



Universidad de Concepción

Facultad de Ingeniería

Dpto. Ingeniería Informática y Ciencias de la Computación

Proyecto Semestral

Integrantes

Cecilia Huerta San Martín

Rodrigo Rivera Fierro

Meraioth Ulloa Salazar

Docentes

Diego Seco

Andrea Rodriguez

Julio 2017 - Concepción, CHILE

Índice

Iteración 1	3
Tarea 1	3
Tarea 2	4
Tarea 3	6
Tarea 4	8
Iteración 2	10
Tarea 1	10
Tarea 2	16
Referencias	19

Iteración 1

Tarea 1

Para comenzar, se descargaron los archivos TIFF, para los campus Concepción y Chillán de la UdeC, desde la fuente señalada, para luego importarlos a QGIS. La visualización de los datos se presenta a continuación.

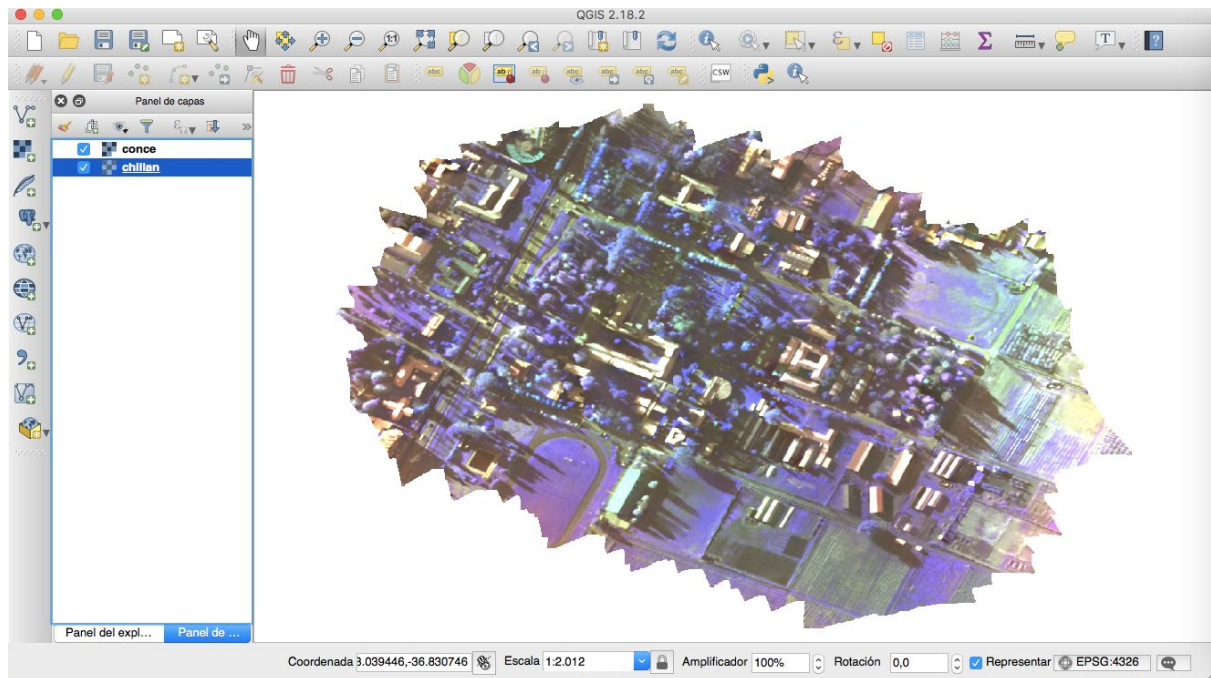


Figura 1. TIFF UdeC campus Chillán.

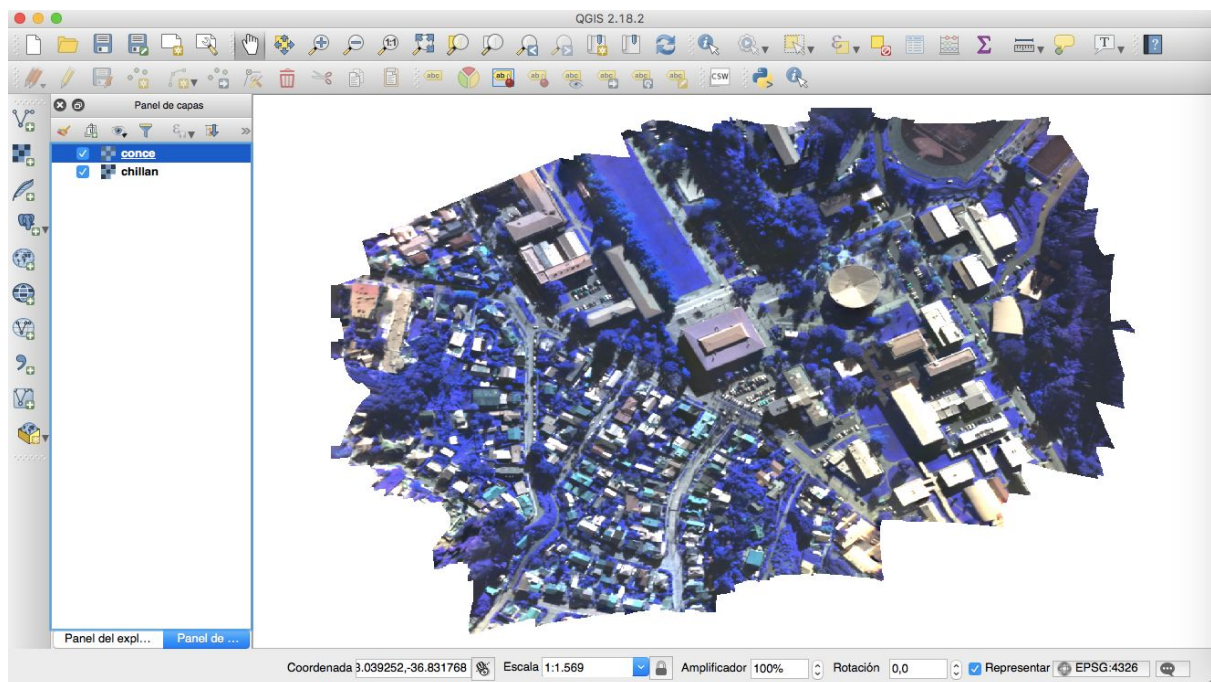


Figura 2. TIFF UdeC campus Concepción.

Tarea 2

Se procede a la extracción de los elementos de información en base a los datos del campus Concepción, y creación de las bases de datos para su almacenamiento. Los datos fueron extraídos desde OpenStreetMap con la herramienta *JOSM*[1], que dispone de una interfaz gráfica para obtener los archivos OSM.

Luego de obtener el archivo OSM, se filtró de acuerdo a 6 elementos de información, con la herramienta *osmfilter*. El procedimiento se detalla a continuación:

- Edificios
 - `osmfilter conce2.osm --keep="building"`
`-o=conce_edificios.osm`
- Canchas
 - `osmfilter conce2.osm --keep="leisure"`
`-o=conce_canchas.osm`
- Estacionamientos
 - `osmfilter conce2.osm --keep="amenity=parking"`
`-o=conce_estacionamientos.osm`
- Autopista
 - `osmfilter conce2.osm --keep="highway=service or highway=residential or highway=tertiary or highway=unclassified"` `-o=conce_autopista.osm`
- Peatonal
 - `osmfilter conce2.osm --keep="highway=footway or highway=steps or highway=pedestrian or highway=path or highway=unclassified"` `-o=conce_peatonal.osm`
- Peatonal Techado
 - `osmfilter conce2.osm --keep="tunnel"`
`-o=conce_techado.osm`

Los archivos OSM generados se adjuntan en el archivo comprimido.

Luego, se crearon 6 bases de datos, una para cada componente, *conce_autopistas*, *conce_techado*, *conce_peatonal*, *conce_estacionamientos*, *conce_canchas*, *conce_edificios*.

Se detalla el procedimiento para el llenado de una base de datos, ya que los demás sólo varían en el nombre de ésta. Se utilizó una aplicación Java de línea de comandos, llamada *osmosis*[2], para el procesamiento de los archivos OSM.

A continuación, se presenta los scripts utilizados.

```
./createdb --u postgres conce_edificios
```

Creación de extensiones para la base de datos:

```
./psql -u postgres -d conce_edificios 'CREATE EXTENSION postgis;  
CREATE EXTENSION hstore;'
```

Importación de los esquemas de Osmosis:

```
./psql -U postgres -d conce_edificios -f  
/usr/local/Cellar/osmosis/0.45/libexec/script/pgsnapshot_schema_0.  
6.sql
```

```
./psql -U postgres -d conce_edificios -f  
/usr/local/Cellar/osmosis/0.45/libexec/script/pgsnapshot_schema_0.  
6_action.sql
```

```
./psql -U postgres -d conce_edificios -f  
/usr/local/Cellar/osmosis/0.45/libexec/script/pgsnapshot_schema_0.  
6_bbox.sql
```

```
./psql -U postgres -d conce_edificios -f  
/usr/local/Cellar/osmosis/0.45/libexec/script/pgsnapshot_schema_0.  
6_linestring.sql
```

Importación del archivo OSM a la base de datos *conce_edificios*:

```
osmosis --read-xml conce_edificios.osm --write-pgsql user=postgres  
password=**** database=conce_edificios
```

Geoserver no reconoce el tipo de dato 'hstore', de la columna 'tags', utilizado por osmosis al importar los datos de OpenStreetMaps a la base de datos Postgres, por lo que es necesario crear una columna extra por cada tag que se quiera conservar.

En nuestro caso sólo guardamos el tag 'name' almacenando como VARCHAR(128) en Postgres.

Los script utilizados para la tarea descrita fueron los siguientes:

```
ALTER TABLE ways  
ADD name VARCHAR(128);  
  
UPDATE ways  
SET name = tags->'name';
```

Tras esto, se juntan todas las bases de datos en una sola, dejando cada elemento en una única tabla. Para ejecutar la siguiente línea por la base de datos de cada elemento:

```
pg_dump -h localhost -U postgres ways conce_edificios | psql -h localhost -U postgres conce
```

Adicionalmente se eliminó de cada tabla todas las columnas innecesarias para evitar problemas de interpretación en geoserver, quedando solo las columnas 'id', 'linestring' y 'name'.

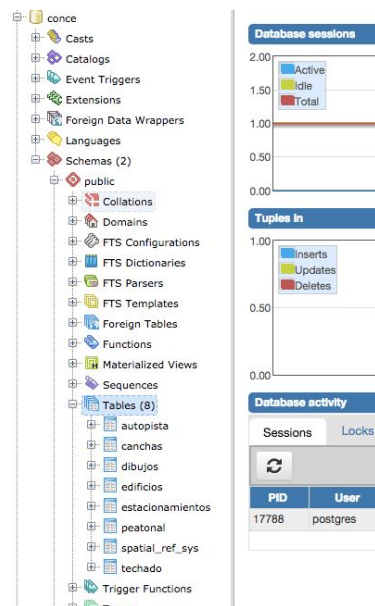


Figura 3. Bases de datos para elementos de información.

Tarea 3

Para la publicación de las capas correspondientes al fichero TIFF para el campus Concepción y los archivos OSM obtenidos en el ítem anterior, se creó un nuevo espacio de trabajo, llamado *conce*, seguido de un almacén de datos para cada elemento. A continuación, y para cada uno, se añadió el elemento correspondiente a *ways*, publicando y visualizando la capa para su comprobación.

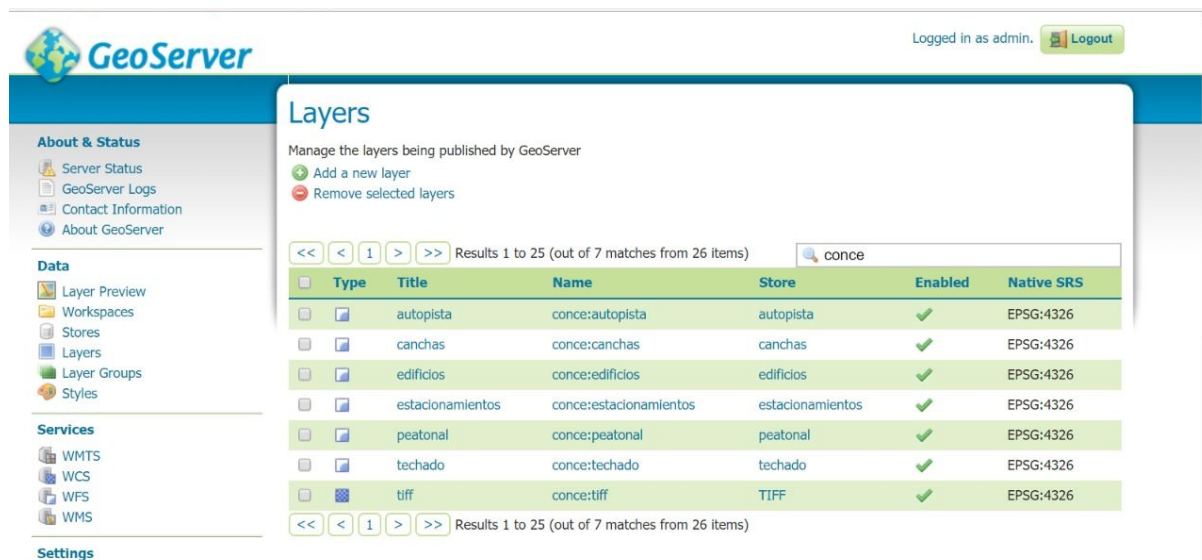


Figura 4. Capas publicadas en GeoServer.

Al ejecutar la operación *getCapabilities* desde el navegador, se obtiene el siguiente resultado:

```
<?xml version='1.0' encoding='UTF-8'>
<wfs:WFS_Capabilities xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:conce="conce" version="2.0.0"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://localhost:8080/geoserver/schemas/wfs/2.0/wfs.xsd"
  updateSequence="373">
  <ows:ServiceIdentification>...</ows:ServiceIdentification>
  <ows:ServiceProvider>...</ows:ServiceProvider>
  <ows:OperationsMetadata>...</ows:OperationsMetadata>
  <FeatureTypeList>
    <FeatureType xmlns:conce="conce">
      <Name>conce:autopista</Name>
      <Title>autopista</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:canchas</Name>
      <Title>canchas</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:edificios</Name>
      <Title>edificios</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:estacionamientos</Name>
      <Title>estacionamientos</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:peatonal</Name>
      <Title>peatonal</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:techado</Name>
      <Title>techado</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
    <FeatureType xmlns:conce="conce">
      <Name>conce:tiff</Name>
      <Title>tiff</Title>
      <Abstract/>
      <ows:Keywords>...</ows:Keywords>
      <DefaultCRS>urn:ogc:def:crs:EPSG::4326</DefaultCRS>
      <ows:WGS84BoundingBox>...</ows:WGS84BoundingBox>
    </FeatureType>
  </FeatureTypeList>
</wfs:WFS_Capabilities>
```

Figura 5. Parte del resultado obtenido de operación *getCapabilities* de WFS.

El fichero completo se adjunta en el archivo comprimido.

Finalmente, la visualización de las capas utilizando el servicio WFS desde QGis.

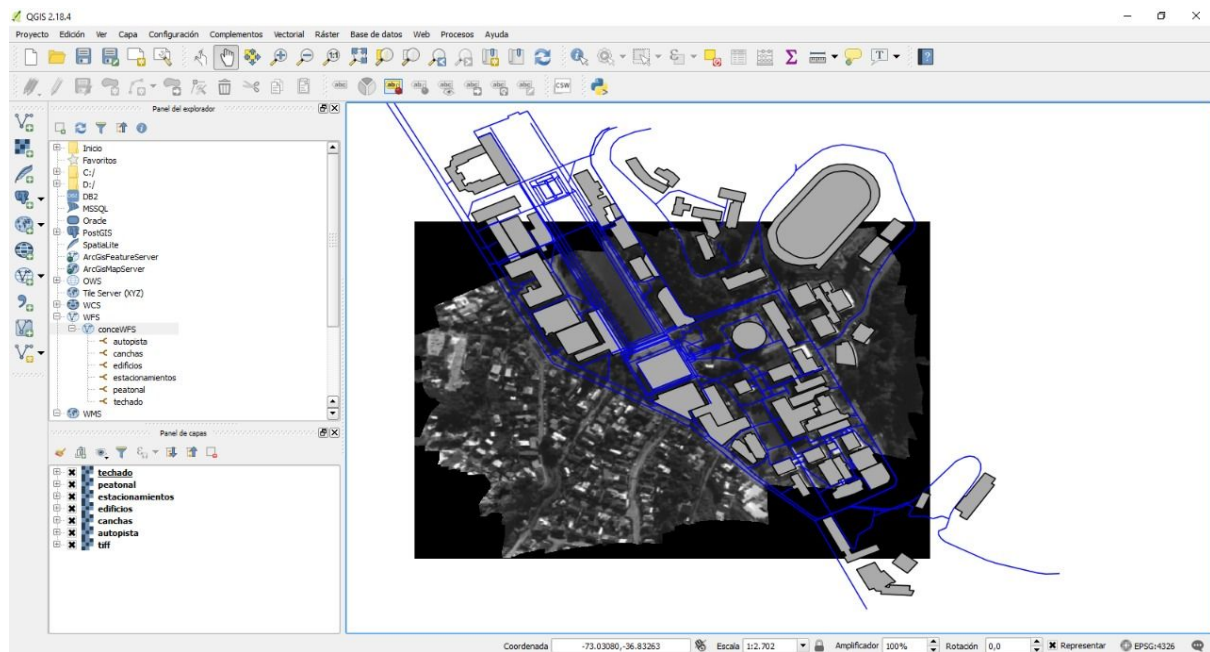


Figura 6. Visualización de capas en Qgis desde servicio WFS de GeoServer.

Tarea 4

Utilizando como base los ejemplos utilizados en el laboratorio, se creó una aplicación GIS web con Leaflet, que permite la visualización de las capas publicadas en Geoserver para el campus Concepción, desde un navegador.

Para comenzar, se utilizó como base para la visualización de las capas, el mapa del campus Concepción, en el cual se superpondrán las capas correspondientes al TIFF del campus, y los elementos generados a partir del OSM, *edificios*, *canchas*, *estacionamientos*, *calles*, *camino peatonales* y *camino techados*.

Las capas fueron agrupadas para aplicar un control sobre éstas. De esta forma, se pueden visualizar sólo las capas seleccionadas. Por defecto, al ejecutar la aplicación se desplegarán las capas del TIFF, los edificios y las calles, tal como se muestra en la siguiente imagen:

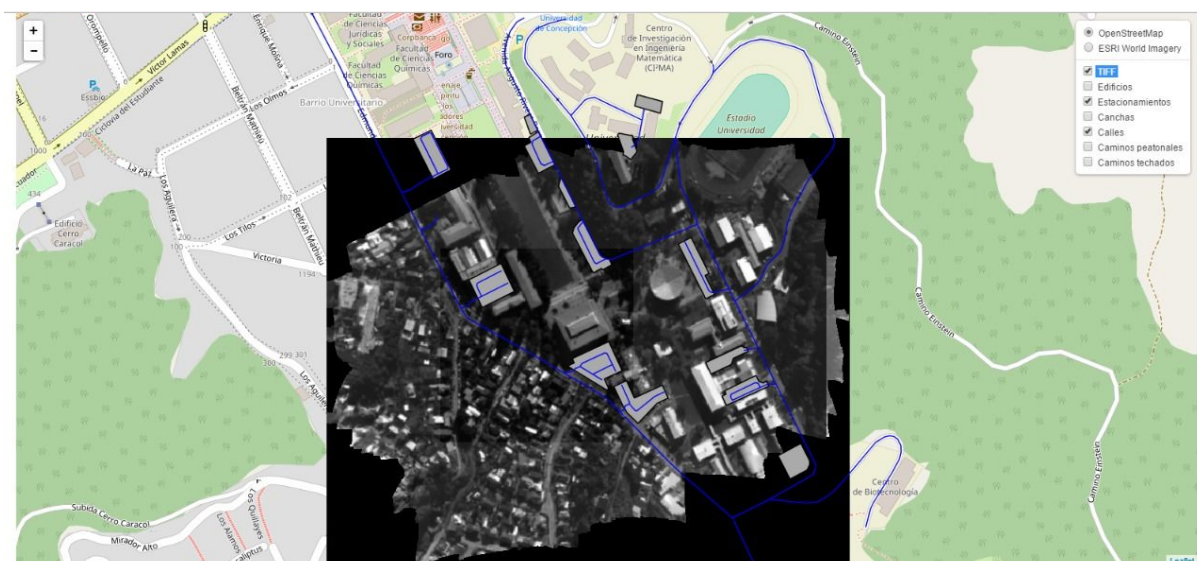


Figura 7. Visualización de aplicación Leaflet en Navegador, con 6 features.

El código fuente de la aplicación, “Aplicación Leaflet Udec”, se adjunta en el archivo comprimido.

Iteración 2

Tarea 1

Normalized Difference Vegetation Index (NDVI): corresponde al índice de vegetación más utilizado. Se basa en el comportamiento radiométrico de la vegetación, el cual se relaciona con la actividad fotosintética y las estructura foliar de las plantas, lo que permite determinar su vigor. Los valores del NDVI están en función de la energía que absorbe o refleja la planta, en diversos segmentos del espectro electromagnético.[3]

Este índice se define de acuerdo a la siguiente relación:

$$NDVI = \frac{NIR - R}{NIR + R}$$

Los valores obtenidos están en el rango [-1, 1], lo cual facilita su interpretación.

Dentro de sus principales utilidades se pueden destacar que ha sido utilizado para estimar variables como, el índice de área foliar (LAI), el flujo neto de CO₂, la radiación fotosintética activa que ha absorbido la planta (APAR), la productividad neta de la vegetación, el contenido de clorofila, la evapotranspiración potencial y la cantidad de lluvia que ha sido recibida por la cubierta vegetal. Habitualmente tiene aplicaciones en procesos como el seguimiento de las condiciones de la cubierta vegetal a escala global, estudios de deforestación, detección de incendios, caracterización de biomas a escala global, prevención de sequía y riesgo de incendios forestales mediante el análisis de series multitemporales.

Normalized Difference Red Edge (NDRE): utiliza un filtro de borde rojo para ver la reflectancia de la copa del cultivo. El borde rojo es una región en la zona rojo-NIR del espectro de reflectancia de la vegetación y marca el límite entre la absorción por la clorofila en la región roja visible y la dispersión debido a la estructura interna de la hoja. Esto permite determinar muchas variables diferentes con el manejo del cultivo. La comprensión de los niveles de clorofila puede proporcionar la capacidad de monitorear la actividad de la fotosíntesis[4].

Puede ser usado para identificar:

- Contenido de Clorofila en la hoja
- Vigor de la planta
- Detección de estrés
- Demanda de Fertilizante
- Captación de Nitrógeno

Su fórmula corresponde a:

$$NDRE = \frac{NIR - RE}{NIR + RE}$$

Green NDVI (GNDVI): corresponde a una variante del NDVI, en donde se utiliza la banda Green, en vez de la banda Red. Se define mediante la siguiente función:

$$GNDVI = \frac{NIR - G}{NIR + G}$$

Dentro de las principales utilidades de este índice están realizar la estimación del rendimiento del cultivo de arroz inundado, en donde se dispone de parcelas fertilizadas con residuos orgánicos y nitrógeno mineral en distintas dosis.

Green RVI (GRVI): Green Ratio Vegetation Index o Green Vegetation Index (GVI). En este índice se corrigen los problemas que presenta el índice NDVI. Su representación se hace con la siguiente función:

$$GRVI = \frac{NIR}{G}$$

En este caso, el índice muestra lo mismo que NDVI, pero tomando en cuenta el estado fenológico de la vegetación, permitiendo realizar mediciones en épocas del año en donde la vegetación no es necesariamente verde.

Los 4 índices antes descritos fueron implementados utilizando la consola Python de QGIS. El código fuente para cada uno, se encuentra en el archivo adjunto *Código_índices.txt*. En general, la implementación de los índices es similar, donde sólo varía la fórmula con la cual se calcula cada uno. En este caso se detalla el código para la implementación del índice *Normalized Difference Vegetation Index (NDVI)*.

```
from qgis.analysis import QgsRasterCalculator,  
QgsRasterCalculatorEntry
```

```
cLayer = qgis.utils.iface.activeLayer()  
entries = []
```

```
c2 = QgsRasterCalculatorEntry()  
c2.ref = 'conce@2'  
c2.raster = cLayer  
c2.bandNumber = 2  
entries.append(c2)
```

```
c4 = QgsRasterCalculatorEntry()  
c4.ref = 'conce@4'  
c4.raster = cLayer  
c4.bandNumber = 4  
entries.append(c4)
```

```

calc = QgsRasterCalculator(
' ( ( conce@4 - conce@2 ) / ( conce@4 + conce@2 ) ) ',
'C:\\Users\\Cecilia\\Desktop\\NDVI.tif',
'GTiff',
cLayer.extent(),
cLayer.width(),
cLayer.height(),
entries
)
print calc.processCalculation()

```

A continuación se muestran las capturas de pantalla de los índices visualizados en QGis. Esclarecer que éstos son visibiles inicialmente en escala de grises, por lo cual es necesario cambiar el estilo para su posterior visualización.

Para el NDVI se utilizó una paleta de colores, que variaba entre el verde y el rojo. De esta forma se diferenció lo que corresponde a vegetación de lo que no lo es, edificaciones, calles, etc.

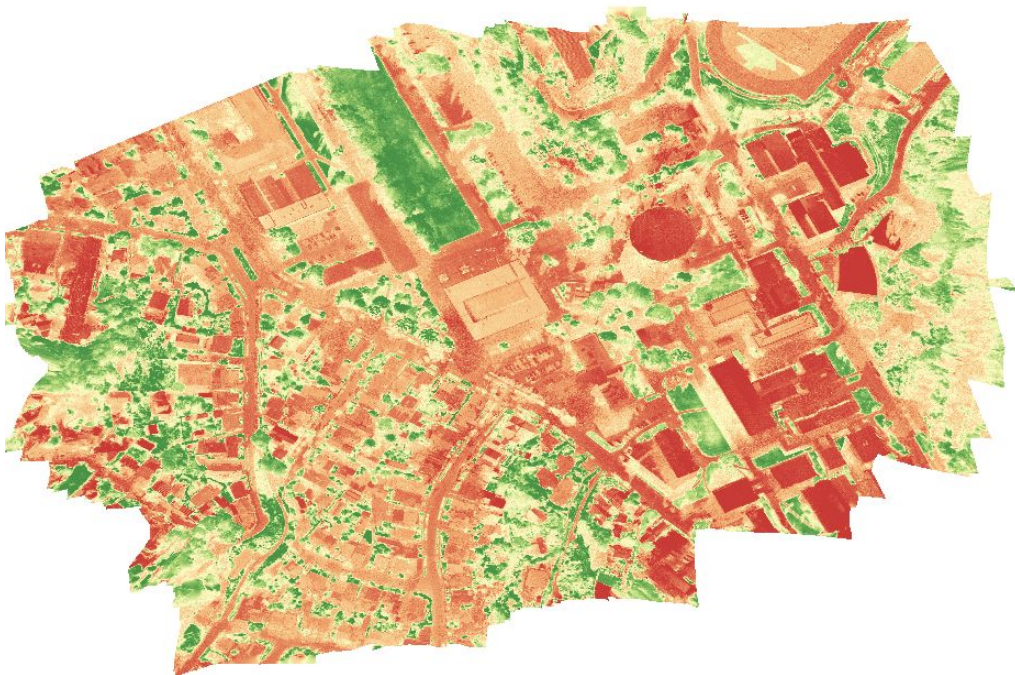


Figura 8. Índice NDVI

Para el NDRE, como no se utilizó un control para referenciar lo que no corresponde a vegetación, se optó por obviar el color morado de la imagen, considerando solamente el color verde. Éste indica diferencias en la densidad de la vegetación existente, y la intensidad del color verde de la misma, lo que tiene directa relación con la cantidad de clorofila tipo A que posee la vegetación.

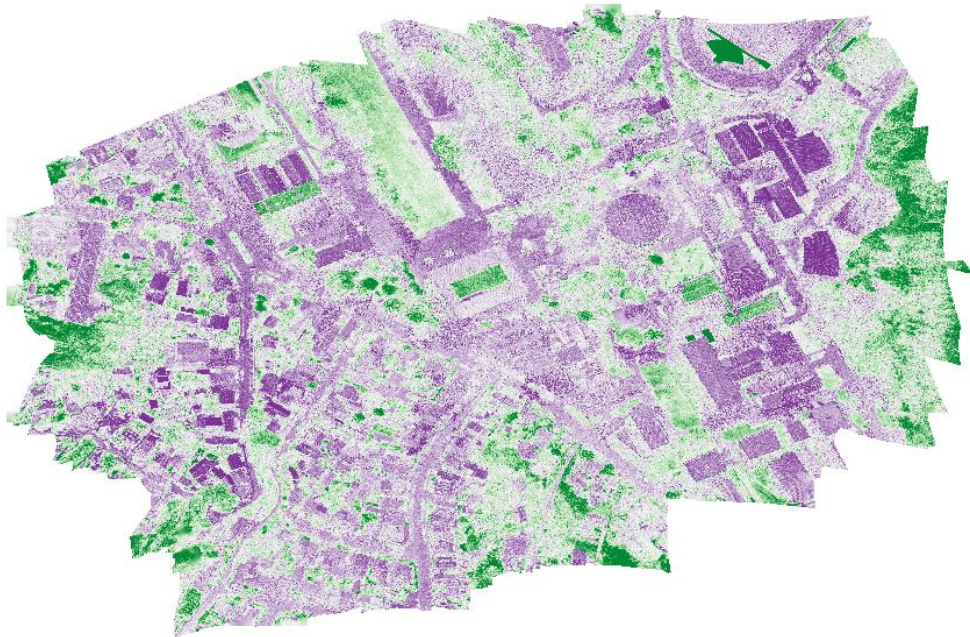


Figura 9. Índice NDRE

Para el GRVI y el GNDVI se utilizó la misma paleta de colores que para el primer índice, de verde a rojo.

En este caso, el GRVI permite observar y diferenciar los puntos donde hay vegetación con mayor vigor. Por otra parte, el GNDVI permite observar lo mismo que con el NDVI, pero con un mayor grado de precisión.



Figura 10. Índice GRVI



Figura 11. Índice GNDVI

Frente a la pregunta ¿Qué campus de la UdeC cuenta con más vegetación?, se utilizó el índice NDVI, la elección fue porque entre los 4 índices anteriormente mencionados es el que muestra más claramente la vegetación y se establecen los rangos :

- 1 □ NDVI < -0.1 Corresponde a cuerpos de Agua
- 0.1 □ NDVI < 0.2 Corresponde usualmente a roca, arena, nieve
- 0.2 □ NDVI < 0.4 Arbustos, césped
- 0.4 □ NDVI □ 1 Árboles y Bosques

Con ayuda del lenguaje python dentro de Qgis se pudo efectuar el siguiente script para calcular la cantidad de vegetación en los campus, se sumó la cantidad de celdas que pasaron el umbral 0.2 y se dividió por el tamaño de las celdas con valor dentro del rango NDVI (celdas válidas).

```
#script para calcular cantidad de vegetación en %
from __future__ import division
layer1 = iface.mapCanvas().layer(0)
layer1.name()
import numpy as np
from osgeo import gdal
provider=layer1.dataProvider()
filePath = str(provider.dataSourceUri())
print filePath
ds = gdal.Open(filePath)
myarray = np.array(ds.GetRasterBand(1).ReadAsArray())
count = 0
total_terreno = 0
for x in xrange(myarray.shape[0]):
    for y in xrange(myarray.shape[1]):
        if myarray[x][y]> -1 and myarray[x][y] < 1:
            total_terreno+=1
            if myarray[x][y]>0.2:
                count+=1

print str(100*count/total_terreno)+"%"
```

Obteniendo los siguientes resultados :

Campus	% de Vegetación
Concepción	31.42%
Chillán	36.12%

Tarea 2

Para la realización de este ítem se detalla la lista de los pasos más importantes, primero asumimos que existe un workspace en Geoserver llamado “conce” y un almacén con el mismo nombre de la iteración 1.

1. Creamos una tabla dentro de nuestra base de datos a utilizar, en nuestro caso llamada “dibujos”, esta tiene un atributo , id , linestring y name; bigint, geometry y string respectivamente, donde id es Primary Key.
2. Al cargar el almacén de datos a Geoserver es necesario activar la opción ‘Expose primary keys’.
3. Dar permisos de escritura al usuario Anonymous dentro de Geoserver.
4. Instalar plugin Allow-Control-All-Origin[8] en Google Chrome para que este permita las peticiones al WFS-T.
5. Se utilizaron los plugins Leaflet.wfs-t [6] y Leaflet.draw [7], para realizar las inserciones en el servicio WFS de Geoserver y dibujar respectivamente.

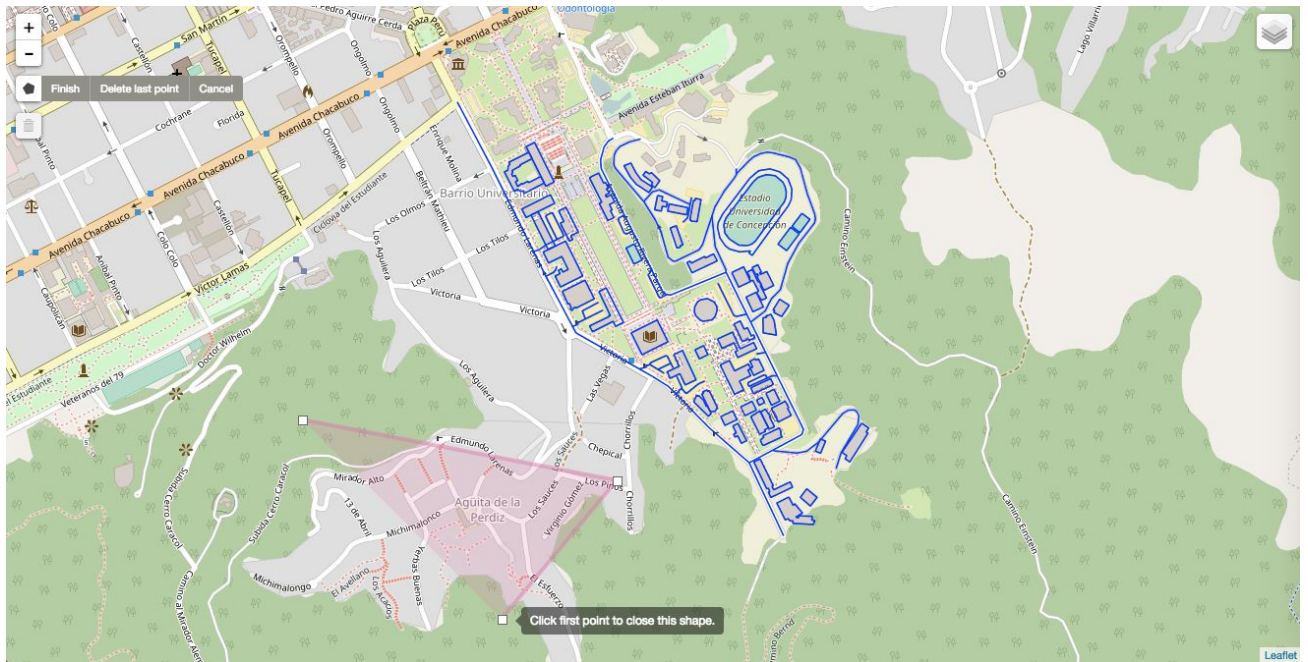


Figura 12. Dibujando Polígono

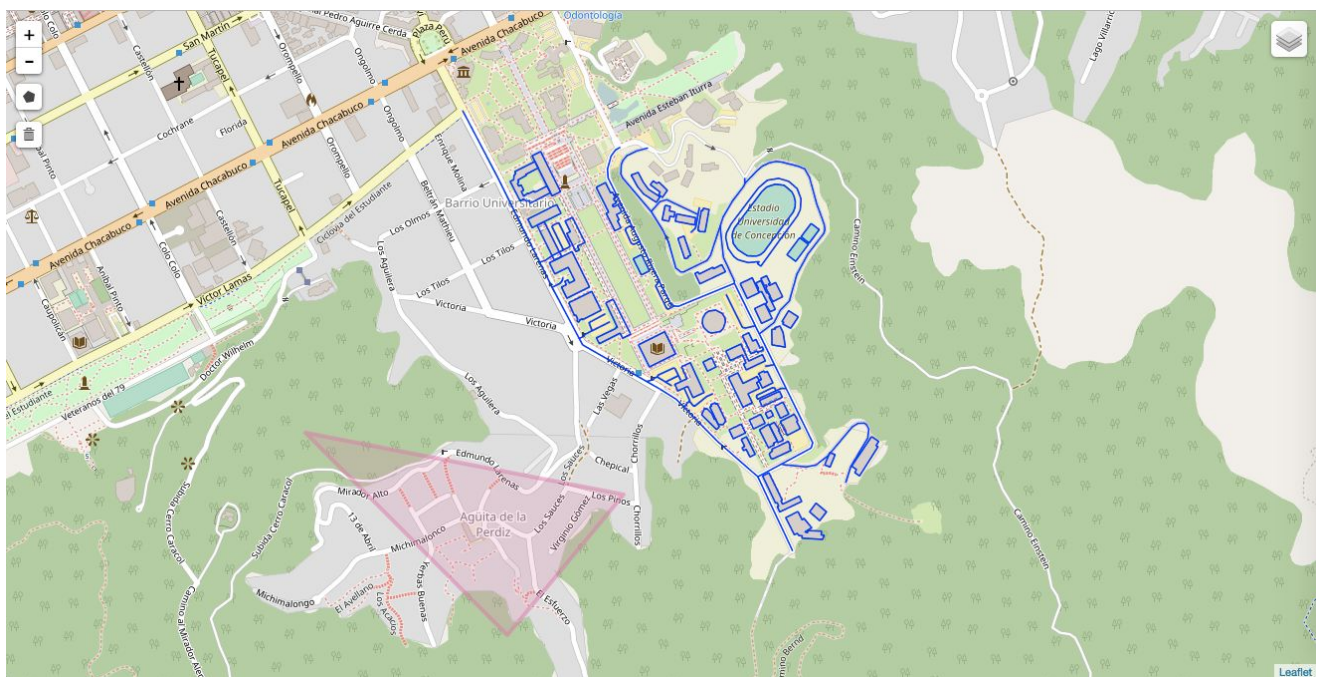


Figura 13. Polígono Dibujado

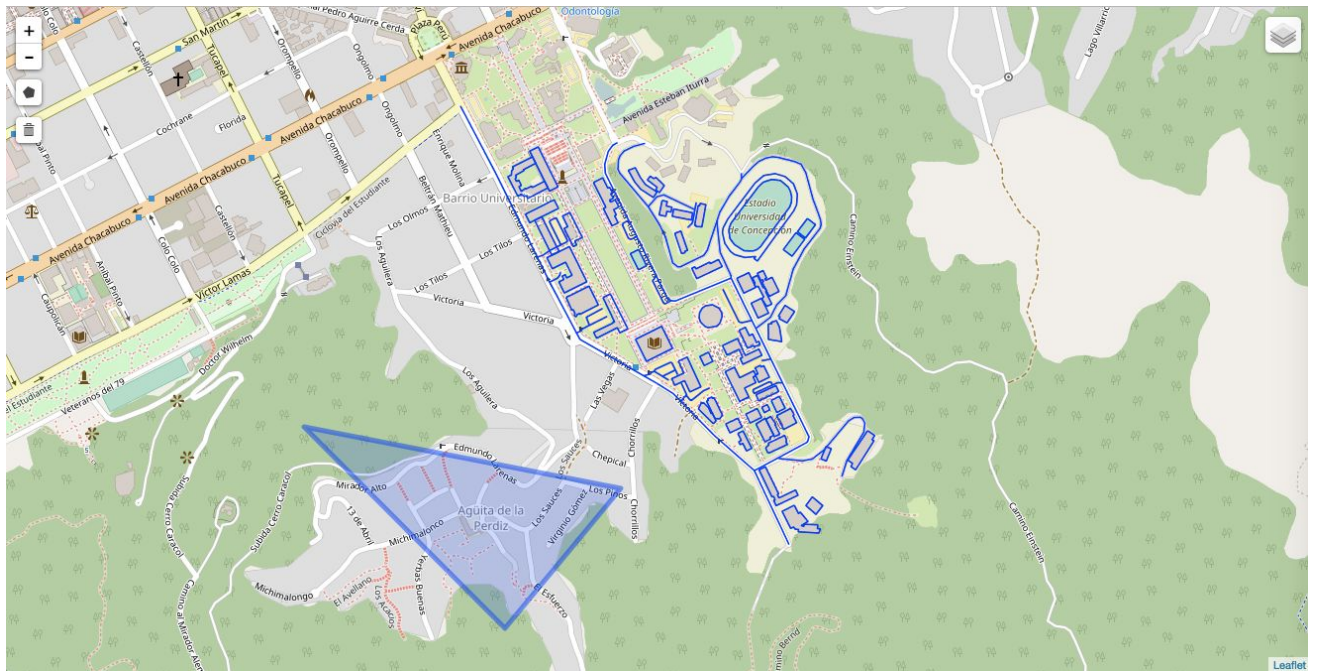


Figura 14. Polígono Persistente en BD.

Referencias

- [1] <https://josm.openstreetmap.de>
- [2] <http://wiki.openstreetmap.org/wiki/Osmosis>
- [3] http://eprints.ucm.es/31423/1/TFM_Juan_Diaz_Cervignon.pdf
- [4] <http://aerialagimagery.com/NDRE.html>
- [5] <https://airborninsight.com.au/mapping-services/agriculture.html>
- [6] <https://github.com/respec/leaflet.wfs-t>
- [7] <https://github.com/Leaflet/Leaflet.draw>
- [8] <https://chrome.google.com/webstore/detail/allow-control-allow-origi/nlfbmbojpeacfgkhkpbjhddihlkkiljbi>