#siglibre 2022

Universitat de Girona
Servei de Sistemes d'Informació Geogràfica i Teledetecció

Complemento de QGIS para el proceso de datos ráster en remoto

Daniel Ponsa Mussarra, Felipe Lumbreras Ruiz, Robert Benavente Vidal

(Centre de Visió per Computador / Dept. Ciències de la computació UAB)

Jornadas de SIG Libre, 8 y 9 de junio de 2022 | Girona

Índice

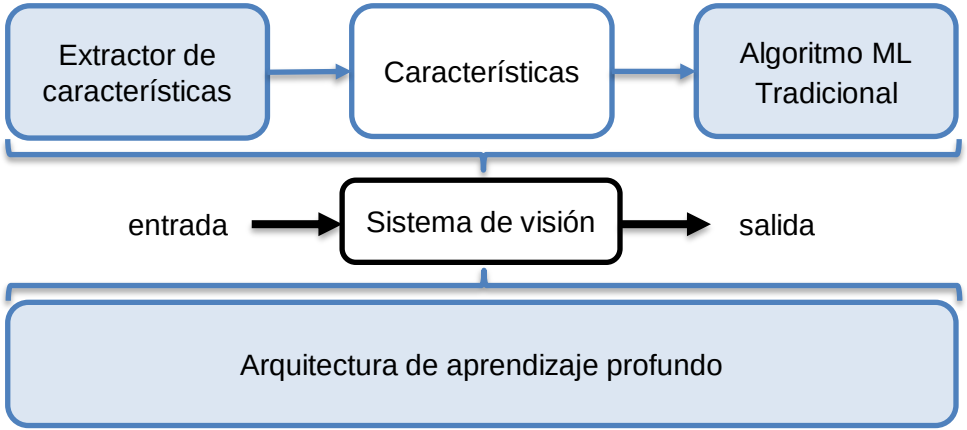
- Contexto inicial :
  - necesidades
  - opciones existentes

- Sistema propuesto
  - Complemento QGIS
  - Proveedor de micro-servicios en la nube
  - Detalles de implementación

- Conclusiones
  - Funcionalidades conseguidas
  - Líneas de mejora

Contexto

- En el grupo MSIAU (MultiSpectral Image Analysis and Understanding) exploramos el uso de información espectral complementaria al espectro visible para desarrollar sistemas avanzados de visión artificial.

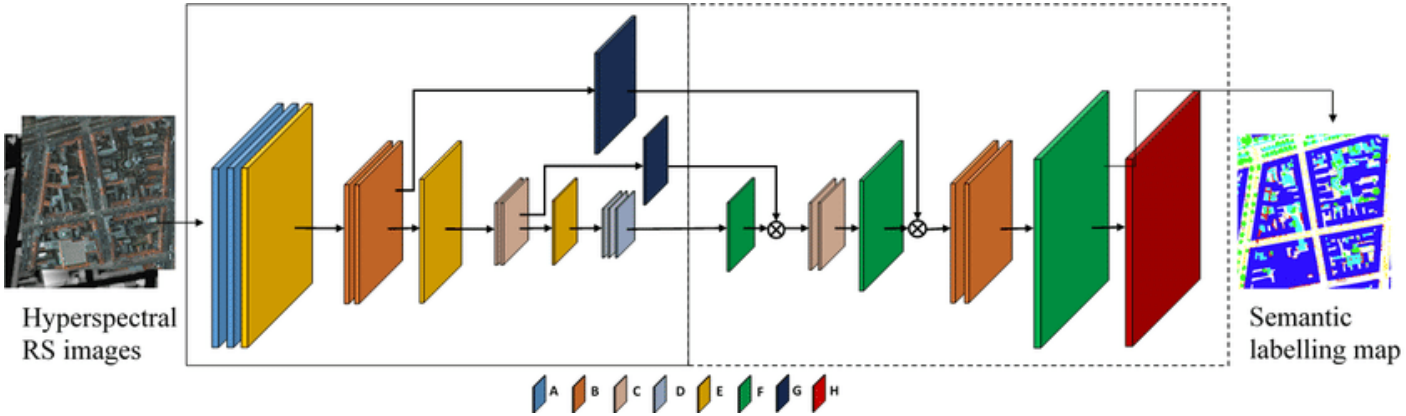


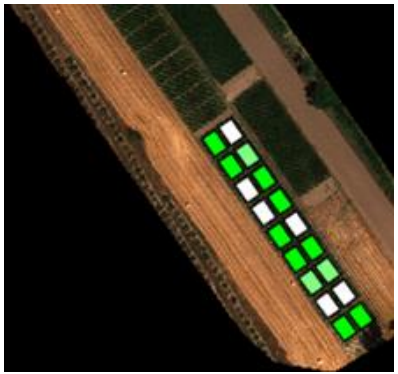
Sistema diseñado o *artesanal*
Flujo de aprendizaje computacional clásico
Modelos con decenas de parámetros

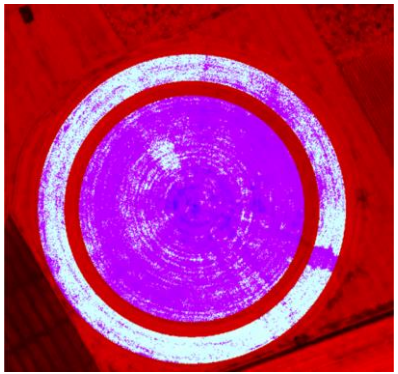
Sistema completamente aprendido
Flujo de aprendizaje profundo
Modelos con decenas de miles de parámetros
Necesidad de datos masivos y cálculo paralelo

- Cooperaciones recientes han centrado parte de nuestro trabajo en el análisis de imágenes aéreas georeferenciadas, en distintos ámbitos.

Temas de investigación: Segmentación semántica

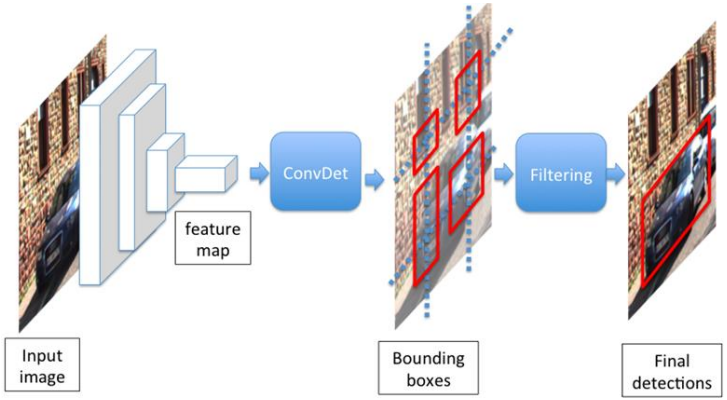


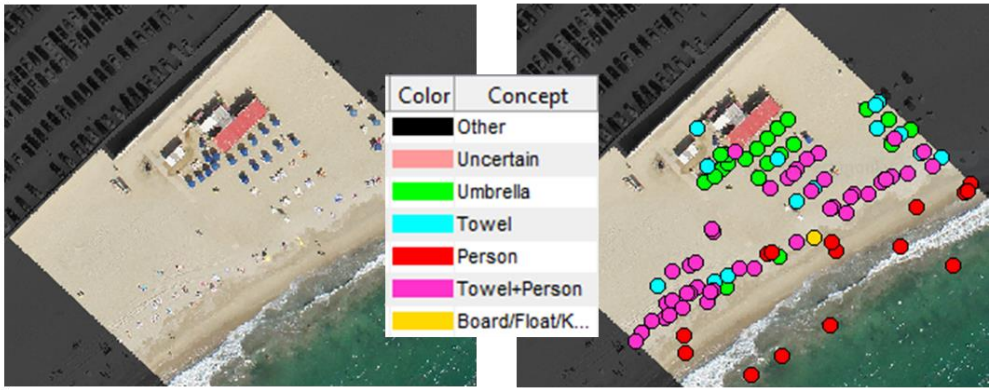




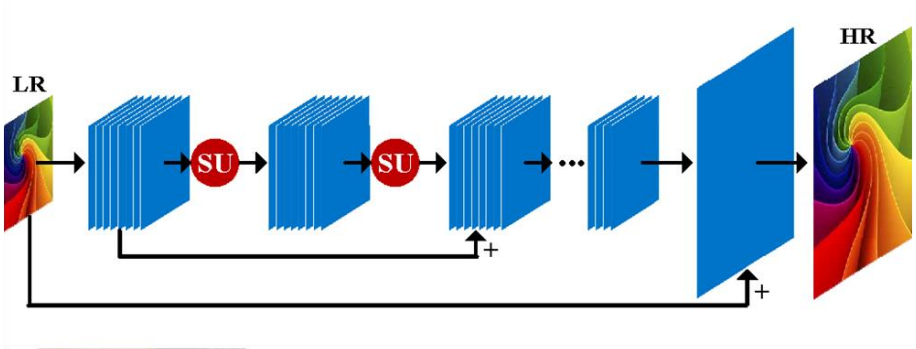
Estudio de distintas estrategias de nitrogenación de cultivos y su incidencia en el vigor de la plantación.

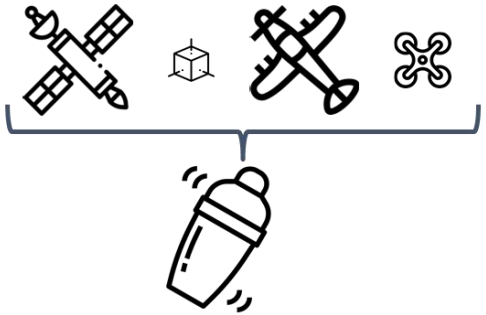
Temas de investigación: Detección de objetos





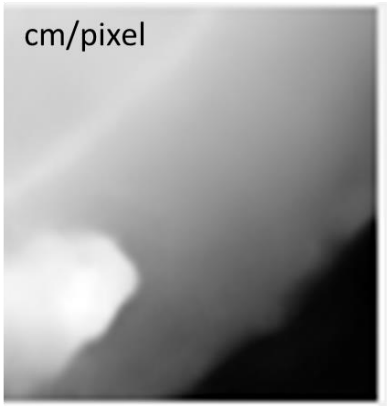
Temas de investigación: Super-resolución / Fusión

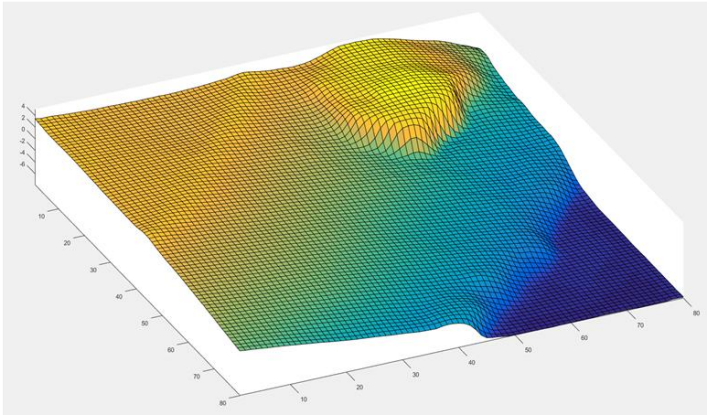




cm/pixel

m/pixel





Necesidades

- Requisitos para trabajar operativamente con datos georeferenciados
  - Herramienta completa para visualizar y analizar resultados
  - Desarrollo y aplicación de modelos de alto rendimiento

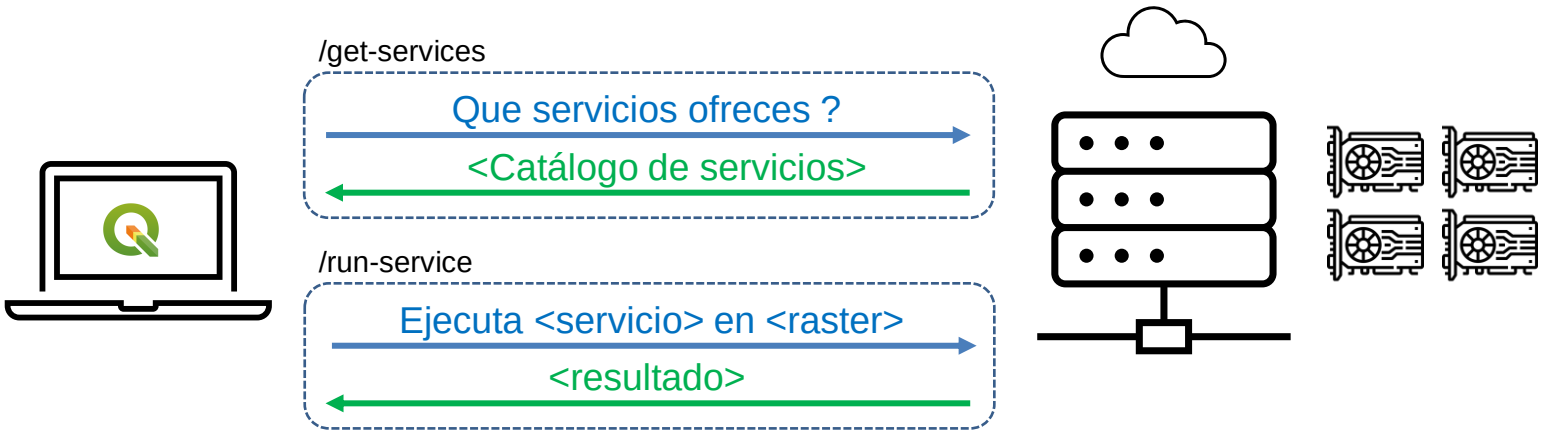
Complementos de QGIS

- QGIS ofrece un mecanismo de complementos (*plugins*) que posibilita integrar nuestros algoritmos de visión por computador dentro de sus opciones de proceso de datos.
  - Para una ejecución rápida de los algoritmos, el ordenador ejecutando QGIS debería de disponer de capacidad de proceso paralelo (GPUs i/o TPUs).
  - Disponer de esta capacidad para todos nuestros ordenadores tiene un alto coste, y no se aprovecha intensivamente el hardware necesario.

- **Propuesta**: desarrollar un complemento de QGIS que permita conectar este sistema de información geográfica con un servidor de cómputo remoto compartido, el cual procese los datos usando hardware paralelo y retorne su resultado de vuelta.

Componentes del Sistema

- Nuestra aproximación implementa una arquitectura de servicios web basada en la aproximación de transferencia de estado representacional (REST API)



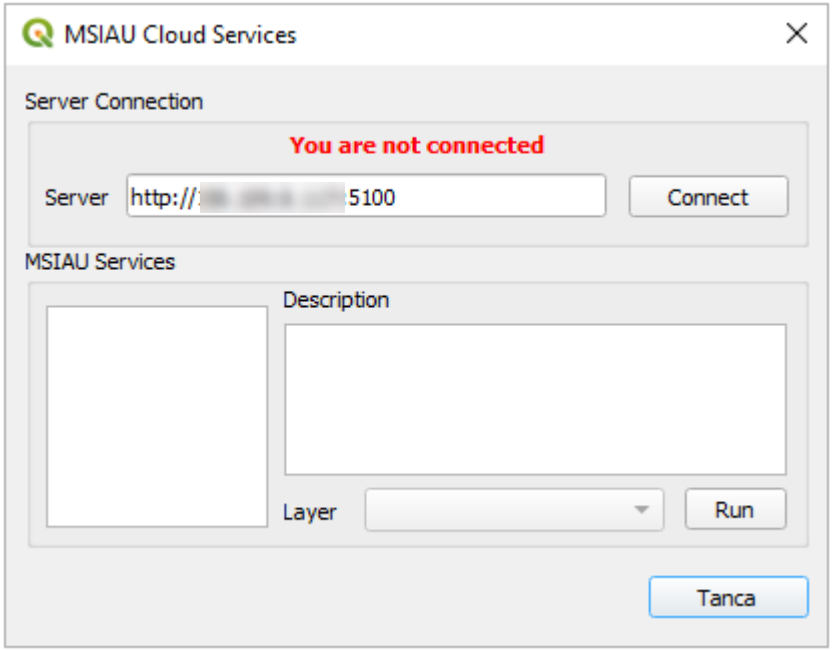
Ordenador personal
- QGIS
- Complemento de conexión

Servidor de cálculo
- Flask (Gestión Microservicios Web)
- Docker (Despliegue de contenedores)

GPU icon created by linector

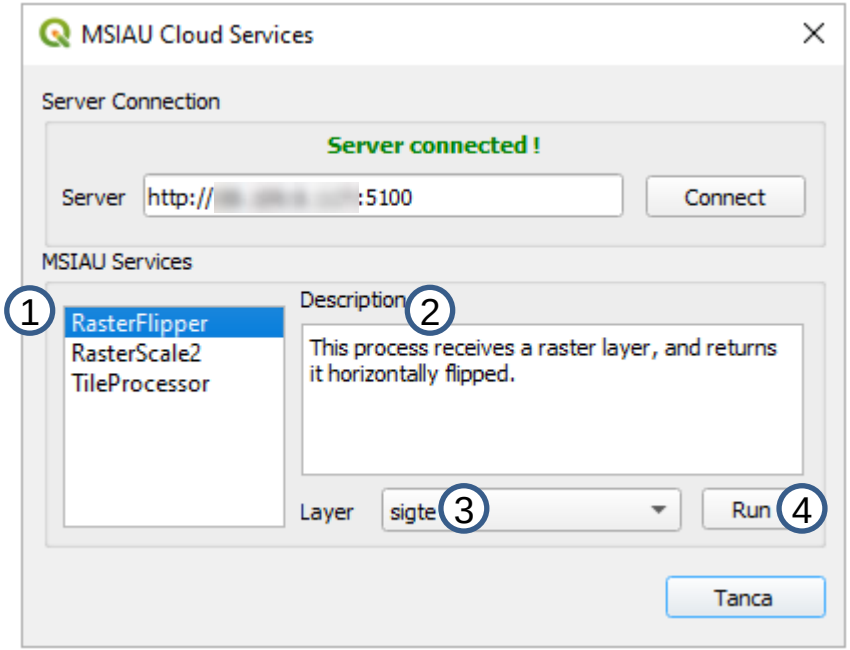
Complemento QGIS de Conexión (I)

- Complemento desarrollado en Python, basado la librería HTTP *Requests* para comunicar-se con el proveedor de microservicios remoto.



Complemento QGIS de Conexión (II)

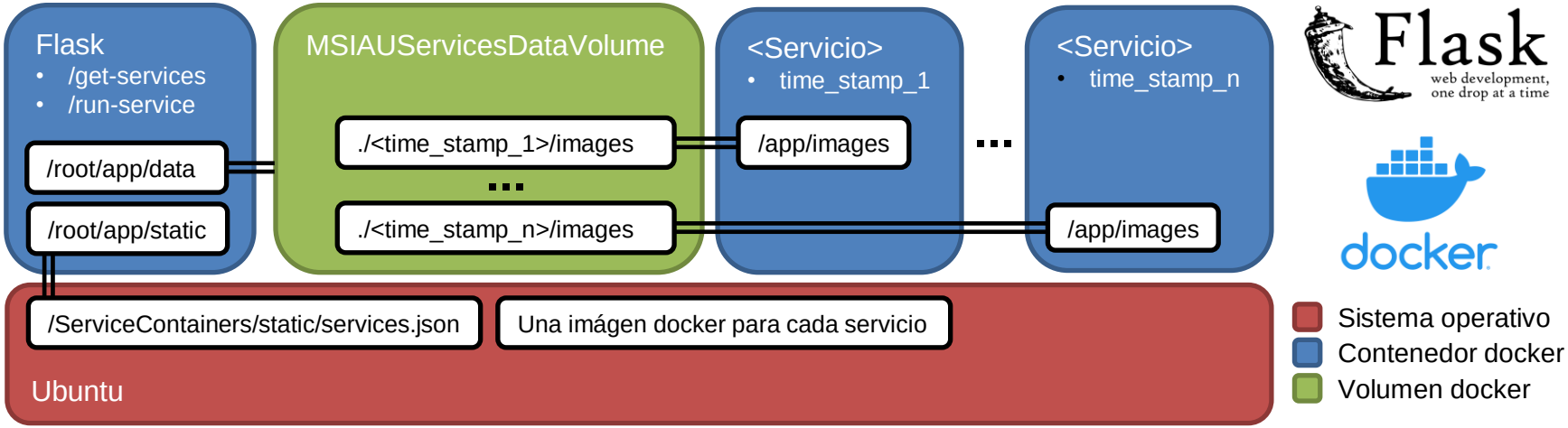
- Desarrollado en Python, usando la librería HTTP *Requests* para comunicarse con el proveedor de microservicios externo.



1. Catálogo de servicios
2. Descripción del servicio seleccionado.
3. Selector de la capa a procesar
4. Gestión del envío de datos, proceso y recepción de resultados.

Gestión de Microservicios

- Servidor basado en Ubuntu, lanzando un contenedor de Docker donde se ejecuta el framework Flask para ofrecer los microservicios.



1. Para cada petición /run-service el contenedor Flask lanza un contenedor de servicio.
2. La transferencia de datos entre el contenedor Flask y el contenedor de servicio se vehicula mediante un volumen Docker, con directorios con marca de tiempo.
3. Los contenedores de servicio desaparecen al finalizar su tarea.
4. El contenedor Flask recoge el resultado del volumen Docker y lo envía al complemento QGIS.

Catálogo de servicios

- /ServiceContainers/static/services.json


```
"services": [
    {
            "id": 1001,
            "name": "RasterFlipper",
            "desc": "This process receives a raster layer, and returns it horizontally flipped.",
            "docker_image": "flip-service",
            "valid_extensions": ["tiff"],
            "result_prefix": "flipped_"
    },
    {
            "id": 1002,
            "name": "RasterScale2",
            "desc": "This process receives a raster layer, and returns it scaled with a 2 factor.",
            "docker_image": "resize-service",
            "valid_extensions": ["tiff"],
            "result_prefix": "resized_"
    },
    {
            "id": 1003,
            "name": "TileProcessor",
            "desc": "This process receives a raster layer, processes it divided in tiles, and returns it.”,
            "docker_image": "retile-example-service",
            "valid_extensions": ["tiff"],
            "result_prefix": "retiled_"
    }

  ]
}
```

Composición de un servicio

- Un servicio corresponde a un código Python que recibe una imagen a procesar y el nombre de la imagen de resultado a generar, y que se ejecuta mediante un contenedor Docker

flip_image.py

```
import cv2
import sys
import os
import gdal

if len(sys.argv)!=3:
    print('Wrong syntax: Incorrect number of parameters')
    print(' > flip_image <input_image> <output_image>')
else:
    input_image = sys.argv[1]
    output_image = sys.argv[2]

    try:
        os.chdir('/app/images')
        print('Loading {}'.format(input_image))
        original_image = cv2.imread(input_image)

        # Get projection and geotransform  from the input_image
        gdal_dataset = gdal.Open(input_image);
        projection = gdal_dataset.GetProjection()
        geo_transform = gdal_dataset.GetGeoTransform()

        # Image process provided by the service
        result = cv2.flip(original_image, 1)

        # Result saving on disk
        print('Saving {}'.format(output_image))
        cv2.imwrite(output_image, result)
        print('Image saved')

        # Stablishment of geo info to the generated images
        gdal_dataset = gdal.Open(output_image, gdal.GA_Update)
        gdal_dataset.SetGeoTransform(geo_transform)
        gdal_dataset.SetProjection(projection)

    except cv2.error as e:
        print('Problems loading or saving {}'.format(input_image))
```

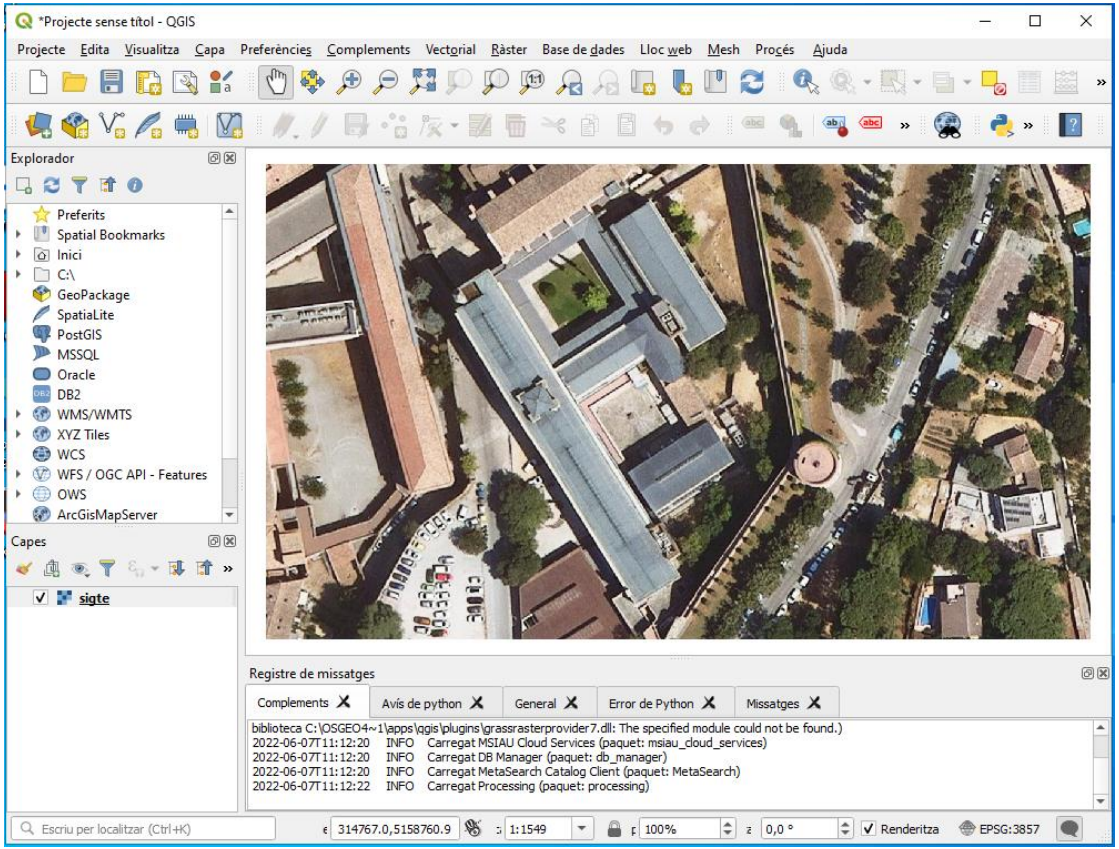
Dockerfile

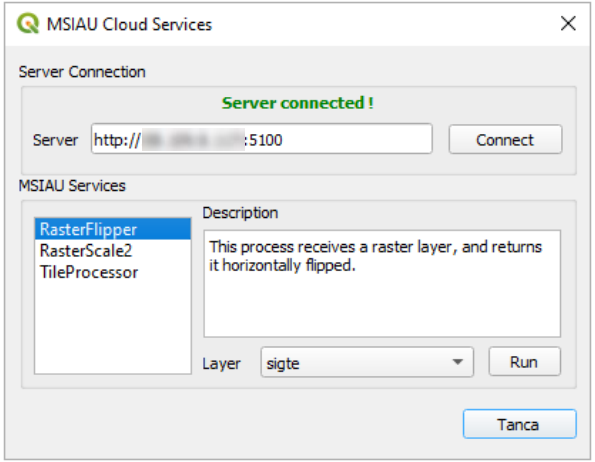
```
FROM ubuntu:18.04

WORKDIR /app

RUN apt-get update
RUN apt-get -y upgrade

# Python 3 installation
RUN apt-get update \
  && apt-get install -y python3-pip python3-dev \
  && cd /usr/local/bin \
  && ln -s /usr/bin/python3 python \
  && pip3 install --upgrade pip

# Install GDAL to avoid missing geotiff referentiation
RUN apt-get -y install software-properties-common
RUN add-apt-repository ppa:ubuntugis/ppa
RUN apt-get update
RUN apt-get install -y gdal-bin
RUN apt-get install -y libgdal-dev
ENV CPLUS_INCLUDE_PATH=/usr/include/gdal
ENV C_INCLUDE_PATH=/usr/include/gdal
RUN pip install GDAL==2.4.2

# With requirements, the required package versions are installed.
COPY requirements.txt .
RUN pip install -r requirements.txt

# The code to be executed is copied on the WORKDIR
COPY flip_image.py .

RUN mkdir -p /app/images

ENTRYPOINT [ "python", "flip_image.py" ]
```

requirements.txt

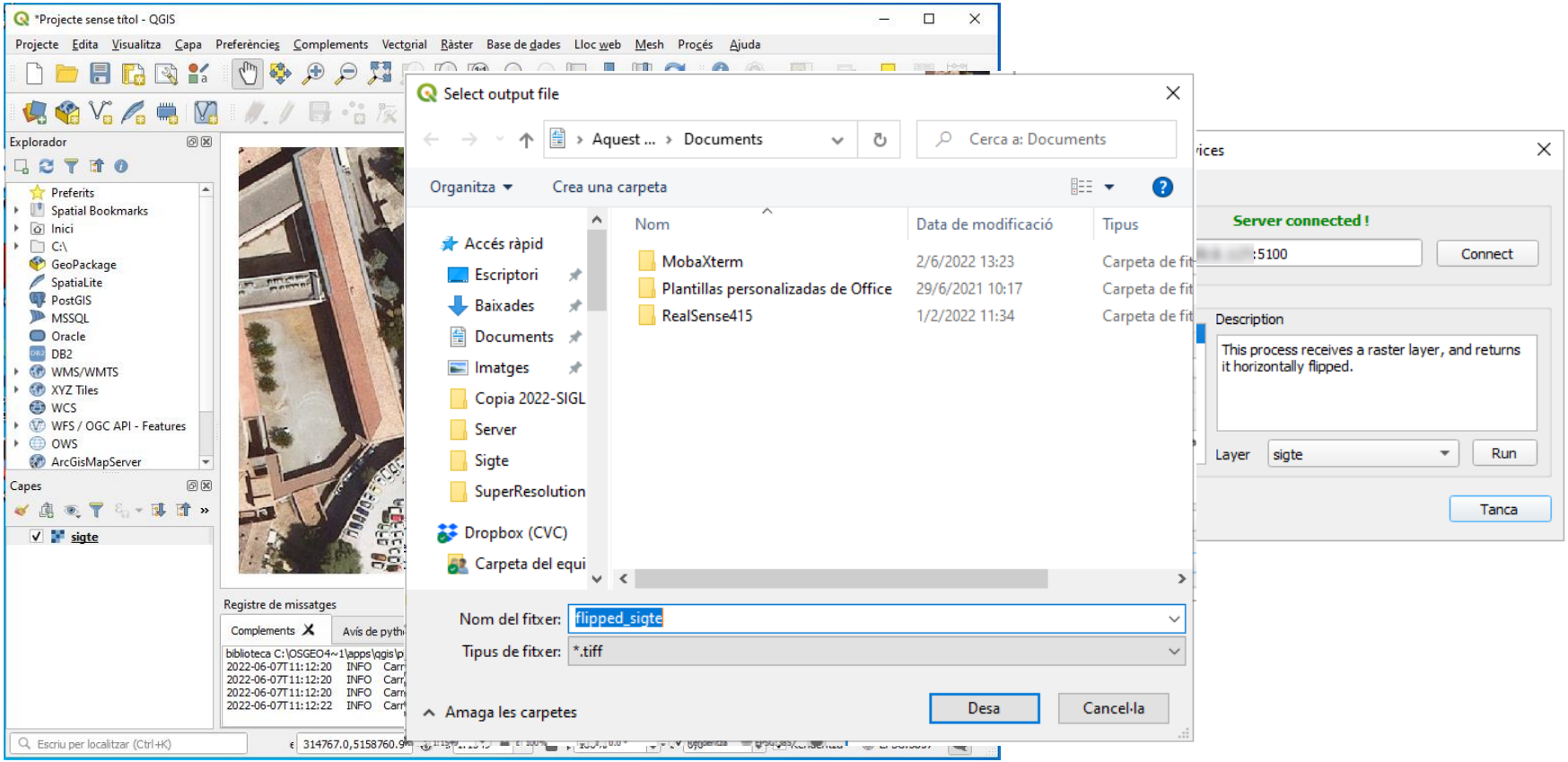
```
setuptools
opencv-python-headlless
numpy
scipy
skia-python
```

Ejemplo

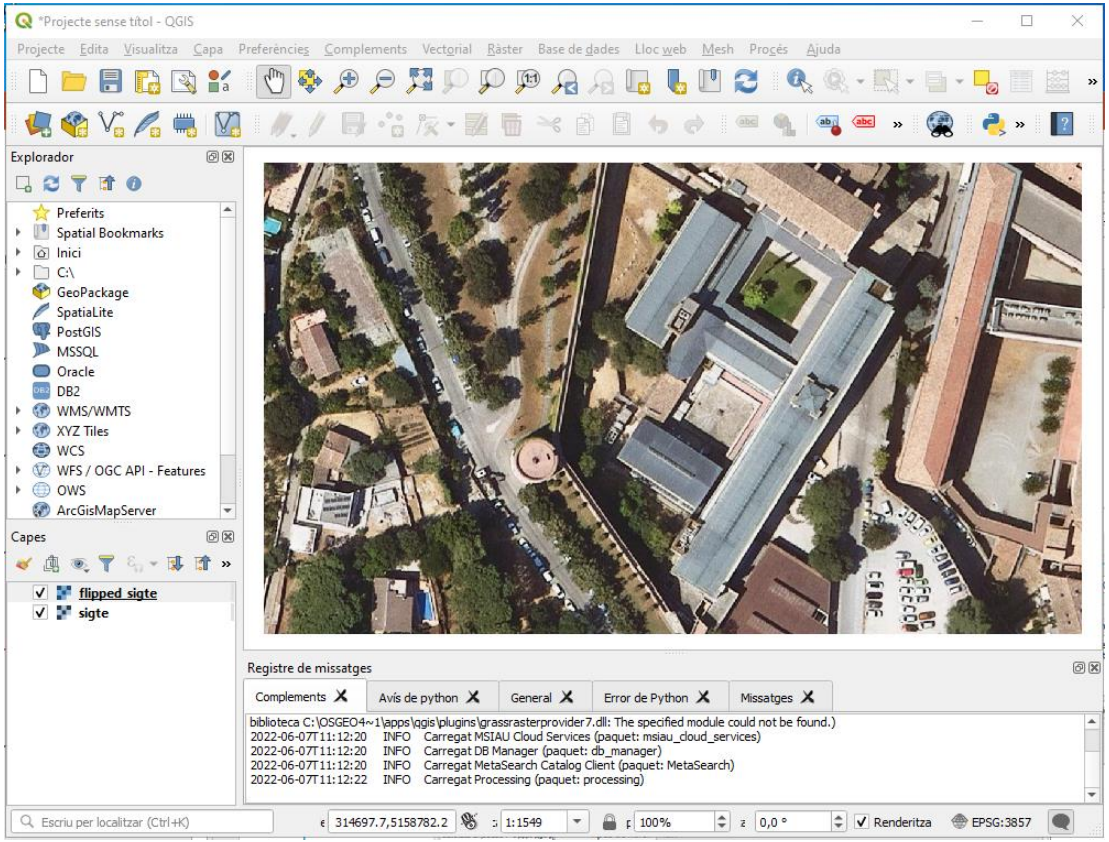

*Projecte sense títol - QGIS
Projecte Edita Visualitza Capa Preferències Complements Vectorial Ràster Base de dades Lloc web Mesh Procés Ajuda
Explorador
Preferits
Spatial Bookmarks
Inici
C:\
GeoPackage
SpatiaLite
PostGIS
MSSQL
Oracle
DB2
WMS/WMTS
XYZ Tiles
WCS
WFS / OGC API - Features
OWS
ArcGisMapServer
Capes
sigte
Registre de missatges
Complements
Avís de python
General
Error de Python
Missatges
biblioteca C:\OSGEO4~1\apps\qgis\plugins\grassrasterprovider7.dll: The specified module could not be found.)
2022-06-07T11:12:20 INFO Carregat MSIAU Cloud Services (paquet: msiau_cloud_services)
2022-06-07T11:12:20 INFO Carregat DB Manager (paquet: db_manager)
2022-06-07T11:12:20 INFO Carregat MetaSearch Catalog Client (paquet: MetaSearch)
2022-06-07T11:12:22 INFO Carregat Processing (paquet: processing)
Escriu per localitzar (Ctrl+K)
314767.0,5158760.9
1:1549
100%
0,0 °
Renderitza
EPSG:3857

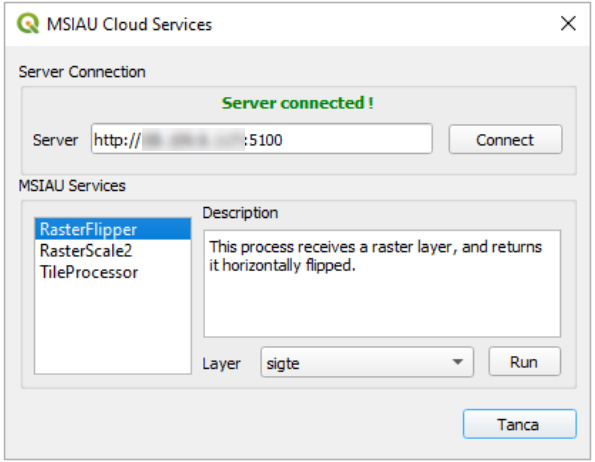

MSIAU Cloud Services
Server Connection
Server connected !
Server http://:5100
Connect
MSIAU Services
RasterFlipper
RasterScale2
TileProcessor
Description
This process receives a raster layer, and returns it horizontally flipped.
Layer sigte
Run
Tanca

Ejemplo


*Projecte sense títol - QGIS
Projecte Edita Visualitza Capa Preferències Complements Vectorial Ràster Base de dades Lloc web Mesh Procés Ajuda
Explorador
Preferits
Spatial Bookmarks
Inici
C:\
GeoPackage
SpatiaLite
PostGIS
MSSQL
Oracle
DB2
WMS/WMTS
XYZ Tiles
WCS
WFS / OGC API - Features
OWS
ArcGisMapServer
Capes
sigte
Registre de missatges
Complements
Escriu per localitzar (Ctrl+K)
Select output file
Aquest ... › Documents
Cerca a: Documents
Organitza
Crea una carpeta
Accés ràpid
Escriptori
Baixades
Documents
Imatges
Copia 2022-SIGL
Server
Sigte
SuperResolution
Dropbox (CVC)
Carpeta del equi
Nom
Data de modificació
Tipus
MobaXterm 2/6/2022 13:23 Carpeta de fit
Plantillas personalizadas de Office 29/6/2021 10:17 Carpeta de fit
RealSense415 1/2/2022 11:34 Carpeta de fit
Nom del fitxer: flipped_sigte
Tipus de fitxer: *.tiff
Amaga les carpetes
Desa
Cancel·la
Server connected !
:5100
Connect
Description
This process receives a raster layer, and returns it horizontally flipped.
Layer
sigte
Run
Tanca

Ejemplo


*Projecte sense títol - QGIS
Projecte Edita Visualitza Capa Preferències Complements Vectorial Ràster Base de dades Lloc web Mesh Procés Ajuda
Explorador
Preferits
Spatial Bookmarks
Inici
C:\
GeoPackage
SpatiaLite
PostGIS
MSSQL
Oracle
DB2
WMS/WMTS
XYZ Tiles
WCS
WFS / OGC API - Features
OWS
ArcGisMapServer
Capes
flipped_sigte
sigte
Registre de missatges
Complements
Avís de python
General
Error de Python
Missatges
biblioteca C:\OSGEO4~1\apps\qgis\plugins\grassrasterprovider7.dll: The specified module could not be found.)
2022-06-07T11:12:20 INFO Carregat MSIAU Cloud Services (paquet: msiau_cloud_services)
2022-06-07T11:12:20 INFO Carregat DB Manager (paquet: db_manager)
2022-06-07T11:12:20 INFO Carregat MetaSearch Catalog Client (paquet: MetaSearch)
2022-06-07T11:12:22 INFO Carregat Processing (paquet: processing)
Escriu per localitzar (Ctrl+K)
314697.7,5158782.2
1:1549
100%
0,0 °
Renderitza
EPSG:3857


MSIAU Cloud Services
Server Connection
Server connected !
Server http://:5100
Connect
MSIAU Services
RasterFlipper
RasterScale2
TileProcessor
Description
This process receives a raster layer, and returns it horizontally flipped.
Layer sigte
Run
Tanca

Conclusiones

- Se ha desarrollado un mecanismo efectivo para procesar datos raster en remoto desde QGIS.

- Un complemento QGIS gestiona la comunicación con un servidor remoto.

- El proceso de datos se centraliza en dicho servidor, que ejecuta una REST API implementada en Flask y despliega un contenedor Docker para cada petición de servicio recibida.
  - Hardware de proceso paralelo es compartido por todos los usuarios, incrementando su utilización.
  - El uso de docker promueve la escalabilidad del sistema y evita colisiones entre las librerías usadas para proveer los servicios.
  - Los servicios ofrecidos se pueden actualizar e incrementar sin necesidad de actualizar el complemento QGIS ni reiniciar Flask.

Líneas de mejora

- Si el algoritmo de proceso es muy costoso, o si la capa raster a procesar es muy pesada, la generación del resultado no es inmediata.
    - Seria interesante implementar un mecanismo de barra de progreso informativa para indicar como evoluciona el servicio solicitado.
- El sistema de microservicios es operativo para una carga moderada.
    - El servidor web integrado en Flask se proporciona para facilitar el desarrollo. En un entorno exigente se tendría que substituir por un servidor más potente (Nginx, Gunicorn, ...)

Agradecimientos

- Queremos agradecer la participación en el desarrollo del sistema de Raúl Estrada Herrerias y en especial de Oriol Casas Carrasquer.

Gracias por su atención !

Para cualquier duda o aclaración, no dude en contactarme

- Daniel Ponsa : daniel.ponsa@cvc.uab.cat