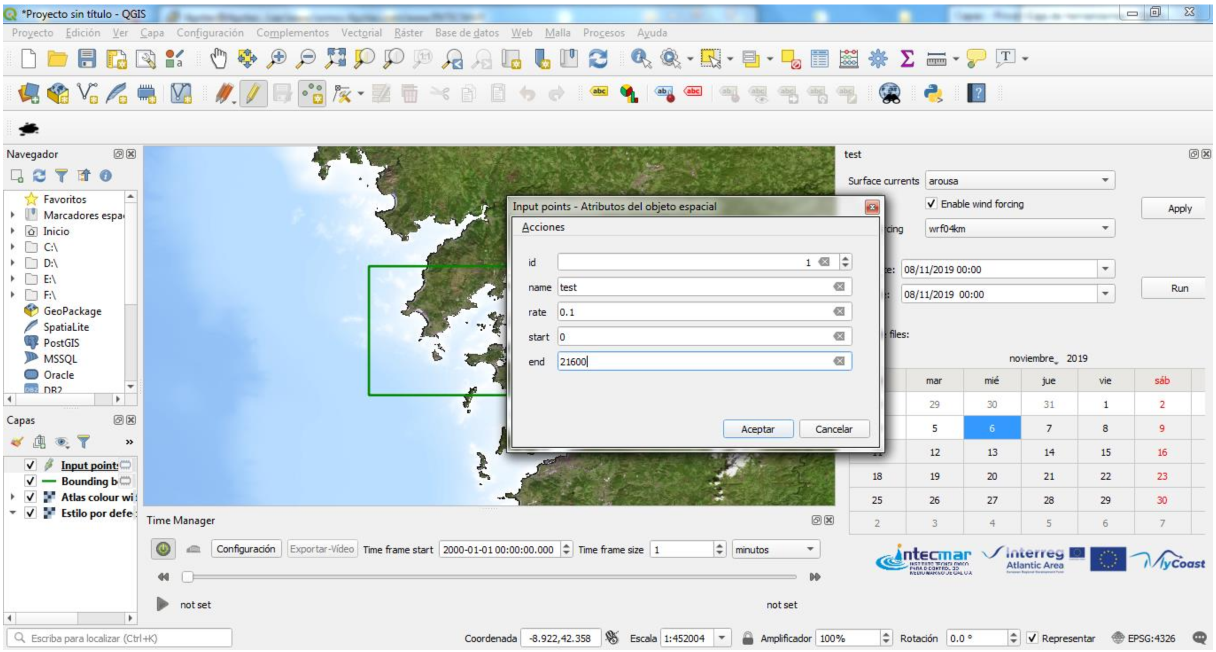


Figure 2.2.- Form to characterize the spill point

- Spill point characteristics.
  - Id and name to identify the spill
  - “Emission rate” (number of particles per second. 0.1 will be a typical value. Lagrangian model time step is 20 minutes, obtaining 120 particles per time step)
  - Start and end of the spill. For example, if the simulation starts on October 28 and the spill occurred at 00h that day and it was pouring fuel until 06h, we will select 0 as the start value and 21600 as the end value. If we wanted it to be a continuous broadcast, we would put the last value in seconds of the simulation (86400 if it were a simulation day). On the contrary, if we want to simulate a punctual discharge at 06h, we will define start as 21600 and end as 22800 (21600 + 1200) and we would include all the spilled fuel in the variable "rate".

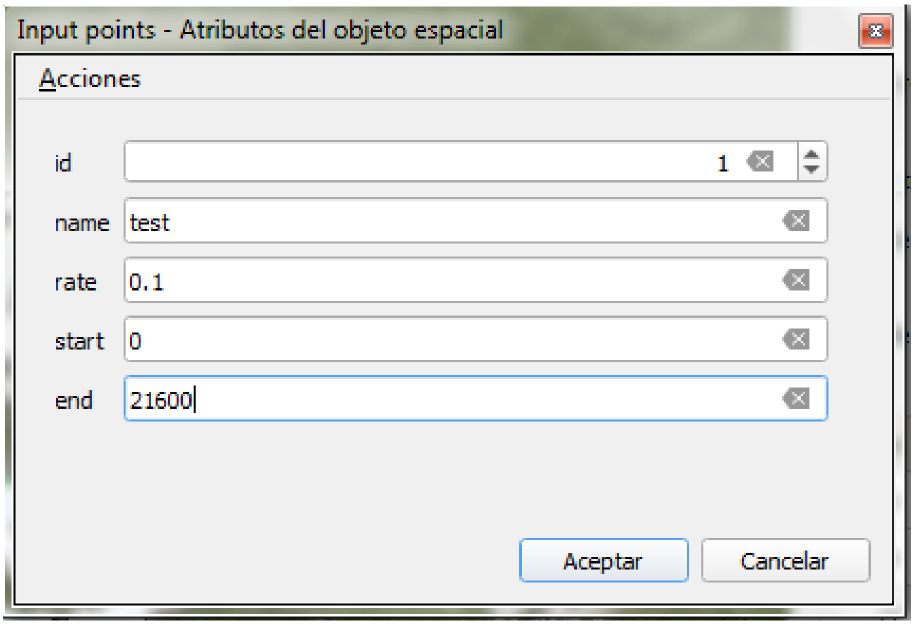


Figure 2.3- Atributos con los que definimos el punto de vertido

Once all the attributes of the spill point have been defined, we can define another spill point, which does not have to coincide at all with the first one. That is, it can occur at another time, it can end at a different time than the first, and it may even have another emission rate. At the end of recording each spill point, it will appear marked on our layer (Figure 2.4)

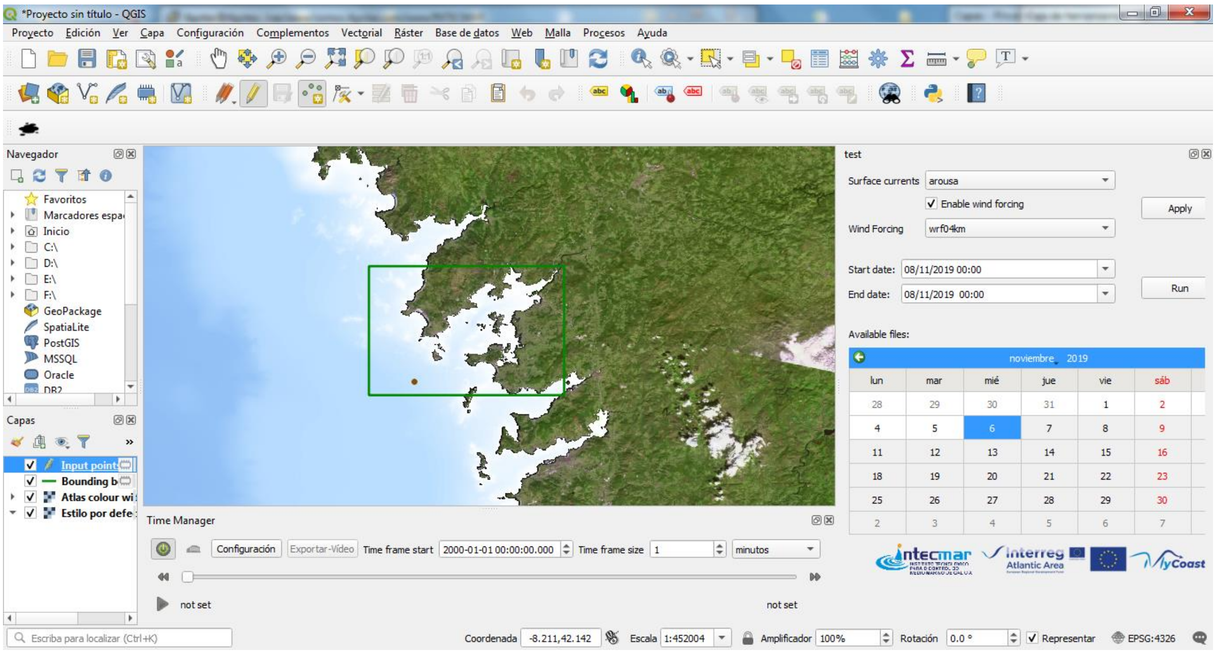


Figure 2.4- Spill point situation

- Model Run.

  Once spill points were defined, click on the "Run" button. All necessary data for numerical models will be downloaded with this click (If dates selected involve more than one day, the first 24 hours of each day simulation will be used to complete all the selected period). These files will be stored in "nc_fields" folder. At the same time, all XML files will be created in order to use, as entry files, for the Lagrangian model.

  In "Data" folder we have 2 files

  - "NamesLibrary,xml" where name of variables were defined
  - "OutputFields.xml" with output information of each particle.

  In the execution folder we have another two XML files with the grid name where inputs are defined (start and end date, name and location of numerical models) and other simulation variables as horizontal diffusion coefficient, or beached coefficient.

  Location of executable Lagrangian model will be requested. Once confirmed, the simulation starts.

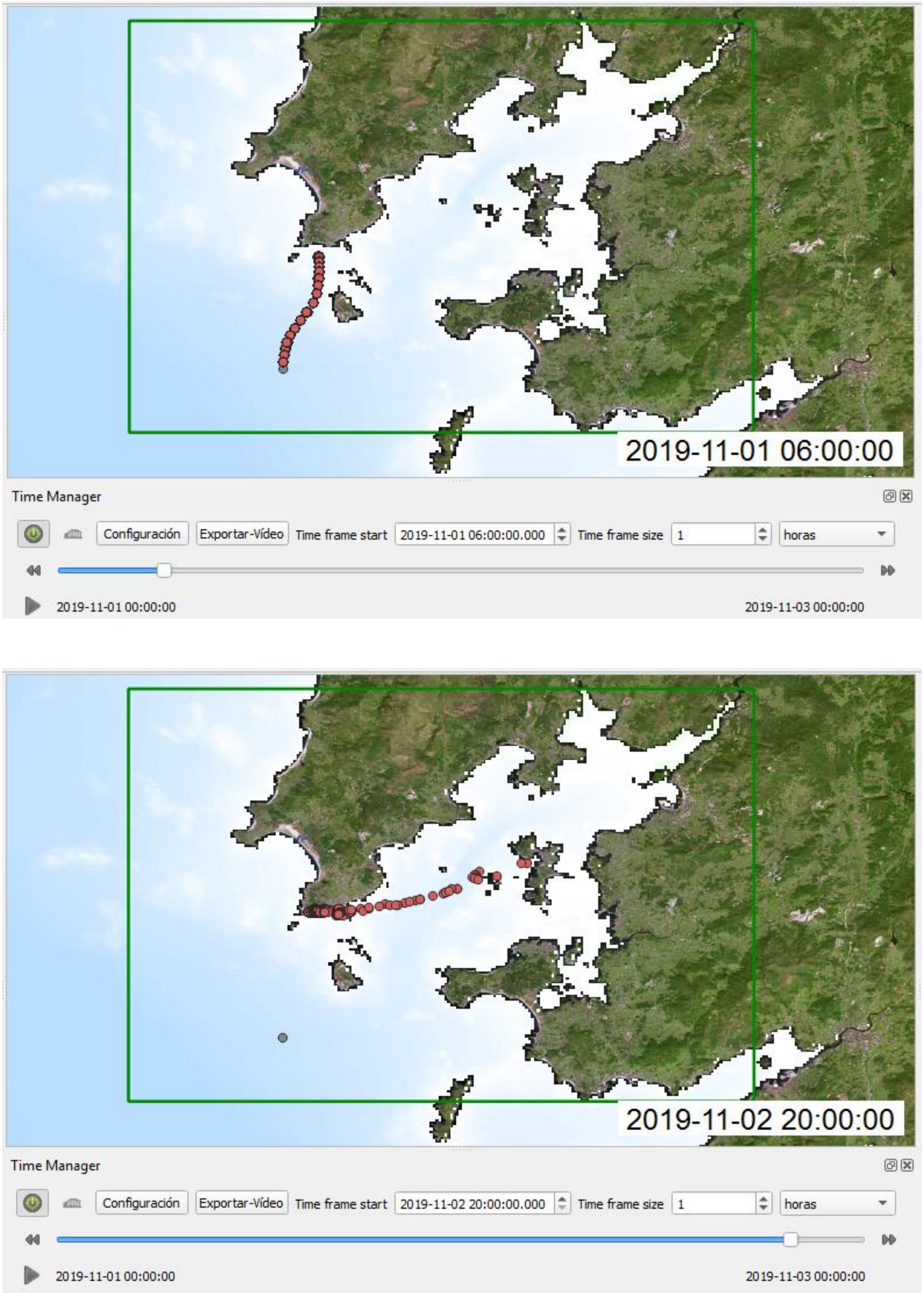


Figure 2.5- Results example

- Results visualization

When the model finishes, a temporal layer with the simulation results will be created. This temporal layer will be obtained with *.vtu files (model output) recorded in the work folder.

To visualize the results, install the "Time Manager" plug-in. In "Plug-in/Manage and install plug-ins" look for "Time Manager" and install it.

Once "Time Manager" has installed, a scroll will appear in the bottom of our project. Click on "Configuration", select "resultado" layer and create the animation based in "time" variable in order to obtain images as Figure 2.5.